Diskrete Mathematik und Logik

Marcus Kracht
kracht@math.fu-berlin.de

3. Mai 2002

Dieser Text ist die Grundlage der Vorlesung, welche ich im Sommersemester 2001 an der BTU Cottbus gehalten habe. Die vorliegende Fassung stammt vom 24. September 2001. Ich danke Markus Gottwald für das sorgfältige Lesen dieses Manuskripts. Kritik und Anregungen sind jederzeit willkommen.

1. Teil: Mengenlehre I: Mengen, Funktionen und Relationen

Mengen sind die grundlegendsten Objekte in der Mathematik. Prinzipiell lässt sich die gesamte Mathematik mit Hilfe von Mengen darstellen, sodass die Mengenlehre innerhalb der Mathematik eine zentrale Grundlagendisziplin ist. Eine der wichtigsten Errungenschaften der Mengenlehre ist eine Aufklärung des Anzahl- und Ordnungsbegriffs. Die letzten beiden Konzepte sollen uns in diesem und dem nächsten Abschnitt beschäftigen.

Die populärste Theorie der Mengen ist die sogenannte Zermelo–Fränkelsche Mengenlehre ZFC. (Der Buchstabe C steht hier für Englisch *choice*, Deutsch *Wahl*, weil das Auswahlaxiom mit dabei ist.) Wir wollen ihre Axiome hier kurz vorstellen. Die Sprache von ZFC kennt nur einen Typ von Objekt: den der Menge. Es gibt zwei Relationen zwischen Mengen, nämlich die Relation $\in$ und die Gleichheit, geschrieben $=$. Die Relation $\in$ ist weder transitiv noch reflexiv (für eine Definition siehe Teil 2), sie ist sogar sehr kompliziert, wie die Axiome noch lehren werden. Ist $N \in M$, so sagt man, N sei **Element von** M. Dabei ist sowohl M als auch N eine Menge. Trotzdem unterscheidet man Menge und Element oft typographisch, etwa indem man schreibt

$x \in N$. Wir sagen, M sei **Teilmenge von** N, in Zeichen $M \subseteq N$, falls für jedes $x \in M$ auch $x \in N$ ist. Die folgenden Postulate beschreiben die interne Struktur von Mengen.

> Extensionalitätsaxiom. Es gilt $M = N$ genau dann, wenn aus $x \in M$ folgt $x \in N$ und wenn aus $x \in N$ folgt $x \in M$.
> Fundierungsaxiom. Ist M eine Menge, so existiert keine Folge $M = M_0 \ni M_1 \ni M_2 \ni \cdots$.

Extensionalität lässt sich auch so formulieren: $M = N$ genau dann, wenn $M \subseteq N$ und $N \subseteq M$. Das Fundierungsaxiom versteht man am Besten, indem man sich Mengen als Schachteln vorstellt. Eine Schachtel kann wiederum Schachteln enthalten, und diese wiederum Schachteln, in denen wieder Schachteln sein können, und so weiter. Aber erstens ist eine Schachtel nicht in sich selbst enthalten, auch nicht in einer Schachtel, die in ihr enthalten ist, und zweitens kann man Schachteln nicht unendlich oft hintereinander auspacken. Die Schachteln, die man da auspackt, werden immer kleiner und kleiner, und irgendwann ist man fertig mit dem Auspacken. Die kleinste Schachtel ist notwendigerweise leer.

Nun folgen Postulate, die uns sagen, wie wir aus alten Mengen neue Menge machen können. Das erste ist das

> Zweiermengenaxiom. Zu je zwei Mengen M und N existiert die Menge $\{M, N\}$.

Ist insbesondere $M = N$, so bekommen wir die Einermenge $\{M\}$. Es sei M eine Menge. Dann bezeichnen wir mit $\wp(M)$ die Menge der Teilmengen von M. Dass es sich hierbei tatsächlich um eine Menge handelt, sichert man sich durch ein Axiom. Dies formulieren wir so.

> Potenzmengenaxiom. Zu jeder Menge M existiert eine Menge, deren Elemente genau die Teilmengen von M sind. Wir bezeichnen diese Menge mit $\wp(M)$.

Ein nächstes Axiom sichert die Existenz der Vereinigung von Mengen. Wir wollen haben, dass mit Mengen N_1 und N_2 auch die Menge $N_1 \cup N_2$ existiert. Allerdings ist dies für viele Zwecke zu wenig. Man möchte gerne unendlich viele Mengen vereinigen können. Daher formuliert man das etwas kompliziertere

Vereinigungsmengenaxiom. Zu jeder Menge M existiert eine Menge N derart, dass $x \in N$ genau dann, wenn ein $S \in M$ existiert mit $x \in S$. Wir bezeichnen N auch mit $\bigcup M$.

Die Idee hinter diesem Axiom ist die folgende. Ist $M = \{N_1, N_2\}$, so haben wir $\bigcup M = N_1 \cup N_2$. Da wir aber nicht nur zwei oder gar endlich viele Mengen vereinigen können wollen, sondern so viele wie möglich, fassen wir die zu vereinigenden Mengen in eine Menge M zusammen und bilden anschließend $\bigcup M$. Diese enthält genau die Elemente der zu vereinigenden Mengen.

Ist $\varphi(x)$ eine Eigenschaft von Mengen, so schreibt man $\{x : \varphi(x)\}$ für die Menge aller Mengen, welche $\varphi(x)$ erfüllen. Allerdings dürfen wir nicht unkritisch damit umgehen, wie folgendes Beispiel lehrt.

Satz 1 (Russell) *Es gibt keine Menge* $\{x : x \notin x\}$.

Beweis. Angenommen, es gibt diese Menge. Nennen wir sie V. Dann ist entweder (Fall 1) $V \in V$ oder (Fall 2) $V \notin V$. Im Fall 1 gilt $V \notin V$, nach Definition von V. Im Fall 2 gilt $V \in V$, wiederum nach Definition von V. Ein Widerspruch. Q. E. D.

Wir bemerken, dass für jede Menge M gilt: $M \notin M$. Also ist die Menge, wenn sie denn existiert, nicht leer, und sie enthält jede nur denkbare Menge. Man nennt daher V auch das **Universum**. Das Universum können wir uns zwar irgendwie vorstellen, aber es ist keine Menge. Um dennoch nach solch einem Prinzip sorgenfrei Mengen schaffen zu können, definiert man wie folgt.

Aussonderungsaxiom. Es sei $\varphi(x)$ eine Eigenschaft von Mengen und sei M eine Menge. Dann ist $\{x : x \in M \text{ und } \varphi(x)\}$ eine Menge.

Zu je zwei Mengen M und N existieren ferner die Mengen

$$\begin{array}{rcl} M \cap N & := & \{x : x \in M \text{ und } x \in N\} \\ M \cup N & := & \{x : x \in M \text{ oder } x \in N\} \\ M - N & := & \{x : x \in M \text{ und nicht } x \in N\} \end{array}$$

Die erste und die dritte existieren nach dem Aussdonderungsaxiom (wir sondern aus M alle die Elemente aus, die zu N (bzw. nicht zu N) gehören), die zweite Menge existiert nach dem Vereinigungsmengenaxiom.

In der Mathematik bedient man sich sehr oft des Begriffs der *Folge* und des *Paares*. Beide können auf sehr einfache Weise durch Mengen dargestellt werden.

Definition 2 (Kuratowski, Wiener) *Es seien U und V Mengen. Dann ist $\langle U, V\rangle := \{U, \{U, V\}\}$ ein sogenanntes **geordnetes Paar**.*

Wir müssen als erstes nachweisen, dass diese Definition das Gewünschte leistet. Als Vorüberlegung betrachten wir zwei Mengen $M = \{a, b\}$ und $N = \{c, d\}$. Wann ist $M = N$? Die Antwort ist: falls $a = c$ und $d = d$ oder falls $a = d$ und $b = c$. Dabei ist es unerheblich, ob $a = b$ ist oder nicht, oder ob $c = d$ ist oder nicht. Man beachte also, dass hier die Reihenfolge der Elemente, wie sie in der Auflistung erscheinen, keine Rolle spielt.

Satz 3 *Es seien U, U', V, V' Mengen und $\langle U, V\rangle = \langle U', V'\rangle$. Dann ist $U = U'$ und $V = V'$.*

Beweis. Nach Definition haben wir $\langle U, V\rangle = \{U, \{U, V\}\}$ und $\langle U', V'\rangle = \{U', \{U', V'\}\}$. Falls nun $\langle U, V\rangle = \langle U', V'\rangle$, so ist $\{U, \{U, V\}\} = \{U', \{U', V'\}\}$. Dann ist entweder (Fall 1) $U = U'$ oder (Fall 2) $U = \{U', V'\}$. Im Fall 1 ist dann $\{U, V\} = \{U', V'\}$, woraus wegen $U = U'$ zudem folgt, dass $V = V'$. Im Fall 2 haben wir $\{U, V\} = U'$. Also erhalten wir $U = \{U', V'\} = \{\{U, V\}, V'\}$. Das ist aber ausgeschlossen, da U ansonsten nicht fundiert ist. Denn dann haben wir $U \ni \{U, V\} \ni U \ni \{U, V\} \ni \ldots$. Q. E. D.

Definition 4 *Es seien M und N Mengen. Dann sei $M \times N := \{\langle x, y\rangle : x \in M, y \in N\}$. Diese Menge heißt das **kartesische Produkt von** M **und** N. Im Falle, dass $M = N$, schreibt man anstelle von $M \times N$ auch M^2.*

Definition 5 *Es seien M und N Mengen. Eine **Relation von** M **nach** N ist eine Teilmenge R von $M \times N$. Ist $\langle x, y\rangle \in R$, so sagt man, x **stehe in der Relation** R **zu** y. Man schreibt alternativ $x \mathrel{R} y$.*

Insbesondere ist die leere Menge eine Relation von $M \times N$. Eine andere Relation ist die sogenannte **Allrelation**: dies ist die Menge $M \times N$ selbst. Zu zwei Relationen R und S von M nach N existiert dann der Schnitt $R \cap S$, die Vereinigung $R \cup S$, und das Komplement $(M \times N) - R$ von R bezüglich $M \times N$. Ferner ist

$$R^{\smile} := \{\langle y, x\rangle : \langle x, y\rangle \in R\}$$

die sogenannte zu R **konverse** Relation. Ist $R \subseteq M \times N$ und $S \subseteq N \times O$, so definieren wir

$$R \circ S := \{\langle x, z\rangle : \text{ es existiert } y \in N : x \mathrel{R} y \mathrel{S} z\}$$

Dies ist das sogenannte **Produkt von** R **und** S. Man beachte, dass $S \circ R$ nicht definiert sein muss. Ist $M = N$, so kann man noch mehr spezielle Relationen definieren. Zunächst existiert die sogenannte **Diagonale**, $\Delta_M = \{\langle x, x\rangle : x \in M\}$. Ferner hat man

$$\begin{array}{lcl} R^0 & := & \Delta_M \\ R^{n+1} & := & R^n \circ R \\ R^* & := & \bigcup_{n \in \mathbb{N}} R^n \end{array}$$

R^n ist das sogenannte n–fache Produkt von R mit sich selbst, R^* die **Iteration von** R.

Definition 6 *Es seien M und N Mengen. Eine* ***Funktion von*** M ***nach*** N *ist eine Relation $f \subseteq M \times N$ derart, dass (1) für jedes $x \in M$ ein $y \in N$ existiert mit $\langle x, y\rangle \in f$ und (2) aus $\langle x, y_0\rangle, \langle x, y_1\rangle \in f$ folgt $y_0 = y_1$. Wir schreiben $f : M \to N$ um zu sagen, dass f eine Funktion von M nach N ist. $f(x)$ bezeichnet das eindeutig bestimmte y mit $\langle x, y\rangle \in f$. Es heißt M der* ***Definitionsbereich*** *der Funktion f und N ihr* ***Wertebereich****. f heißt* ***injektiv*** *(oder auch eine* ***Einbettung****), falls aus $f(x_0) = f(x_1)$ folgt $x_0 = x_1$,* ***surjektiv****, falls zu jedem $y \in N$ ein $x \in M$ existiert mit $y = f(x)$. f heißt* ***bijektiv****, falls f sowohl injektiv wie surjektiv ist.*

Gelegentlich wird zwischen der Funktion und ihrem **Graphen** unterschieden. Letzterer ist nicht anderes als die Menge $\langle x, f(x)\rangle$, $x \in M$. Angesichts der eben getroffenen Definition sind Funktion und Graph identisch. Man beachte nun Folgendes: ist $M_1 \subseteq M$, so existiert zu jeder Funktion $f : M \to N$ eine eindeutig bestimmte Funktion $g : M_1 \to N$ mit $f(x) = g(x)$ für alle $x \in M_1$. Diese bezeichnet man mit $f \restriction M_1$ und nennt sie die **Einschränkung von** f **auf** M_1. Wir haben $g = f \cap (M_1 \times N)$.

Ist $A \subseteq M$ und $f : M \to N$, so bezeichnet $f[A]$ die Menge $\{f(x) : x \in A\}$, welche auch das **direkte Bild von** A **unter** f heißt. Dabei muss man sorgfältig zwischen $f(A)$ und $f[A]$ unterscheiden. Ist zum Beispiel $f : \{a, \{a\}\} \to \{b, c\}$ eine Funktion mit $a \mapsto b$ und $\{a\} \mapsto c$, so ist $f(\{a\}) = c$ aber $f[\{a\}] = \{b\}$.

2. Teil: Mengenlehre II: Ordnungen

In diesem Kapitel wollen wir uns mit Relationen, speziell mit *Ordnungen* befassen.

Definition 7 *Wir definieren folgende Eigenschaften von Relationen $R \subseteq M^2$. R heißt* ***transitiv****, falls aus $x \; R \; y$ und $y \; R \; z$ folgt $x \; R \; z$. R heißt* ***reflexiv****, falls $x \; R \; x$ für alle $x \in M$. R heißt* ***strikt****, falls für kein $x \in M$ gilt $x \; R \; x$. R heißt* ***symmetrisch****, wenn aus $x \, R \, y$ folgt $y \, R \, x$, und R heißt* ***antisymmetrisch****, wenn aus $x \; R \; y$ und $y \; R \; x$ folgt, dass $x = y$. R heißt* ***linear****, falls für alle $x, y \in R$ gilt: $x \; R \; y$ oder $x = y$ oder $y \; R \; x$. Eine* ***Äquivalenzrelation*** *ist eine Relation, welche reflexiv, symmetrisch und transitiv ist.*

Beispiel 1. Es sei $m \mid n$ genau dann, wenn m Teiler von n ist. Es ist $\mid$ auf den natürlichen Zahlen > 0 transitiv, reflexiv, und antisymmetrisch, aber nicht linear.

Beispiel 2. Die Relation $<$ auf den ganzen Zahlen ist transitiv, linear und strikt.

Beispiel 3. Es sei $x \, S_5 \, y$, falls $|x - y| \leq 5$ ist. S_5 ist reflexiv und symmetrisch aber nicht transitiv.

Man kann manche der obigen Eigenschaften auch etwas konziser aufschreiben. R ist genau dann reflexiv, wenn $\Delta_M \subseteq R$. R ist genau dann symmetrisch, wenn $R = R^{\smile}$. R ist genau dann transitiv, wenn $R \circ R \subseteq R$.

Definition 8 *Es sei $R \subseteq M^2$ eine Relation. R heißt* ***strikte Ordnung auf*** *M, falls R transitiv und strikt ist. R heißt* ***fundiert****, falls jede nichtleere Menge $U \subseteq M$ ein bezüglich R kleinstes Element enthält.*

Beispiel 4. $< \subseteq \mathbb{N}^2$ ist eine strikte, lineare, fundierte Ordnung. (Wir weisen darauf hin, dass für uns stets $0 \in \mathbb{N}$ ist, also 0 eine natürliche Zahl ist.)

Beispiel 5. Die Menge $\mathbb{Z}$ der ganzen Zahlen mit der Relation $<$ ist hingegen nicht fundiert geordnet. Die folgende Relation auf $\mathbb{Z}$ ist allerdings fundiert. Wir sagen, $m \sqsubset n$, wenn entweder (a) $|m| < |n|$ oder (b) $|m| = |n|$ und $m < 0$. Die Ordnung sieht wie folgt aus.

$$0, -1, 1, -2, 2, -3, 3, -4, 4, \ldots$$

Beispiel 6. Es sei B die Menge der endlichen Folgen aus 0 und 1. (Mit der Definition von Folgen, die wir weiter unten geben werden, ist B die Menge der Funktionen von den endlichen Zahlen nach $2 = \{0, 1\}$.) Wir setzen $\vec{x} \, P \, \vec{y}$, falls $\vec{x}$ ein echtes Anfangsstück von $\vec{y}$ ist. P ist dann eine strikte, fundierte Ordnung, aber nicht linear.

Eine Relation R auf einer Menge M ist genau dann fundiert, wenn es keine $x_i \in M$ gibt, $i \in \mathbb{N}$, wo $x_{i+1} \, R \, x_i$ ist für alle $i \in \mathbb{N}$. Man vergleiche dies mit

Fundiertheitsaxiom aus dem vorigen Teil. Dies besagt, mit den Worten der Definition 8 dass jede Menge M die Relation $\in$ auf den hereditären Elementen von M fundiert ist. (Die hereditären Elemente von M sind diejenigen, die entweder Elemente von M sind oder Elemente von Elementen oder Elemente von Elementen von Elementen und so weiter.)

Definition 9 *$R \subseteq M^2$ heißt eine **Wohlordnung**, falls R eine strikte, fundierte, lineare Ordnung auf M ist. Eine **wohlgeordnete Menge** ist ein Paar $\langle M, R\rangle$, sodass R eine Wohlordnung auf M ist. Man notiert auch gerne $<$ anstelle von R.*

Ein besonders wichtiges Axiom der Mengenlehre ist das sogenannte *Auswahlaxiom*. Es ist beweisbar äquivalent mit dem nicht minder nützlichen

> Wohlordnungsaxiom. Zu jeder Menge M existiert eine Wohlordnung auf M.

Die Nützlichkeit des Wohlordnungsaxioms wird sich noch später erweisen.

Es seien $\underline{M} = \langle M, <\rangle$ und $\underline{N} = \langle N, <'\rangle$ wohlgeordnete Mengen. Wir schreiben $\underline{M} \preccurlyeq \underline{N}$, falls es eine Funktion $f : M \to N$ gibt derart, dass (a) für alle $x, y \in M$ gilt $x < y$ genau dann, wenn $f(x) <' f(y)$ und (b) ist $y < f(x)$ für $x \in M$ und $y \in N$, so existiert ein $z \in M$ mit $f(z) = y$. (b) bedeutet mit anderen Worten, dass $f[M]$ nach unten abgeschlossen ist. Ist f surjektiv, so heißen $\underline{M}$ und $\underline{N}$ **isomorph** und f ein **Isomorphismus von $\underline{M}$ nach $\underline{N}$**.

Ist $\underline{M}$ eine wohlgeordnete Menge und $x \in M$, so sei $\downarrow x := \{y : y < x\}$. Ferner sei $H_{\underline{M}}(x) := \langle \downarrow x, <\upharpoonright (\downarrow x)\rangle$. Man nennt dies den **Abschnitt von** x. So ist zum Beispiel für $\underline{\mathbb{N}} := \langle \mathbb{N}, <\rangle$ $\downarrow 5 = \{0, 1, 2, 3, 4\}$ und $H_{\underline{\mathbb{N}}}(5)$ die Einschränkung der Ordnung auf auf $\{0, 1, 2, 3, 4\}$.

Satz 10 (Cantor) *Es seien $\underline{M} = \langle M, <\rangle$ und $\underline{N} = \langle N, <'\rangle$ wohlgeordnete Mengen. Dann ist $\underline{M} \preccurlyeq \underline{N}$ oder $\underline{N} \preccurlyeq \underline{M}$. Gilt ferner sowohl $\underline{M} \preccurlyeq \underline{N}$ als auch $\underline{N} \preccurlyeq \underline{M}$, so ist $\underline{M}$ isomorph zu $\underline{N}$.*

Beweis. Wir definieren eine Relation $Z \subseteq M \times N$ wie folgt. $\langle x, y\rangle \in Z$ genau dann, wenn Z ein Isomorphismus von $H_{\underline{M}}(x)$ auf $H_{\underline{N}}(y)$ ist. Dies bedeutet im Klartext Folgendes. Wenn wir wissen wollen, ob $x \in M$ in Relation Z zu y steht, so müssen wir nur schauen, ob Z einen Isomorphismus der Abschnitte $H_{\underline{M}}(x)$ und $H_{\underline{N}}(y)$ vermittelt. Wir wollen nun als erstes zeigen, dass Z eine bijektive Funktion ist (auf ihrem Definitions- und Wertebereich). Daraus folgt nach Definition, dass Z ein Isomorphismus ist. Wir nehmen an, dass $z \neq y$

ein Element in N ist, und dass $\langle x, y\rangle, \langle x, z\rangle \in Z$. Dann ist entweder $z < y$ oder $y < z$. Wir können oBdA annehmen, dass $y < z$. Dann haben wir: Z ist ein Isomorphismus von $H_{\underline{M}}(x)$ auf $H_{\underline{N}}(z)$ sowie auf $H_{\underline{N}}(y)$. Aber $y \in \downarrow z$, während $z \in \downarrow y$ nicht gilt. Dies ist ein Widerspruch. Also kann weder $y < z$ noch $z < y$ gelten. Ähnlich zeigt man, dass aus $\langle x, y\rangle, \langle z, y\rangle \in Z$ folgt $x = y$. Also bestimmen sich x und y gegenseitig.

Jetzt zeigen wir noch, dass entweder Z auf allen Elementen von M definiert ist oder das Bild von Z alle Elemente von N enthält. Im ersten Fall ist dann $\underline{M} \preccurlyeq \underline{N}$ und im zweiten Fall $\underline{N} \preccurlyeq \underline{M}$. (Im zweiten Fall ist $Z^{\smile}$ die gesuchte Funktion.) Wir setzen U die Menge aller $x \in M$, auf denen Z nicht definiert ist, und V die Menge aller $y \in N$, die nicht im Bild von Z sind. Zu zeigen ist, dass eine der beiden Menge nicht leer ist. Angenommen, U und V sind beide nicht leer. Dann hat U ein eindeutig bestimmtes kleinstes Element x und V ein eindeutig bestimmtes kleinstes Element y. Dann ist $Z \subseteq (\downarrow x) \times (\downarrow y)$ wie gerade gezeigt eine Bijektion. Ferner ist Z ein Isomorphismus von $H_{\underline{M}}(x)$ nach $H_{\underline{N}}(y)$, sodass laut Definition nunmehr $\langle x, y\rangle \in Z$. Widerspruch. Q. E. D.

Definition 11 *Eine* ***Ordinalzahl*** *ist eine Menge M derart, dass $\langle M, \in\rangle$ eine wohlgeordnete Menge ist. Da $\in$ auf M bereits festliegt, wird zwischen der Ordinalzahl M und der wohlgeordneten Menge $\langle M, \in\rangle$ nicht unterschieden.*

Ordinalzahlen werden mit kleinen griechischen Buchstaben gekennzeichnet. Als Beispiel für Ordinalzahlen geben wir folgende, auf János Neumann zurückgehende Definition der natürlichen Zahlen:

$$\begin{array}{lcl} 0 & := & \varnothing \\ n+1 & := & n \cup \{n\} \end{array}$$

Man rechnet leicht nach, dass mit diesen Definitionen $n = \{0, 1, \ldots, n-1\}$. Wir haben somit

$$\begin{array}{rcl} 0 & = & \varnothing \\ 1 & = & \{\varnothing\} \\ 2 & = & \{\varnothing, \{\varnothing\}\} \\ 3 & = & \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\} \end{array}$$

Wir erhalten aus dem vorigen Satz die Folgerung, dass für je zwei Ordinalzahlen κ und λ gilt $\kappa \preccurlyeq \lambda$ oder $\lambda \preccurlyeq \kappa$.

Satz 12 *Es seien κ und λ Ordinalzahlen. Ist $\kappa \preccurlyeq \lambda$ sowie $\lambda \preccurlyeq \kappa$, so gilt schon $\kappa = \lambda$.*

Beweis. Da $\kappa \preccurlyeq \lambda$ und $\lambda \preccurlyeq \kappa$, existieren ordnungserhaltende Einbettungen $f : \kappa \to \lambda$ und $g : \lambda \to \kappa$. Es ist $f[\kappa]$ ein Anfangsstück von λ, und so $g \circ f[\kappa] = g[f[\kappa]]$ ein Anfangsstück von κ. (Wir erinnern hier an den Unterschied zwischen $f(\kappa)$ (welches nicht definiert ist) und $f[\kappa]$, dem direkten Bild der Definitionsmenge κ unter f.) Angenommen, $g \circ f[\kappa]$ sei ein echtes Anfangsstück. Dann existiert ein bezüglich $\in$ minimales $x \in \kappa$, derart, dass $g \circ f(x) \neq x$. Wir haben dann $y := g \circ f(x) < x$. Aber dann ist nach Wahl von x $g \circ f(y) = y$, im Widerspruch zur Annahme. Also ist $g \circ f : \kappa \to \kappa$ eine Bijektion. Ebenso zeigt man, dass $f \circ g : \lambda \to \lambda$ eine Bijektion ist. Als Letztes werden wir zeigen, dass f schon die identische Abbildung ist. Daraus folgt dann $\kappa = \lambda$. Angenommen, dies sei nicht der Fall. Dann existiert ein kleinstes x mit $f(x) \neq x$. Nun ist $f(x) = \{y : y \in \lambda, y < f(x)\}$ und $x = \{z : z \in \kappa, z \in x\}$. Nach Annahme ist für jedes $z \in x$ $f(z) = z$. Daher ist $z = f(z) \in f(x)$. Also haben wir $x \subseteq f(x)$. Ferner existiert zu jedem $y < f(x)$ ein $z' \in \kappa$ mit $f(z') = y$. Es ist leicht zu sehen, dass $f(z') = z'$ ist. Damit haben wir auch $f(x) \subseteq x$ gezeigt. Damit ist $f(x) = x$, wie versprochen. Q. E. D.

Halten wir also fest: je zwei Ordinalzahlen sind mittels $\preccurlyeq$ vergleichbar. Sind zwei Ordinalzahlen isomorph, so sind sie sogar gleich. Man schreibt nun $\kappa \leq \lambda$, falls $\kappa \preccurlyeq \lambda$ ist, und $\kappa < \lambda$, falls $\kappa \leq \lambda$ aber $\kappa \neq \lambda$. Ferner: ist $\kappa < \lambda$, so ist $\kappa \in \lambda$ — und umgekehrt. Wir haben oben bereits die Ordinalzahlen für die natürlichen Zahlen definiert. Diese sind endliche Zahlen. Dabei sind wir die Definition der Endlichkeit noch schuldig geblieben.

Definition 13 *Eine Menge M heißt* ***endlich****, falls jede injektive Funktion $f : M \to M$ auch surjektiv ist.*

Nehmen wir noch folgendes Axiom hinzu.

Unendlichkeitsaxiom. Es existiert eine unendliche Menge.

Es zeigt sich, dass damit auch die folgende Menge existiert

$$\omega = \{0, 1, 2, 3, 4, \ldots\}$$

Diese ist gleichzeitig eine Ordinalzahl, und sie ist auch die kleinste unendliche Ordinalzahl. Sie enthält genau alle endlichen Zahlen. Als wohlgeordnete Menge ist sie isomorph $\underline{\mathbb{N}} = \langle \mathbb{N}, < \rangle$. Elemente von ω heiß von jetzt ab auch **natürliche Zahlen**. Dass diese Menge nicht endlich ist (im Sinne der eben gegebenen Definition), ist leicht zu sehen. Die Funktion $f : \omega \to \omega : n \mapsto n+1$ ist injektiv aber nicht surjektiv.

Ferner verlangen wir noch das

> Ersetzungsaxiom. Ist $f : M \to N$ eine Funktion und $U \subseteq M$, so auch $f[U]$.

Wir schließen diesen Abschnitt mit einer Definition der Folge.

Definition 14 *Es sei κ eine Ordinalzahl. Dann ist eine κ–**lange Folge über** M eine Funktion $f : \kappa \to M$.*

Sei zum Beispiel die Folge $1, 4, 9, 16$ gegeben. Diese ist jetzt laut Definition eine Funktion $f : 4 \to \mathbb{N}$ mit $f(0) = 1$, $f(1) = 4$, $f(2) = 9$ und $f(3) = 16$. Die gesamte Folge der Quadratzahlen ist eine Funktion $g : \omega \to \mathbb{N}$ mit $g(n) = (n+1)^2$. Es ist $g \supseteq f$. Man möge sich nicht irritieren lassen, dass einmal ω steht und das andere Mal $\mathbb{N}$. Dies hat rein ästhetische Gründe. Wir werden in Zukunft überwiegend ω benutzen, wobei dies mit $\mathbb{N}$ synonym ist.

Meist braucht man nur zwei Typen von Folgen: die endlichen und die ω–langen, die deswegen oft auch schlicht *unendliche Folgen* genannt werden. Bei der Formulierung des Fundiertheitsaxioms hatten wir auf diesen Begriff übrigens schon zurückgegriffen.

3. Teil: Mengenlehre III: Induktionsprinzipien

Zu den wichtigsten Instrumenten der formalen Begriffsbildung der Mathematik zählen die sogenannte Induktion und die Rekursion. Bei der Induktion handelt sich um ein allgemeines Prinzip, mit dem man Beweise führt wie auch Begriffe formal definiert. Der am meiste gebrauchte Typ ist die **Nachfolgerinduktion**.

> Nachfolgerinduktion. Eine Eigenschaft P trifft genau dann auf alle natürliche Zahlen zu, wenn gilt:
>
> 1. $P(0)$.
> 2. Für alle $n \in \mathbb{N}$: ist $P(n)$, so auch $P(n+1)$.

Auch wenn man dieses Prinzip sogar beweisen kann, wollen wir dies hier nicht tun, sondern uns mit der Anschauung seiner Richtigkeit begnügen. Cantor hat nun erkannt, dass sich eine Eigenschaft für alle Ordinalzahlen beweisen lässt, wenn man der Nachfolgerinduktion noch ein drittes Element hinzufügt, welches die Richtigkeit einer Eigenschaft für alle Limeszahlen sichert. Dazu zunächst einige Definitionen.

Definition 15 *Es seien κ und λ Ordinalzahlen. λ ist* ***direkter Vorgänger*** *von κ, falls $\lambda < \kappa$ ist und keine Ordinalzahl μ existiert mit $\lambda < \mu < \kappa$. Wir schreiben dann auch $\lambda + 1$ anstelle von κ.*

Es ist leicht zu sehen, dass

$$\lambda + 1 = \lambda \cup \{\lambda\} .$$

Definition 16 *Eine Ordinalzahl κ heißt* ***Nachfolgerzahl****, falls ein λ existiert mit $\kappa = \lambda + 1$. κ heißt* ***Limeszahl****, falls $\kappa \neq 0$ und κ keinen direkten Vorgänger hat.*

So ist zum Beispiel die kleinste Limeszahl ω. Denn $\omega \neq 0$, und der Nachfolger von n ist $n + 1 \neq \omega$. ω hat also keinen unmittelbaren Vorgänger.

Lemma 17 *Auf eine beliebige Ordinalzahl κ trifft genau einer der folgenden Fälle zu.*

1. *$\kappa = 0$.*
2. *κ ist Nachfolgerzahl.*
3. *κ ist Limeszahl.*

> Transfinite Induktion. Eine Eigenschaft P trifft genau dann auf alle Ordinalzahlen zu, wenn gilt:
>
> 1. $P(0)$.
> 2. Für alle Ordinalzahlen κ: ist $P(\kappa)$, so auch $P(\kappa + 1)$.
> 3. Für alle Limesordinalzahlen μ: gilt $P(\kappa)$ für alle $\kappa < \mu$, so auch $P(\mu)$.

Zum Beweis nehmen wir an, P treffe nicht auf alle Ordinalzahlen zu. Dann existiert eine kleinste Ordinalzahl, etwa μ, sodass $P(\mu)$ nicht der Fall ist. (Fall 1.) $\mu = 0$; kann sicher nicht gelten. (Fall 2.) μ ist Nachfolgerzahl. Dann ist $\mu = \kappa + 1$. Nach Wahl von μ gilt $P(\kappa)$, also nach Annahme auch $P(\kappa + 1) =$

$P(\mu)$. (Fall 3.) μ ist Limeszahl. Dann gilt nach Wahl von μ $P(\kappa)$ für alle $\kappa < \mu$, und daher haben wir $P(\mu)$ nach der dritten Klausel.

Die dritte Klausel allerdings trägt schon alleine die ganze Last. Um dies zu sehen, zeigen auch folgendes Prinzip.

> Ordnungsinduktion. Eine Eigenschaft P trifft genau dann auf alle Ordinalzahlen zu, wenn für alle Ordinalzahlen μ: ist $P(\kappa)$ für alle $\kappa < \mu$, so auch $P(\mu)$.

Wir wollen die Behauptung beweisen. Sei im Gegensatz zur Behauptung $P(\mu)$ falsch für ein μ. Dann existiert eine kleinste Ordinalzahl ν derart, dass $P(\nu)$ falsch ist. Dann gilt nach Wahl von ν $P(\kappa)$ für alle $\kappa < \nu$, nach Annahme über P daher $P(\nu)$. Dies widerspricht der Wahl von ν. Also existiert kein μ derart, dass $P(\mu)$ falsch ist.

Die Rekursion ist im Gegensatz zur Induktion ein Verfahren zur Definition beziehungsweise Herstellung von Funktionen. Wir geben analog zu den erwähnten Induktionsprinzipien nun Rekursionsprinzipien an, welche uns gestatten, eine Funktion zu definieren. Zunächst die sogenannte

> Primitive Rekursion. Es sei M eine Menge, $m \in M$, sowie $F(-,-) : \mathbb{N}\times M \to M$ eine Funktion, deren erstes Argument eine natürliche Zahl ist. Dann existiert eine und nur eine Funktion $f : \mathbb{N} \to M$, welche folgende Bedingungen erfüllt.
>
> 1. $f(0) = m$.
> 2. $f(n+1) = F(n, f(n))$.

Die Eindeutigkeit kann man nun wiederum durch Induktion über die natürlichen Zahlen zeigen. Die Existenz ist etwas schwieriger, und geht wie folgt. Zu jeder Zahl k definieren wir eine Funktion $f_k : k = \{0, 1, \ldots, k-1\} \to M$, welche die obigen Bedigungen für $n < k - 1$ erfüllt. Anschließend zeigt man, dass aus der Existenz von f_k auch die Existenz von f_{k+1} folgt. Ferner gilt $f_k \subsetneq f_{k+1}$. Nun setzen wir

$$f := \bigcup_{k\in\mathbb{N}} f_k .$$

Dies ist die gewünschte Funktion.

Als Beispiel definieren wir die Fakultät und die Exponentiation. Es sei $F_1(n, k) := (n + 1)k$. Dann existiert eine eindeutig bestimmte Funktion !, welche folgende Bedingungen erfüllt:

$$0! = 1, \qquad (n+1)! = F_1(n, n!) .$$

Man rechnet leicht nach, dass gilt $1! = 1$, $2! = 2$, $3! = 6$, und so weiter. Dies ist ein Beispiel einer Funktion von ω nach ω.

Die Exponentiation r^n für ein festes r ist eine Funktion von ω nach $\mathbb{R}$ (oder auch $\mathbb{C}$, ganz nach Belieben), welche wie folgt definiert ist.

$$r^0 := 1, \qquad r^{n+1} := r^n \cdot r \, .$$

Auch diese ist rekursiv definiert und zwar unter Verwendung der Funktion $F_2(n, x) := x \cdot r$. (Diese hängt also von n gar nicht ab, aber das macht nichts.) Wir weisen darauf hin, dass man auf diese Weise lediglich das Potenzieren r^n für ein festes r definiert hat. Nun ist es ein Leichtes, in der Definition auch noch Parameter zuzulassen. Es hängt dann F wie auch m zusätzlich von einem oder mehreren Parametern ab, und wir bekommen eine ganze Schar von Funktionen, für jeden Wert der Parameter eine.

Es ist an dieser Stelle zu bemerken, dass man mit Hilfe der primitiven Rekursion allein aus der Nachfolgerfunktion sämtliche arithmetischen Funktionen definieren kann. Wir haben eben gesehen, wie man das Potenzieren mittels primitiver Rekursion aus der Multiplikation definieren kann. Ebenso kann man die Multiplikation mit Hilfe der Addition und schließlich die Addition aus der Nachfolgerfunktion definieren. Den Nutzen aus dieser Überlegung werden wir im nächsten Teil ziehen, wenn wir diese Operationen auf den gesamten Ordinalzahlen definieren werden. Damit dies gelingt, müssen wir dieses Schema zu einem Schema der transfiniten Rekursion ergänzen.

> Transfinite Rekursion. Es sei M eine Menge, $m \in M$, λ eine Ordinalzahl, $F(-,-) : \lambda \times M \to M$ eine Funktion und $H(-) : \wp(\lambda \times M) \to M$ eine Funktion, welche als Argument Relationen von λ nach M nimmt und als Wert ein Element aus M ausgibt. Dann existiert eine und nur eine Funktion $f : \lambda \to M$, welche folgende Bedingungen erfüllt.
>
> 1. $f(0) = m$.
> 2. $f(\kappa + 1) = F(\kappa, f(\kappa))$.
> 3. $f(\mu) = H(f \restriction \mu)$, falls μ Limeszahl.

Die dritte Klausel ist am schwierigsten zu interpretieren, obwohl ihre Grundidee ganz einfach ist. Ist μ eine Limeszahl, so haben wir in Form von H ein Prinzip, welches $f(\mu)$ bestimmt, indem es den Verlauf der Funktion für alle Zahlen $< \mu$ anschaut und dann sagt, wie es weitergehen soll. Der Verlauf der

Funktion unterhalb von μ ist gerade die Funktion $f \upharpoonright \mu$. Falls sie vorliegt, so ist sie eine Teilmenge von $\lambda \times M$, somit ein Element von $\wp(\lambda \times M)$. Damit ist $H(f \upharpoonright \mu)$ wohldefiniert und aus M.

Man mag sich fragen, warum wir nicht einfach die Rekursion über alle Ordinalzahlen laufen lassen. Die Antwort ist ganz einfach. Falls wir dies täten, so müssten wir eine Funktion von der Gesamtheit aller Ordinalzahlen nach M bekommen, aber die Gesamtheit der Ordinalzahlen ist keine Menge. (Im nächsten Abschnitt werden wir sehen, warum das so ist.) Trotzdem kann man sicht helfen, indem man bemerkt, dass für jede Ordinalzahl $\mu < \lambda$ bei gleichem Anfangswert die transfinite Rekursion mit $F \upharpoonright \mu$ anstelle von F gerade die Einschränkung $f \upharpoonright \mu$ anstelle der Funktion f liefert. Somit bekommt man im Wesentlichen keine neue Funktionen, wenn man die Ordinalzahl größer oder kleiner macht, sondern man verändert nur den Ausschnitt ein und derselben 'Funktion', den man zu sehen bekommt.

Wir geben ein instruktives Beispiel für den Limesschritt. Kunze hat unendlich großen Durst. Er bestellt also um 23 Uhr zehn Gläser Cola, stellt sie in einer Reihe auf, und trinkt das erste Glas. Nach einer halben Stunde bestellt er wieder zehn, stellt sie hinter die anderen und trinkt das zweite Glas. Nach einer viertel Stunde schließlich bestellt er wieder zehn Gläser, trinkt das dritte. Und so weiter. Lauterbach nebenan hat ebenfalls unendlich großen Durst. Er bestellt auch um 23 Uhr zehn Gläser Apfelsaft, reiht sie auf und trinkt das letzte Glas aus. Nach einer halben Stunde lässt er weitere zehn Gläser kommen, reiht sie nach den anderen auf und trinkt das letzte aus. Nach einer viertel Stunde wieder zehn Gläser hintan gestellt, und das letzte getrunken. Als es Mitternacht ist, schließt der Wirt. Zu seiner Überraschung stehen bei Lauterbach eine unendliche Reihe voller Gläser, bei Kunze kein einziges. Aber beide beteuern, unendlich viel getrunken zu haben. Was ist passiert?

Es kommt hier wesentlich darauf an, dass die Gläser eine Ordnung bekommen. Nennen wie die Gläser, die Kunze zuerst bestellt, 0, 1, 2 bis 9. Kunze trinkt Glas Nr. 0 und bestellt die nächsten 10. Die heißen jetzt 10, 11, bis 19. Kunze trinkt Glas Nr. 2. Und so weiter. Das bedeutet: bei der $n$ten Runde bestellt Kunze die Gläser Nr. $10n$ bis $10n + 9$ und trinkt Glas Nr. n aus. Offensichtlich wird jedes Glas irgendwann ausgetrunken. Bei Lauterbach sieht das anders aus. Auch er lässt sich Gläser 0 bis 9 kommen, trinkt aber Glas Nr. 9 aus. Die nächste Runde besteht jetzt aus Glas Nr. 10 bis 19. Er trinkt Glas 19. Die dritte Runde besteht aus Glas 20 bis 29, und Glas 29 wird getrunken. In jeder Runde kommen die Gläser $10n$ bis $10n+9$ hinzu und das Glas $10n + 9$ muss dran glauben. Offensichtlich werden die ersten neun Gläser jeder Runde von Lauterbach nie wieder angerührt. Um Mitternacht findet sie dann der Wirt.

Dieses Beispiel macht im Übrigen deutlich, wie eminent wichtig die Ordnung ist. Schlechte Ordnungen können dazu führen, dass wir unsere Aufgabe gar nicht erledigen können. Das Paradox bei Kunze und Lauterbach ist aber, dass sie in endlicher Zeit immer gleichauf sind: nach der $n$ten Runde befinden sich $9n$ volle Gläser auf dem Tisch. Nach unendlich vielen Schritten aber offenbart sich ein schwerwiegender Unterschied: Kunze trinkt die Gläser in der Reihe, wie sie ankommen, Lauterbach entgegen dieser Reihenfolge. Das erste Prinzip heißt Englisch **first in first out** (FIFO), das zweite **last in first out** (LIFO).

Doch zurück zu unserer Rekursion. Um diese Beispiel formal zu diskutieren, definieren wir zunächst die Funktionen $F_k, F_\ell : \omega \times \wp(\omega) \to \wp(\omega)$. Diese sagen uns, was in jedem Einzelschritt passiert.

$$\begin{array}{rcl} F_k(n, M) & := & M \cup \{10(n+1), \dots, 10(n+1)+9\} - \{n\} \\ F_\ell(n, M) & := & M \cup \{10(n+1), \dots, 10(n+1)+9\} - \{10n+9\} \end{array}$$

F_k ist diejenige Funktion, welche die nächsten zehn Zahlen hintanhängt und die erste streicht. F_ℓ ist diejenige Funktion, welche die letzte Zahl streicht und hinten die nächsten zehn Zahlen anhängt. So bekommen wir die Funktionen $f_k \restriction \omega, f_\ell \restriction \omega : \omega \to \wp(\omega)$, wenn wir noch verlangen, dass $f_k(0) = f_\ell(0) = \{0, 1, \dots, 9\}$. Damit können wir genau beschreiben, was in jedem endlichen Schritt (also ab elf und vor Punkt zwölf) passiert. Wenn es zwölf Uhr ist, ist der erste Limesschritt getan. Wie aber bestimmen wir $f_k(\omega)$ und $f_\ell(\omega)$? Hier ist die folgende Definition. Wir sagen für Limeszahlen μ:

$$H(f \restriction \mu) := \{k : \text{ für fast alle } \kappa < \mu : k \in f(\kappa)\}$$

Hierbei heißt 'fast alle': es gibt nur endlich viele Ausnahmen. Im Klartext: wir nehmen an, ein Glas ist um zwölf Uhr auf dem Tisch von Kunze bzw. Lauterbach, wenn sie zu fast jeder Runde vorher auf dem Tisch war. Nun setzen wir

$$f_k(\omega) := H(f_k \restriction \omega)$$

Da jedes Glas von Kunze tatsächlich getrunken wird und dann unweigerlich vom Tisch verschwindet, ist $f_k(\omega) = \varnothing$. Bei Lauterbach aber verschwinden nur die Gläser mit der Nummer $10n+9$, alle anderen bleiben auf dem Tisch. Also ist

$$f_\ell(\omega) := H(f_\ell \restriction \omega) = \omega - \{10n + 9 : n \in \omega\}$$

Wiederum ist es so, dass bei der Rekursion die dritte Klausel schon die Hauptlast der Rekursion trägt, sodass wir wie folgt formulieren können.

> Wertverlaufsrekursion. Es sei λ eine Ordinalzahl, $H(-) : \wp(\lambda \times M) \to M$ eine Abbildung, welche als Argument Relationen von λ nach M nimmt und als Wert ein Element aus M ausgibt. Dann existiert eine und nur eine Abbildung $f : \lambda \to M$, welche für alle $\mu < \lambda$ die folgende Bedingung erfüllt.
>
> $$f(\mu) = H(f \restriction \mu) \ .$$

Es ist eine Erkenntnis von Gödel, dass man für natürliche Zahlen (das heißt für $\lambda = \omega$) jede Funktion, welche sich durch Wertverlaufsrekursion definieren lässt, auch schon durch primitive Rekursion definieren kann. Die Umkehrung gilt übrigens auch: Wertverlaufsrekursion ist mindestens so stark wie primitive Rekursion.

Die Ordnungsinduktion sowie die Wertverlaufsrekursion können auch für beliebige fundierte Ordnungen benutzt werden.

> Allgemeine Ordnungsinduktion. Es sei R eine fundierte Ordnung auf einer Menge M. Genau dann gilt eine Eigenschaft P von Elementen aus M für ganz M, wenn für jedes $x \in M$ gilt: $P(x)$ gilt dann, wenn $P(y)$ für alle $y \, R \, x$.

Zum Beweis nehmen wir wieder an, dies sei nicht so. Dann existiert ein bezüglich R kleinstes x, auf das P nicht zutrifft. Dann ist $P(y)$ für alle $y \, R \, x$, nach Wahl von x. Also $P(x)$, nach Annahme über P. Widerspruch.

Analog dazu gibt es die

> Allgemeine Ordnungsrekursion. Es sei R eine fundierte Ordnung auf einer Menge N. Ferner sei $H : M \times \wp(N \times M) \to M$ eine beliebige Funktion. Dann existiert genau eine Funktion $f : N \to M$ mit
>
> $$f(x) = H(x, f \restriction \{y : y \; R \; x\}) \ .$$

Man beachte, dass wir im Gegensatz zur Ordnungsrekursion über Ordinalzahlen nunmehr x als zusätzliches Argument haben. Bei der ersteren kann man sich das Argument μ aus dem Abschnitt $f \restriction \mu$ besorgen. Es ist nämlich

$$\mu = \{\kappa : \text{ es existiert } x \in M : \langle \kappa, x \rangle \in f \restriction \mu\}$$

Insofern kann man dort auf das zusätzliche Argument verzichten. In beliebigen Ordnungen ist dies natürlich nicht mehr gegeben.

Wir haben hierbei noch nicht einmal die Transitivität benötigt. Es war lediglich die Fundiertheit entscheidend. Trotzdem ist der transitive Fall der am meisten gebrauchte. Er erlaubt uns im Übrigen endlich den Beweis zu erbringen, dass es zu jeder wohlgeordneten Menge $\underline{M}$ auch eine Ordinalzahl gibt, die (als wohlgeordnete Menge aufgefasst) isomorph zu $\underline{M}$ ist. Es sei $<$ eine transitive, fundierte Ordnung auf M. Wir definieren die Höhe eines Elements wie folgt.

$$h_<(x) = \{h_<(y) : y < x\}$$

Die Funktion $h_<$ ist durch Allgemeine Ordnungsrekursion gewonnen worden. Hierbei ist M eine genügend große Menge (die genaue Definition von M ersparen wir uns hier), und für alle $Q \subseteq N \times M$ und alle $x \in M$ setzen wir

$$H(x, Q) := \{y : \text{ es gibt } x \in N : \langle x, y\rangle \in Q\}$$

Man rechnet leicht nach, dass

$$h_<(x) = H(x, h_< \restriction \{y : y < x\}) \ .$$

Wir wenden nun erstmals die allgemeine Ordnungsinduktion an, um zu zeigen, dass die Höhe eines Elements immer eine Ordinalzahl ist.

Proposition 18 *Falls $< \subseteq M^2$ transitiv und fundiert ist, so ist für alle $x \in M$ $h_<(x)$ eine Ordinalzahl.*

Beweis. Wir wenden die Ordnungsinduktion an. Wir zeigen: ist $h_<(y)$ für alle $y < x$ eine Ordinalzahl, so auch $h_<(x)$. Es ist $h_<(x) = \{h_<(y) : y < x\}$, und nach Voraussetzung ist $h_<(y)$ für alle $y < x$ eine Ordinalzahl. Also ist $h_<(x)$ eine Menge von Ordinalzahlen. Die Relation $\in$ ist darauf linear und fundiert. Was zu zeigen bleibt, ist, dass $h_<(x)$ transitiv bezüglich $\in$ ist. Dazu sei $\kappa \in h_<(x)$, etwa $\kappa = h_<(y)$. Ferner sei $\lambda < \kappa$. Dann ist $\lambda \in h_<(y)$, woraus folgt, dass ein $z < y$ existiert mit $\lambda = h_<(z)$. Nun ist $z < y < x$, also $z < x$, da $<$ transitiv ist. Daher ist $\lambda \in h_<(x)$. Q. E. D.

Daraus folgt mit dem Wohlordnungssatz nun folgendes Ergebnis.

Satz 19 *Zu jeder Menge M existiert eine Ordinalzahl κ und eine bijektive Abbildung $f : \kappa \to M$.*

Beweis. Zunächst existiert eine Wohlordnung $<$ auf M. Betrachten wir nun die Zuordnung $x \mapsto h_<(x)$. Diese ordnet jedem Element eine Ordinalzahl zu. Sei $\kappa := \{h_<(x) : x \in M\}$. Dies ist eine Ordinalzahl, wie oben gesehen. $h_<$ ist surjektiv nach Konstruktion. Sie ist aber auch injektiv. Denn sei $h_<(x) = h_<(y)$. Da $<$ linear ist, gilt $x < y$, $x = y$ oder $x > y$. Ist aber $x < y$, so gilt $h_<(x) \in h_<(y)$, und ist $x > y$, so ist $h_<(y) \in h_<(x)$. Beides kann nicht gelten. Also ist tatsächlich $x = y$. $h_<$ ist folglich bijektiv, und wir setzen $f := h_<^{\smile}$. Dies ist die gewünschte Bijektion. Q. E. D.

4. Teil: Mengenlehre IV: Kardinal- und Ordinalzahlen

In diesem Teil wollen wir etwas tiefer die Beziehung zwischen Ordnung und Anzahl beleuchten. Zunächst aber betrachten wir den Unterschied zwischen endlichen und unendlichen Mengen. Das Hilbert Hotel hat unendlich viele Zimmer, mit den Nummern 0, 1, 2, und so weiter. Die Ordnungsnummern der Zimmer sind also gerade die natürlichen Zahlen. Am ersten Abend kommt ein unendlicher Bus. Der Hotelchef füllt das Hotel mit ihnen. Gast Nummer n bekommt Zimmer Nummer n. Spät abends kommt noch ein einzelner Gast und bittet um ein Zimmer. Das Hotel ist ausgebucht. Der Chef aber weckt alle Gäste, lässt sie ins nächste Zimmer umziehen und quartiert den neuen Gast ins Zimmer 0. Am nächsten Abend kommt ein Bus mit unendlich vielen Reisenden. Der Chef überlegt nicht lange, bittet alle alten Gäste aus dem Zimmer n nunmehr in die Nummer $2n$, und quartiert die neuen Gäste in die Zimmer mit ungeraden Nummern ein. Am dritten Abend schließlich kommt eine unendlich lange Kolonne von Bussen mit unendlich vielen Reisenden. Auch kein Problem, sagt der Chef. Er bittet Gast n aus Bus m in Zimmer $2^m 3^n$ und hat immer noch unendlich viele Zimmer frei, die er den alten Gästen überlässt.

Was man daraus lernt ist, dass die Kombinatorik unendlicher Mengen gänzlich anders als die der endlichen ist. Ist ein endliches Hotel voll, so ist es voll, da passt keiner mehr hinein. Unendliche Mengen lassen sich aber injektiv in einen echten Teil von sich abbilden. Falls die Größe einer Menge die Anzahl ihrer Elemente ist, so gelten hier offensichtlich ganz und gar merkwürdige Gesetze. Es sei $\aleph_0$ die Größe der Menge ω. Wie wir oben gesehen haben, gilt $\aleph_0 + \aleph_0 = \aleph_0$, da wir die Menge zweimal in sich packen können. Ferner gilt

$\aleph_0 \cdot \aleph_0 = \aleph_0$, da wir die Menge so oft in sich packen können, wie sie selbst Elemente hat.

Um das Rechnen mit Anzahlen zu verstehen, muss man sich erst mit Ordinalzahlen vertraut machen. Zunächst einmal verschaffen wir uns einen Überblick über die ersten unendlichen Ordinalzahlen. Wir haben ja schon die endlichen kennengelernt. Die erste unendliche ist ω. Nach den Gesetzen der Mengenlehre kann man nun zu jeder Ordinalzahl κ eine nächstgrößere finden, nämlich $\kappa \cup \{\kappa\}$. Wir nennen nun eine **Zählreihe** eine unendliche Folge von aufeinanderfolgenden Ordinalzahlen, deren erstes Glied keine Nachfolgerzahl ist. Die erste Zählreihe ist diejenige, die mit 0 beginnt. Sie enthält genau die natürlichen Zahlen.

$$0, 1, 2, 3, \ldots$$

Am Ende jeder Zählreihe steht nun wieder eine Limeszahl. Sie ist die Menge ihrer Vorgänger. Die zweite Zählreihe ist

$$\omega, \omega + 1, \omega + 2, \omega + 3, \ldots$$

An ihrem Ende steht die Zahl, welche wir $\omega+\omega$ nennen. Darauf folgt die dritte Zählreihe, an deren Ende wir $\omega + \omega + \omega$ setzen. Wie erklärt man nun diese Addition unendlicher Ordinalzahlen? Anschaulich ist $\kappa + \lambda$ die Ordinalzahl jener Ordnung, welche durch Anhängen von λ an κ entsteht. Um dies formal zu definieren, verwenden wir das Schema der transfiniten Rekursion.

$$\begin{array}{lcll} \kappa + 0 & := & \kappa & \\ \kappa + (\mu + 1) & := & (\kappa + \mu) + 1 & \\ \kappa + \lambda & := & \bigcup_{\mu<\lambda}(\kappa + \mu) & \text{falls } \lambda \text{ Limeszahl} \end{array}$$

Wir schreiben anstelle von $\omega + \omega$ auch $\omega \cdot 2$. Ebenso schreiben wir $\omega \cdot 3$ für $\omega + \omega + \omega$. Der zweite Faktor bestimmt also, wie oft wir die erste Ordinalzahl zu sich selbst addieren müssen. Auch dies können wir durch transfinite Rekursion definieren.

$$\begin{array}{lcl} \kappa \cdot 0 & := & 0 \\ \kappa \cdot (\mu + 1) & := & \kappa \cdot \mu + \kappa \\ \kappa \cdot \lambda & := & \bigcup_{\mu<\lambda}(\kappa \cdot \mu) \end{array}$$

Betrachten wir nun folgende Menge von Ordinalzahlen.

$$\{0, 1, 2, \ldots, \omega, \omega{+}1, \omega{+}2, \ldots, \omega{\cdot}2, \omega{\cdot}2{+}1, \omega{\cdot}2{+}2, \ldots, \omega{\cdot}3, \omega{\cdot}3{+}1, \omega{\cdot}3{+}2, \ldots, \ldots\}$$

Diese ist wiederum eine Ordinalzahl, nämlich $\omega \cdot \omega$. Diese nennen wir nun auch ω^2, ähnlich wie beim Potenzieren mit endlichen Zahlen. So hat man also das Multiplizieren definiert und kann schließlich sogar das Potenzieren definieren.

$$\begin{array}{lcl} \kappa^0 & := & 1 \\ \kappa^{\lambda+1} & := & \kappa^\lambda \cdot \kappa \\ \kappa^\mu & := & \bigcup_{\lambda<\mu} \kappa^\lambda \end{array}$$

So bekommen wir also die Ordinalzahlen ω^3, ω^4, ω^ω und so weiter. Diese sind Mengen. Aber sind sie wirklich größer als ω? Enthalten sie mehr Elemente? Die Antwort ist überraschenderweise 'nein'. Alle diese Mengen sind genauso groß wie ω. Um überhaupt von Größe reden zu können, sei dies jetzt formal definiert.

Definition 20 *Es seien M und N Mengen. Wir schreiben $|M| \leq |N|$ und sagen, N sei* ***mindestens so mächtig wie*** *M, falls eine injektive Abbildung $f : M \to N$ existiert. Gilt nicht auch $|N| \leq |M|$, so heißt N* ***mächtiger als*** *M. Ferner schreiben wir $|M| = |N|$, falls es eine Bijektion von M nach N gibt, $|M| < |N|$, falls $|M| \leq |N|$ aber nicht $|M| = |N|$.*

Zum Beispiel ist die leere Funktion eine injektive Abbildung von der leeren Menge in jede beliebige Menge. Also gilt $|\varnothing| \leq |M|$ für jedes M. Ist ferner $M \subseteq N$, so ist die Identität eine injektive Abbildung von M nach N, weswegen $|M| \leq |N|$.

Satz 21 (Cantor) *Es seien M und N Mengen. Dann ist entweder $|M| < |N|$ oder $|M| = |N|$ oder $|M| > |N|$.*

Also ist die Mächtigkeit einer Menge tatsächlich ein lineares Maß. Die Idee ist nun, als Maß für die Mächtigkeit wieder eine Menge zu nehmen, gewissermaßen einen Vertreter der Klasse aller gleichmächtigen Mengen. Hier bieten sich die Ordinalzahlen an. Zunächst einmal existiert nämlich auf jeder Menge eine Wohlordnung, und zu jeder Wohlordnung eine Ordinalzahl, die zu ihr isomorph ist. Das bedeutet aber nichts anderes, als dass zu jeder Menge M eine Ordinalzahl κ existiert mit $|\kappa| = |M|$ (siehe Satz 19). Dabei gilt es aber zu bedenken, daß nicht alle Ordinalzahlen verschieden mächtig sind. Wir haben nämlich zum Beispiel $|\omega + 1| = |\omega|$. Deswegen nehmen wir einfach die kleinste in ihrer Mächtigkeitsklasse. Diese existiert und ist sogar eindeutig bestimmt.

Definition 22 *Eine Ordinalzahl κ heißt* ***Kardinalzahl****, falls sie mächtiger ist als jede ihrer Vorgänger.*

Wir schreiben $\alpha <_k \beta$, falls $|\alpha| < |\beta|$ gilt. Wir zeigen zunächst mal, dass $\alpha <_k \beta$ genau dann gilt, wenn $\alpha < \beta$ (als Ordinalzahle). Damit erübrigt sich natürlich der Unterschied zwischen $<_k$ und $<$. Sei also $\alpha <_k \beta$. Es ist $\alpha < \beta$, $\alpha = \beta$ oder $\beta < \alpha$, da diese ja Ordinalzahlen sind. Aber aus $\beta \leq \alpha$ folgt $\beta \subseteq \alpha$, woraus wiederum $|\beta| \leq |\alpha|$ folgt. Also $\beta \leq_k \alpha$, im Widerspruch zur Annahme. Nun sei umgekehrt $\alpha < \beta$. Dann gilt $\alpha \subseteq \beta$, woraus $|\alpha| \leq |\beta|$ folgt und somit $\alpha \leq_k \beta$. Wäre nun aber $|\alpha| = |\beta|$, so wäre α keine Kardinalzahl, da ja $\beta < \alpha$ (hier benötigen wir diese Annahme zum ersten Mal). Also $\alpha <_k \beta$. Ferner erinnern wir daran, dass $\alpha < \beta$ genau dann, wenn $\alpha \in \beta$.

Proposition 23

1. *Jede endliche Kardinalzahl ist eine Ordinalzahl.*
2. *Es seien α und β Kardinalzahlen. Dann gilt $\alpha < \beta$ genau dann, wenn $\alpha \in \beta$.*
3. *Jede Menge von Kardinalzahlen hat ein kleinstes Element.*

Die dritte Behauptung folgt so: jede Menge von Kardinalzahlen ist eine Menge von Ordinalzahle, und diese hat ein kleinstes Element, welches auch als Kardinalzahl minimal in der Menge ist. Die Kardinalzahlen sind also, wie die Ordinalzahlen, durch $<$ (d.h. $\in$) wohlgeordnet. Die endlichen Kardinalzahlen sind genau die endlichen Ordinalzahlen. Die erste unendliche Kardinalzahl ist ω. Sie heißt auch $\aleph_0$ insofern, als sie eine Kardinalzahl ist. Die nächste Kardinalzahl heißt $\aleph_1$. Von ihrer Existenz werden wir uns weiter unten vergewissern. Diese Notation muß erklärt werden. Man kann die Kardinalzahlen mit Hilfe von Ordinalzahlen durchnummerieren. Ist α eine Ordinalzahl, so ist $\aleph_\alpha$ die αte Kardinalzahl (sofern sie existiert). Wir bezeichnen generell mit κ^+ die kleinste Kardinalzahl, welche größer ist als κ. Wir werden zeigen, dass sie existiert. Damit definieren wir induktiv wie folgt.

$$\begin{array}{lcl} \aleph_0 & := & \omega, \\ \aleph_{\alpha+1} & := & \aleph_\alpha^+, \\ \aleph_\mu & := & \bigcup_{\lambda<\mu} \aleph_\lambda \quad \mu \text{ Limeszahl.} \end{array}$$

Das Rechnen mit Kardinalzahlen ist ganz anders als das Rechnen mit Ordinalzahlen. Wir definieren zunächst einmal Summe und Produkt. Haben wir zwei elementefremde Mengen M und N, so ist ihre Vereinigungsmenge so mächtig wie die Summe von $|M|$ und $|N|$. Sind M und N beliebig (also nicht notwendig elementefremd), so ist $M \times \{0\}$ elementefremd zu $N \times \{1\}$ und es ist $M \times \{0\}$ gleichmächtig zu M und $N \times \{1\}$ gleichmächtig zu N. Ferner setzen wir die Mächtigkeit von $M \times N$ als das Produkt der Mächtigkeiten von M und von N.

$$\begin{array}{lll} \alpha + \beta & := & |\alpha \times \{0\} \cup \beta \times \{1\}| \\ \alpha \cdot \beta & := & |\alpha \times \beta| \end{array}$$

Es gilt nun

Satz 24 *Es seien α und β unendliche Kardinalzahlen. Dann ist $\alpha + \beta = \alpha \cdot \beta = \max\{\alpha, \beta\}$. Dies gilt auch, wenn α oder β aber nicht beide zugleich endlich sind.*

Sei endlich ${}^M N$ die Menge der Funktionen von M nach N. Diese ist eine Teilmenge von $\wp(M \times N)$, und damit existiert sie immer. Wir setzen

$$\alpha^\beta := |{}^\beta \alpha|$$

Insbesondere ist für $\alpha = 2$

$$2^\alpha = |\wp(\alpha)| \, .$$

Denn es existiert eine Bijektion zwischen den Teilmengen einer Menge M und den Funktionen von M nach $2 = \{0, 1\}$. (Ist $A \subseteq M$, setze $\chi_A : M \to 2$ mit $\chi_A(x) = 1$ falls $x \in A$ und $\chi_A(x) = 0$ sonst. Diese heißt auch die **charakteristische Funktion von** A.) Bisher hat es so ausgesehen, als gäbe es nur abzählbare Ordinalzahlen. Dass aber auch die Menge der Kardinalzahlen nicht beschränkt ist, hat schon Cantor gezeigt.

Satz 25 (Cantor) *$2^\alpha > \alpha$.*

Beweis. Angenommen, wir haben eine Abbildung $f : \alpha \to \wp(\alpha)$. Setze $H := \{x \in \alpha : x \notin f(x)\}$. Wir behaupten, dass es kein $y \in \alpha$ gibt mit $f(y) = H$. Denn sei dem so. Dann ist entweder $y \in f(y)$ oder $y \notin f(y)$. Angenommen, $y \in f(y)$. Dann ist nach Definition $y \notin H = f(y)$. Falls aber $y \notin f(y)$, so $y \in H = f(y)$. Widerspruch. Q. E. D.

Wir besitzen damit einen Beweis für die Existenz von α^+. Denn es sei

$$M := \{\beta : \beta \text{ Kardinalzahl}, \alpha \in \beta, \beta \in 2^\alpha\} \, .$$

Dann ist M eine Menge von Kardinalzahlen und hat ein kleinstes Element. Dies ist das gesuchte α^+.

Aus dem Satz 25 folgt im Übrigen, dass es die Menge V aller Mengen nicht geben kann, wie Cantor schon bemerkt hat. Denn falls es sie gibt, so hat sie eine Mächtigkeit. Nun existiert auch $\wp(V)$, und hat eine Mächtigkeit, welche größer ist als die von V. Dies aber kann nicht sein. Denn jede Menge M ist in V enthalten, weshalb wir auch $|M| \leq |V|$, insbesondere also $|\wp(V)| \leq |V|$ haben müssen. Ebenso kann man sehen, dass die Ordinalzahlen keine Menge bilden. Angenommen, dies wäre so. Diese Menge, nennen wir sie O, ist eine Ordinalzahl, wie man leicht nachrechnet. Dann hätte sie aber auch einen Nachfolger, der aber nicht in O ist. Widerspruch. Man kann im Übrigen auch zeigen, dass die Kardinalzahlen keine Menge bilden.

Betrachten wir zum Schluss die reellen Zahlen. Wir wollen zeigen, dass es genauso viele reelle Zahlen gibt wie es Mengen natürlicher Zahlen gibt. Um die Lage vorab zu vereinfachen, betrachten wir zunächst reelle Zahlen im Intervall $[0, 1[$. Eine solche reelle Zahl kann als eine unendliche Binärfolge kodiert werden, nämlich die Folge ihrer Nachkommaziffern. Üblicherweise notiert man nur endlich viele, wenn nach ihnen nur noch Nullen folgen, aber dies ist unerheblich. Für unsere Zwecke nehmen wir nur die unendlichen Folgen. Jede Zahl in diesem Intervall hat vor dem Komma eine 0, sodass diese entbehrlich ist. Der Zahl

$$0,1101011001\ldots$$

ordnen wir also die Folge

$$1101011001\ldots$$

zu. Einer unendlichen Binärfolge $\mathfrak{x} = \langle x_i : i \in \omega\rangle$ ordnen wir die Menge $M(\mathfrak{x}) := \{i : x_i = 1\}$ zu. (Diese heißt im Übrigen auch der **Träger** von $\mathfrak{x}$.) Somit kann man jeder reellen Zahl im Intervall $[0, 1[$ eine Teilmenge von ω zuordnen. Offenbar ist diese Abbildung surjektiv: zu jeder Menge $M \subseteq \omega$ existiert eine Folge $\mathfrak{x}$, deren Träger diese Menge ist, und zu dieser wiederum bekommen wir die Zahl

$$r_M := \sum_{i \in M} 2^{-i}$$

Damit haben wir schon $|[0, 1[| \leq |\wp(\omega)|$. Nun existiert eine Bijektion von $\mathbb{R}$ nach $[0, 1[$. Ist $r = k + z$, $k \in \mathbb{N}$, $z \in [0, 1[$, so sei $h(r) := 1 - 2^{-k+1} + 2^{-k-2} \cdot z$. Und es sei $h(-r) := 1/2 - h(r)$. Dies ist eine ordnungstreue Bijektion von $\mathbb{R}$ auf $]0, 1[$, wie man nachrechnen kann. Also gilt $|\mathbb{R}| = |]0, 1[| \leq |[0, 1[|$.

Nun fehlt noch der Nachweis, dass $|\wp(\omega)| \leq |[0,1[|$. Dazu fehlt noch ein kleines Stück. Denn es stellt sich heraus, dass nicht alle Mengen verschiedenen reellen Zahlen entsprechen. Es ist zum Beispiel $0,0\overline{1} = 0,01111 = 0,1$, also entspricht der Folge $\langle 0,1,1,\ldots\rangle$ dieselbe Zahl wie der Folge $\langle 1,0,0,0,\ldots\rangle$. Im allgemeinen gilt $r_M = r_N$ für verschiedene M und N nur dann, falls gilt: M ist endlich und $N = (M - \max M) \cup \{k : k > \max M\}$, oder es ist N endlich und $M = (N - \max N) \cup \{k : k > \max N\}$. Damit bekommen wir sofort $|\wp(\omega)| \leq 2 \cdot |[0,1[|$, da wir jeder reellen Zahl höchstens zwei Folgen zuordnen. Nun gilt auch $2 \cdot |[0,1[| = |2 \times [0,1[| \leq |[0,1[|$. (Dies folgt aus den oben besprochenen Rechengesetzen für Kardinalzahlen, aber wir werden es hier direkt beweisen.) Die Zuordnung $\langle 0,r\rangle \mapsto r/2$, $\langle 1,r\rangle \mapsto (1+r)/2$ ist eine bijektive Abbildung von $2 \times [0,1[$ nach $[0,1[$. Dies vollendet den Beweis der Behauptung.

Satz 26 (Cantor) $|\mathbb{R}| = |\wp(\omega)| = 2^{\aleph_0}$.

Wir haben $2^{\aleph_0} > \aleph_0$. Aber ist $2^{\aleph_0}$ die nächst größere Kardinalzahl hinter $\aleph_0$? Cantor glaubte ganz fest, dass dem so sei. Er nannte dies die **Kontinuumshypothese**, da die reellen Zahlen auch als das Kontinuum bezeichnet wurden. Dies war ein lange offenes und immens schwieriges Problem. Gödel konnte zeigen, dass wenn ZFC ohne die Kontinuumshypothese ein Universum hat, so hat es auch ein Universum, in dem die Kontinuumshypothese wahr ist. (Auch das ist nicht ohne Weiteres klar!) Aber zur allgemeinen Überraschung stellte sich folgendes heraus:

Satz 27 (Cohen) *Man kann nicht beweisen, dass $2^{\aleph_0} = \aleph_1$ ist. Es gibt sogar Mengenuniversen, in denen dies falsch ist.*

5. Teil: Zeichenketten, Halbgruppen und Monoide

Es sei H eine nichtleere Menge und $\cdot : H \times H \to H$ eine Operation. $\cdot$ heißt **assoziativ**, falls für alle x, y und z aus H gilt

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

Definition 28 *Ein Paar $\langle H, \cdot\rangle$, wo H eine nichtleere Menge und $\cdot$ eine zweistellige assoziative Operation auf H ist, heißt eine* ***Halbgruppe****. Ist $1 \in H$*

ein Element derart, dass für alle $x \in H$ gilt $1 \cdot x = x \cdot 1 = x$, so heißt 1 eine ***Eins****, und das Tripel $\langle H, 1, \cdot \rangle$ eine* ***Halbgruppe mit Eins*** *oder* ***Monoid****. $\langle H, \cdot \rangle$ ist* ***kommutativ*** *oder* ***abelsch****, falls für je zwei Elemente $x, y \in H$ gilt $x \cdot y = y \cdot x$. Sind $\langle H, \cdot \rangle$ ($\langle H, 1, \cdot \rangle$) und $\langle K, \odot \rangle$ ($\langle K, 1', \odot \rangle$) zwei Halbgruppen (Monoide) und $h : H \to K$, so heißt h* ***Homomorphismus*** *von Halbgruppen (Monoiden), falls für alle $x, y \in H$ gilt $h(x \cdot y) = h(x) \odot h(y)$ (bei Monoiden zusätzlich $h(1) = 1'$). Wir schreiben $h : \langle H, \cdot \rangle \to \langle K, \odot \rangle$, falls h Homomorphismus ist.*

Hier sind Beispiele von Halbgruppen und Monoiden.

Beispiel 1. $\langle \mathbb{N}, 1, \cdot \rangle$ ist ein Monoid. Die Multiplikation ist assoziativ, und $1 \cdot x = x \cdot 1 = 1$.

Beispiel 2. $\langle \mathbb{N}, 0, + \rangle$ ist ein Monoid. Man beachte, dass 0 hier technisch gesehen die Rolle der *Eins* in der Halbgruppe spielt. Dies mag verwirrend erscheinen, aber man gewöhnt sich schnell daran. Es ist $\langle \mathbb{N}, + \rangle$ eine Halbgruppe, ebenso $\langle \mathbb{N} - \{0\}, + \rangle$. Allerdings besitzt die letzte Halbgruppe keine Eins.

Beispiel 3. Betrachte eine nichtleere Menge A, welche wir ein **Alphabet** nennen. (Die Elemente von A heißen dann auch **Buchstaben**.) Es sei A^* die Menge aller endlichen Folgen über A, welche auch **Zeichenketten** oder **Worte über** A heißen. Eine endliche Folge über A ist nach den Ausführungen von Teil 2 eine Funktion von einer natürlichen Zahl nach A. So ist die Zeichenkette über $\{\mathsf{a}, \mathsf{b}, \mathsf{x}\}$, welche wir anschaulich schlicht mit baca widergeben, formal eine Funktion $\vec{x} : 4 \to \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ mit $\vec{x}(0) = \mathsf{b}$, $\vec{x}(1) = \mathsf{a}$, $\vec{x}(2) = \mathsf{c}$, $\vec{x}(3) = \mathsf{a}$. Diesen Rigorismus werden wir jedoch bald fallenlassen. Eine Zeichenkette bezeichnen wir mit einem Vektorpfeil. Ist $\vec{x} : m \to A$, so heißt m die **Länge** von $\vec{x}$ und wird mit $lg(\vec{x})$ bezeichnet. Sind $\vec{x}, \vec{y} \in A^*$, so sei $\vec{x} \cdot \vec{y}$ das Ergebnis der Anfügung von $\vec{y}$ an $\vec{x}$. Angesichts der Definitionen von Teil 2 können wir dies formal definieren. Sei $\vec{x} : m \to A$ und $\vec{y} : n \to A$, so ist $\vec{x} \cdot \vec{y} : m + n \to A$ definiert durch

$$(\vec{x} \cdot \vec{y})(j) := \begin{cases} \vec{x}(j), & \text{falls } j < m, \\ \vec{y}(j - m), & \text{sonst.} \end{cases}$$

Diese Operation nennt man **Konkatenation**. Das Paar $\langle A^*, \cdot \rangle$ ist eine Halbgruppe. Wir bezeichnen die leere Folge mit ε. (Es ist $\varepsilon : 0 \to A$ die eindeutig bestimmte leere Funktion.) Diese ist eine Eins, wie man leicht nachprüft. Also ist $\langle A^*, \varepsilon, \cdot \rangle$ ein Monoid. Weiter unten kommen wir auf dieses Monoid noch zu sprechen.

Beispiel 4. Sei k eine natürliche Zahl, A eine nichtleere Menge. Es sei $L(A,k)$ die Menge der Folgen der Länge $\leq k$ über A. Es bezeichne $prf_k(\vec{x})$ das Anfangsstück von $\vec{x}$ der Länge k, falls $\vec{x}$ mindestens die Länge k hat, und $\vec{x}$ sonst. Nun definiere die Operation $\smile$ durch $\vec{x}\smile\vec{y} := prf_k(\vec{x}\cdot\vec{y})$. Wir wählen $A = \{\mathtt{a}, \mathtt{b}\}$ und $k = 2$.

$\smile$	ε	a	b	aa	ab	ba	bb
ε	ε	a	b	aa	ab	ba	bb
a	a	aa	ab	aa	aa	ab	ab
b	b	ba	bb	ba	ba	bb	bb
aa	aa	aa	aa	aa	aa	aa	aa
ab	ab	ab	ab	ab	ab	ab	ab
ba	ba	ba	ba	ba	ba	ba	ba
bb	bb	bb	bb	bb	bb	bb	bb

Die Halbgruppen bzw. Monoide von Beispiel 1 und 2 sind abelsch, die aus 3 und 4 sind es nicht, wie man leicht nachrechnet. Ein Homomorphismus $h : \mathfrak{G} \to \mathfrak{H}$ heißt **Endomorphismus**, falls $\mathfrak{G} = \mathfrak{H}$, **Isomorphismus**, falls h bijektiv ist, und **Automorphismus**, falls h Isomorphimus und Endomorphismus ist.

Die Multiplikation von natürlichen Zahlen ist assoziativ. Man macht von dieser Tatsache Gebrauch, wenn man bei Produkten auf die Klammern verzichtet, indem man zB schreibt $7 \cdot 43 \cdot 2 \cdot 8$. Streng genommen bedeutet Assoziativität aber lediglich, dass man bei *drei* Multiplikanden auf die Klammern verzichten kann. Es lässt sich aber zeigen, dass daraus bereits folgt, dass ein Produkt aus beliebig vielen Zahlen stets das gleiche Ergebnis liefert, egal wie es geklammert wird. Wir zeigen dies hier allgemein für Halbgruppen. Ist $\langle H, \cdot \rangle$ eine Halbgruppe, so bilden wir wie folgt Terme über H. Ein Element $x \in H$ ist ein Term; sind s, t Terme, so auch $(s \cdot t)$. Zu einem Term assoziieren wir eine Zeichenkette $\kappa(t) \in H^*$ wie folgt. Für $x \in H$ ist $\kappa(x) = x$; ansonsten ist $\kappa((s \cdot t)) = \kappa(s) \cdot \kappa(t)$. Mit anderen Worten, wir vergessen die Klammern. Terme kann man 'ausrechnen'. Auf diese Weise bestimmt jeder Term ein Element aus H, das wir den **Wert** des Terms nennen. Der Wert von $x \in H$ ist schlicht x; der Wert von $(s \cdot t)$ ist das Produkt $u \cdot v$, wo u der Wert von s und v der Wert von t ist. So ist der Wert von $((2 \cdot 4) \cdot 7) = (8 \cdot 7) = 56$.

Hilfssatz 29 *Es sei $\langle H, \cdot \rangle$ eine Halbgruppe. Es seien s und t Terme über H. Falls $\kappa(s) = \kappa(t)$, so haben s und t denselben Wert.*

Beweis. Wir zeigen: mit Hilfe das Assoziativgesetzes lässt sich jeder Term nach links klammern. Dies geschieht natürlich unter Beibehaltung des Werts; denn die Operation ist ja assoziativ auf H. Dabei heißt t **nach links geklammert**, falls $t = h \in H$ oder $t = (s \cdot h)$ für ein $h \in H$. Analog definiert man **nach rechts geklammert**. Die Behauptung wird induktiv über die Länge der Zeichenkette gezeigt. Für Terme mit Zeichenketten der Länge ≤ 2 ist nichts zu zeigen. Sei $s = (t \cdot u)$ ein Term und gelte die Behauptung für alle Terme kleinerer Länge als s. Ist $u = h \in H$, so ist der Term schon nach links geklammert. Sei dies also nicht der Fall. Dann lässt sich t nach links klammern zu einem Term t' und u lässt sich nach rechts klammern zu einem Term $h \cdot u'$. Es ist also s umformbar in folgenden Term

$$(t' \cdot (h \cdot u'))$$

Einmalige Anwendung des Assoziativgesetzes erbringt

$$((t' \cdot h) \cdot u') \ ,$$

also wieder ein linksgeklammerter Term gefolgt von einem rechtsgeklammerten Term. Ist u' noch nicht von der Form $h \in H$, so wiederholt man diese Operation. Auf diese Weise erhält man schließlich einen linksgeklammerten Term. Analog kann man einen rechtsgeklammerten Term bekommen. Q. E. D.

Wir definieren nun folgendes allgemeines **Produkt**.

$$\begin{array}{lcl} \prod_{i<0} x_i & := & 1 \\ \prod_{i<k+1} x_i & := & (\prod_{i<k} x_i) \cdot x_k \end{array}$$

Dies ist immer definiert für $k > 0$, und für $k = 0$ nur dann, wenn es eine Eins gibt. (Es kann nur eine Eins bezüglich derselben Operation geben. Denn sind 1 und $1'$ Einsen bezüglich $\cdot$, so gilt $1 = 1 \cdot 1' = 1'$.) Der eben gezeigte Satz beweist folgende allgemeine Produktformel:

$$\prod_{i<m} x_i = \prod_{i<k} x_i \cdot \prod_{k \leq i < m} x_i$$

Definition 30 *Es sei $\mathfrak{M} = \langle M, 1, \cdot \rangle$ ein Monoid und $X \subseteq M$. Ist $1 \in X$ und X abgeschlossen unter $\cdot$ (das heißt $x \cdot y \in X$ für alle $x, y \in X$) so heißt $\langle X, 1, \cdot_X \rangle$ ein **Untermonoid** von $\mathfrak{M}$. Hierbei ist $\cdot_X$ die Einschränkung von $\cdot$ auf X, welche normalerweise auch mit $\cdot$ bezeichnet wird. X **erzeugt** $\mathfrak{M}$, falls das kleinste X enthaltende Untermonoid von $\mathfrak{M}$ genau $\mathfrak{M}$ selbst ist.*

Wir nehmen etwa das Monoid $\langle \mathbb{N}, 1, \cdot \rangle$. Die Menge der Zahlen, welche durch eine gegebene Zahl n (etwa 14) teilbar sind, bilden zusammen mit der 1 ein Untermonoid. Denn sind k_1 und k_2 durch 14 teilbar, so auch $k_1 \cdot k_2$, ebenso wenn $k_1 = 1$ oder $k_2 = 1$. Ist $k_1 = k_2 = 1$, so ist das Produkt $= 1$. Ebenso bilden die Worte über A der Länge $\geq n$, n gegeben, zusammen mit ε ein Untermonoid von $\langle A^*, \varepsilon, \cdot \rangle$.

Lemma 31 *Es sei $\mathfrak{M} = \langle M, 1, \cdot \rangle$ ein Monoid und $X \subseteq M$. Es bezeichne $\langle X \rangle$ die kleinste, X enthaltende Teilmenge von M, welche unter $\cdot$ abgeschlossen ist und 1 enthält. Dann gilt*

$$\langle X \rangle = \{\prod_{i<k} x_i : k \in \omega, \text{ für alle } i < k : x_i \in X\}$$

Beweis. Zunächst einmal ist mit $k = 0$ auch die Eins in dieser Menge. Angesichts der oben gezeigten Produktformel bekommen wir, dass mit $\prod_{i<k} x_i$ und $\prod_{i<m} y_i$ auch das Produkt $\prod_{i<k} x_i \cdot \prod_{i<m} y_i$ von der Form $\prod_{i<k+m} z_i$ ist, mit $z_i = x_i$ für $i < k$ und $z_i = y_{i-k}$ sonst. Also ist diese Menge unter Produkten abgeschlossen. Sie ist sicher auch die kleinste Menge dieser Art, welche X und 1 enthält. Q. E. D.

Wir benutzen diese Tatsache für folgenden Satz.

Hilfssatz 32 *Es seien $\mathfrak{M}$ und $\mathfrak{N}$ Monoide und $\mathfrak{M}$ sei von X erzeugt. Sei $v : X \to N$ irgendeine Abbildung. Dann existiert höchstens ein Homomorphismus $h : \mathfrak{M} \to \mathfrak{N}$ mit $h \restriction X = v$.*

Beweis. Sei $\mathfrak{M} = \langle M, 1, \odot \rangle$ und $\mathfrak{N} = \langle N, 1', \odot' \rangle$. Es seien $h, k : \mathfrak{M} \to \mathfrak{N}$ Homomorphismen und $h \restriction X = k \restriction X$. Sei $m \in M$. Ist $m = 1$, so ist $h(1) = k(1) = 1'$. Sei also $m \neq 1$. Dann existiert eine Darstellung $m = x_1 \cdot x_2 \cdot \ldots \cdot x_k$. Dann ist

$$\begin{array}{rcl} h(m) & = & h(x_1) \odot' h(x_2) \odot' \ldots \odot' h(x_k) \quad = \\ & = & k(x_1) \odot' k(x_2) \odot' \ldots \odot' k(x_k) \quad = \quad k(m) \end{array}$$

Q. E. D.

Definition 33 *Es sei $\mathfrak{M} := \langle M, 1, \cdot \rangle$ ein Monoid, und $X \subseteq M$. Wir sagen, $\mathfrak{M}$ wird als Monoid von X **frei erzeugt**, falls für jedes Monoid $\mathfrak{N} = \langle N, 1', \odot \rangle$ und jede Abbildung $v : X \to N$ genau ein Homomorphismus $h : \mathfrak{M} \to \mathfrak{N}$ existiert, für den $h \restriction X = v$. h heißt der v **fortsetzende Homomorphismus** und wird mit $\overline{v}$ bezeichnet.*

Satz 34 *Das Monoid* $\mathfrak{A} = \langle A^*, \varepsilon, \cdot \rangle$ *wird von* A *frei erzeugt.*

Beweis. Sei $\mathfrak{N} = \langle N, 1, \odot \rangle$ ein Monoid und $v : A \to N$. Es gibt wegen Hilfssatz 32 höchstens einen Homomorphismus, welcher v fortsetzt, denn $\mathfrak{A}$ wird von A erzeugt. Dieser Homomorphismus wird über die Länge der Folge definiert. Sei $\vec{x}$ von der Länge 0, Dann $\vec{x} = \varepsilon$ und so $\overline{v}(\vec{x}) = 1$. Ist $lg(\vec{x}) = 1$, so ist $\vec{x} \in A$ und $\overline{v}(\vec{x}) = v(\vec{x})$. Nun sei $lg(\vec{x}) > 1$. Dann ist $\vec{x} = \vec{y} \cdot c$ für ein $c \in A$. Setze $\overline{v}(\vec{x}) := \vec{v}(\vec{y}) \odot v(c)$. Keine andere Wahl ist möglich. Sei $\overline{v}$ auf diese Weise definiert. Es bleibt zu zeigen, dass $\overline{v}$ ein Homomorphismus ist. Dazu betrachte man $\vec{x} \cdot \vec{y}$. Dies ist von der Form $x_1 x_2 \dots x_m y_1 y_2 \dots y_n$. $\overline{v}(\vec{x} \cdot \vec{y})$ ist daher von der Form $v(x_1) \odot v(x_2) \odot \dots \odot v(x_m) \odot v(y_1) \odot v(y_2) \odot \dots \odot v(y_n)$. Da $\odot$ assoziativ ist, ist dies nichts anderes als $\overline{v}(\vec{x}) \odot \overline{v}(\vec{y})$. Q. E. D.

Satz 35 *Es seien* $\mathfrak{M} = \langle M, 1, \cdot \rangle$ *und* $\mathfrak{N} = \langle N, 1', \odot \rangle$ *Monoide und* $X \subseteq M$ *sowie* $Y \subseteq N$ *Mengen gleicher Mächtigkeit. Es sei* $\mathfrak{M}$ *von* X *und* $\mathfrak{N}$ *von* Y *frei erzeugt. Dann ist* $\mathfrak{M}$ *isomorph zu* $\mathfrak{N}$.

Beweis. Da X und Y gleichmächtig sind, existiert eine bijektive Abbildung $v : X \to Y$. Setze $w := v^{-1}$. w ist auch bijektiv. Ferner ist $v \circ w = 1_Y : Y \to Y : y \mapsto y$ sowie $w \circ v = 1_X : X \to X : x \mapsto x$. Nach Voraussetzung über $\mathfrak{N}$ existiert genau ein Homomorphimus $\overline{v \circ w} : \mathfrak{N} \to \mathfrak{N}$, der $v \circ w$ fortsetzt. Einen solchen Homomorphismus können wir angeben, nämlich die Identität auf N. Daher ist $\overline{v \circ w} = 1_N$. Ebenso ist die Identität ein Homomorphismus auf $\mathfrak{M}$ und setzt $w \circ v$ fort. Also ist $\overline{w \circ v} = 1_M$. Nun betrachten wir die Abbildung $\overline{v} \circ \overline{w}$. Diese ist ein Homomorphismus und $\overline{v} \circ \overline{w} \restriction Y = 1_Y$. Daher ist $\overline{v} \circ \overline{w} = 1_N$. Ebenso zeigt man, dass $\overline{w} \circ \overline{w} = 1_M$ ist. Also existieren bijektive Homomorphismen zwischen $\mathfrak{M}$ und $\mathfrak{N}$, die zueinander invers sind. Daher ist $\mathfrak{M} \cong \mathfrak{N}$. Q. E. D.

Dies zeigt in Verbindung mit Satz 34, dass die Monoide bestehend aus den Worten über einem Alphabet bis auf Isomorphie die einzigen frei erzeugten Monoide sind. Ferner kommt es bei dem Alphabet nur auf seine Mächtigkeit an. Kehren wir zu dem Monoid $\mathfrak{A}$ zurück. Sei $\vec{x} \in A^*$. Dann heißt $\vec{y}$ ein **Präfix** von $\vec{x}$, falls es ein $\vec{u}$ gibt mit $\vec{x} = \vec{y} \cdot \vec{u}$. $\vec{y}$ heißt **Suffix** von $\vec{x}$, falls es ein $\vec{v}$ gibt mit $\vec{x} = \vec{v} \cdot \vec{y}$; schließlich heißt $\vec{y}$ ein **Teilwort** von $\vec{x}$, falls $\vec{u}$ und $\vec{v}$ existieren mit $\vec{x} = \vec{v} \cdot \vec{y} \cdot \vec{u}$.

Definition 36 *Eine Halbgruppe hat die **eindeutige Kürzungseigenschaft**, falls jede Gleichung* $x \cdot g = h$ *sowie jede Gleicjung* $g \cdot y = h$ *eine eindeutig bestimmte Lösung für* x *und* y *hat, für jedes* $g, h \in H$.

Eine Halbgruppe mit eindeutiger Kürzungseigenschaft heißt auch **lateinisches Quadrat**. Es ist zum Beispiel $\langle A^*, \cdot\rangle$ ein lateinisches Quadrat. Die Halbgruppe aus Beispiel 4 ist dagegen kein lateinisches Quadrat. Denn es ist $\mathrm{aa}^\smile\mathrm{ab} = \mathtt{aa}^\smile\mathtt{bb}$, aber $\mathtt{ab} \neq \mathtt{bb}$.

Definition 37 *Es sei $\langle H, 1, \cdot\rangle$ ein Monoid. y heißt **linksinvers** zu x, falls $y \cdot x = 1$, und **rechtsinvers** zu x, falls $x \cdot y = 1$.*

Hilfssatz 38 *Es sei $\mathfrak{H}$ ein Monoid. Genau dann ist y linksinvers zu x, wenn x rechtsinvers zu y ist. Ist y_1 linksinvers zu x_1, y_2 linksinvers zu x_2, so ist $y_2 \cdot y_1$ linksinvers zu $x_1 \cdot x_2$.*

Satz 39 *Genau dann hat ein Monoid die eindeutige Kürzungseigenschaft, wenn es zu jedem Element eindeutig bestimmte linksinverse und eindeutig bestimmte rechtsinverse Elemente gibt.*

Definition 40 *Eine Struktur $\mathfrak{G} = \langle G, 1, ^{-1}, \cdot\rangle$ heißt **Gruppe**, falls $\langle G, 1, \cdot\rangle$ ein Monoid ist und x^{-1} sowohl linksinvers als auch rechtsinvers ist zu x.*

In einer Gruppe gelten also folgende Gesetze

$$\begin{array}{lcl} x \cdot (y \cdot z) & = & (x \cdot y) \cdot z \\ x \cdot 1 & = & x \\ 1 \cdot x & = & x \\ (x \cdot y)^{-1} & = & y^{-1} \cdot x^{-1} \\ x \cdot x^{-1} & = & 1 \\ x^{-1} \cdot x & = & 1 \end{array}$$

Das vierte Gesetz folgt unmittelbar aus der Tatsache, dass $y^{-1} \cdot x^{-1}$ linksinvers ist zu $x \cdot y$ und dass (links)inverse Elemente eindeutig sind:

Hilfssatz 41 *Jede Gruppe hat die eindeutige Kürzungseigenschaft. Insbesondere hat jedes Element genau ein linksinverses Element und genau ein rechtsinverses Element, und diese sind gleich.*

Der Beweis ist als Übung überlassen.

Hilfssatz 42 *Es sei $\mathfrak{H} = \langle H, \cdot\rangle$ ein Halbgruppe. Es besitze $\mathfrak{H}$ eine Eins, und es existiere für jedes x ein Element y mit $x \cdot y = y \cdot x = 1$. Setze dann $x^{-1} := y$. Dann ist $\langle H, 1, ^{-1}, \cdot\rangle$ eine Gruppe.*

Viele Autoren reden von einer Halbgruppe $\langle H, \cdot \rangle$ als einer *Gruppe*, falls sie die in dem Hilfssatz 42 aufgeführten Bedingungen erfüllt. Denn dann kann man ja solche Operationen hinzufügen, dass aus der Halbgruppe eine Gruppe in obigem Sinne wird. Dennoch sind die Begriffe technisch verschieden. Deshalb ist es besser, eine Halbgruppe *gruppenartig* zu nennen, wenn sie eine Eins besitzt und zu jedem Element x ein y existiert, das sowohl links– wie rechtsinvers ist zu x.

Definition 43 *Es seien $\mathfrak{M} = \langle M, 1, \cdot \rangle$ und $\mathfrak{N} = \langle N, 1', \cdot' \rangle$ Monoide. Dann ist das **Produkt** von $\mathfrak{M}$ und $\mathfrak{N}$, $\mathfrak{M} \times \mathfrak{N}$, definiert durch*

$$\mathfrak{M} \times \mathfrak{N} := \langle M \times N, 1'', \cdot'' \rangle$$

wobei $1'' := \langle 1, 1' \rangle \in M \times N$ und

$$\langle x_1, y_1 \rangle \cdot'' \langle x_2, y_2 \rangle := \langle x_1 \cdot x_2, y_1 \cdot' y_2 \rangle$$

Es ist nicht schwer zu zeigen, dass die so definierte Struktur ein Monoid ist. Analog definiert man das Produkt von Halbgruppen.

6. Teil: Aussagenlogik I: Wahrheit und Folgerung

Die Aussagenlogik befasst sich mit Aussagen. Eine Aussage ist etwas, das einen sogenannten Wahrheitswert hat. In der klassischen Logik gibt es lediglich zwei solcher Werte: 'wahr' und 'falsch'. Eine Aussage ist also entweder wahr oder falsch. Ein Befehl, eine Frage oder ein Wunsch hingegen sind keine Aussagen, da sie weder wahr noch falsch sind. Aussagen werden durch verschiedene sogenannte *Junktoren* zu komplexen Aussagen zusammengesetzt. Die gebräuchlichsten sind *nicht*, *und*, *oder*, *wenn...dann* sowie *genau dann...wenn*. In der formalen Logik ersetzt man die umgangssprachlichen Junktoren durch Symbole, in der Reihenfolge ihres Auftretens: $\neg$, $\wedge$, $\vee$, $\rightarrow$ und $\leftrightarrow$. Ihre Syntax ist um einiges einfacher als die der natürlichen Sprache. Das einstellige $\neg$ wird seinem Argument vorangestellt, die zweistelligen $\wedge$, $\vee$, $\rightarrow$ und $\leftrightarrow$ werden zwischen ihre Argument gestellt. Dies macht den Gebrauch von Klammern erforderlich. Der Wahrheitswert einer komplexen Aussage ist eindeutig durch die Wahrheitswerte seiner Teilaussagen festgelegt. Zum Beispiel ist $\neg(\varphi \rightarrow \chi)$ genau dann wahr, wenn φ wahr ist und

χ falsch. Die Aussagenlogik unterscheidet zwischen Elementaraussagen und komplexen Aussagen. Der Wahrheitswert von komplexen Aussagen ist determiniert durch den Wahrheitswert der Elementaraussagen. Der Wahrheitswert von Elementaraussagen ist nicht weiter festgelegt, er kann je nach Lage der Dinge das eine oder das andere sein. Der Satz 'Es regnet.' kann gestern wahr und jetzt falsch sein, morgen schon wieder wahr.

Wir werden die Aussagenlogik nun formal entwickeln. Dabei wollen wir im Gegensatz zur gebräuchlichen Herangehensweise Aussagen nicht als Terme auffassen, sondern als Zeichenketten über einem Alphabet. Dies hat den Vorzug, dass wir uns nicht mit der Definition eines Terms herumschlagen müssen. An die Stelle des abstrakten Begriffs 'Term' tritt jetzt der Begriff 'wohlgeformte Zeichenkette'. Obwohl auch hier exakte Formulierungen einen gewissen Aufwand erfordern, lehren sie uns doch den korrekten Gebrauch mit Zeichenketten, welche diejenigen Objekte sind, mit Hilfe derer wir uns den Computern mitteilen. Wir benutzen nicht alle der oben besprochenen Junktoren, sondern nur $\neg$, $\wedge$ und $\vee$. Dies ist keine Einschränkung, wie wir noch zeigen werden. Es erleichtert aber die Arbeit wesentlich.

Definition 44 *Es sei* $P := \{\mathtt{0}, \mathtt{1}, \mathtt{p}, \mathtt{(}, \mathtt{)}, \top, \neg, \wedge, \vee\}$*. Die Menge Aus der* ***wohlgeformten Zeichenketten*** *oder schlicht der* ***Aussagenterme*** *ist die kleinste Teilmenge von* P^**, für die gilt:*

1. $\mathtt{p}\vec{\alpha} \in Aus$*, wo* $\vec{\alpha}$ *eine Binärfolge ist.*
2. $\top \in Aus$*.*
3. *Ist* $\vec{x} \in Aus$*, so ist auch* $\mathtt{(}\neg\vec{x}\mathtt{)} \in Aus$*.*
4. *Sind* $\vec{x}$ *und* $\vec{y}$ *in Aus, so auch* $\mathtt{(}\vec{x} \wedge \vec{y}\mathtt{)}$*.*
5. *Sind* $\vec{x}$ *und* $\vec{y}$ *in Aus, so auch* $\mathtt{(}\vec{x} \vee \vec{y}\mathtt{)}$*.*

Hierbei ist eine ***Binärfolge*** *eine endliche Folge aus* `0` *und* `1`*. Wir nennen die Folge* $\mathtt{p}\vec{\alpha}$ *eine* ***Variable****. Die Menge der Variablen heißt* Var*.*

Hierbei spielen die Variablen die Rolle der nicht weiter analysierbaren Aussagen, etwa

1. $3+4$ `ist größer als 5.`
2. `London ist südlich von Hamburg.`

3. `Der Mond ist aus grünem Käse.`

Ist $\vec{x}$ eine Aussage, so bezeichnet $(\neg\vec{x})$ die dazu gegenteilige, verneinte Aussage. Diese wird umgangssprachlich allerdings anders wiedergegeben. In unseren Beispielen muss für die Verneinung das Wort `nicht` hinter das finite Verb, hier `ist`, gestellt werden.

1. $3+4$ `ist nicht größer als 5.`

2. `London ist nicht südlich von Hamburg.`

3. `Der Mond ist nicht aus grünem Käse.`

Sind $\vec{x}$ und $\vec{y}$ Aussagen, so bezeichnet $(\vec{x} \wedge \vec{x})$ die Konjunktion dieser Aussagen. Das Zeichen $\wedge$ wird im Deutschen durch `und` wiedergegeben und die Klammern fallen weg. Es bezeichnet $(\vec{x} \vee \vec{y})$ die Disjunktion; $\vee$ wird im Deutschen durch `oder` wiedergegeben.

In den meisten Lehrbüchern nimmt man einen unendlichen Vorrat $\mathtt{p}_i$, $i \in \omega$, an Variablen an. Dann hat man allerdings ein unendliches Alphabet. Dies haben wir dadurch vermieden, dass wir die Zahl i durch eine Binärfolge ersetzen. Allerdings werden wir trotzdem die Schreibweise $\mathtt{p}_i$ anstelle von $\mathtt{p}\vec{\alpha}$ verwenden, wobei i die $i$te Zahl ist, welche durch eine Binärfolge kodiert wird. (Also $\varepsilon \mapsto 0$, $\mathtt{0} \mapsto 1$, $\mathtt{1} \mapsto 2$, $\mathtt{00} \mapsto 3$ und so weiter.)

Als erstes wollen wir klären, wann eine Aussage wahr ist.

Definition 45 *Eine **Belegung** ist eine Funktion $\beta : Var \to 2 \quad (= \{0,1\})$.*

Gegeben eine Belegung und eine wohlgeformte Zeichenkette $\vec{x}$, definieren wir $[\vec{x}]^\beta$, den **Wahrheitswert von $\vec{x}$ unter** β, wie folgt.

$$\begin{array}{lll} [\top]^\beta & := & 1 \\ [\vec{x}]^\beta & := & \beta(\vec{x}) \qquad \vec{x} \in Var \\ [(\neg\vec{x})]^\beta & := & -[\vec{x}]^\beta \\ [(\vec{x} \wedge \vec{y})]^\beta & := & [\vec{x}]^\beta \cap [\vec{y}]^\beta \\ [(\vec{x} \vee \vec{y})]^\beta & := & [\vec{x}]^\beta \cup [\vec{y}]^\beta \end{array}$$

Hierbei sind $-$, $\cap$ und $\cup$ die folgenden Funktionen.

$$\begin{array}{c|c} & \neg \\ \hline 0 & 1 \\ 1 & 0 \end{array} \qquad \begin{array}{c|cc} \cap & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array} \qquad \begin{array}{c|cc} \cup & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array}$$

(Man mag sich überlegen, dass die Symbole $-$, $\cap$ und $\cup$ tatsächlich die Operationen Komplement, Schnitt und Vereinigung auf der Menge 2 sind.)

Wir sagen nun, $\vec{x}$ sei **wahr unter** β (oder β **erfüllt** $\vec{x}$), falls $[\vec{x}]^\beta = 1$. Andernfalls heißt $\vec{x}$ **falsch unter** β. Die eben getroffenen Vereinbarungen sind nur Formalisierungen unseres Alltagsverständnisses: $(\neg\vec{x})$ ist genau dann wahr unter β, wenn $\vec{x}$ unter β falsch ist. $(\vec{x} \wedge \vec{y})$ ist genau dann unter β wahr, wenn sowohl $\vec{x}$ wie auch $\vec{y}$ unter β wahr sind. $(\vec{x} \vee \vec{y})$ ist genau dann wahr unter β, wenn $\vec{x}$ oder $\vec{y}$ (oder auch beide) unter β wahr sind.

Wir werden im Folgenden von Aussagen reden und nicht von wohlgeformten Zeichenketten. Dies hat keine besondere Bedeutung. Es erlaubt uns lediglich, von der Darstellungsweise als Zeichenkette abzusehen. Die Identifikation einer Aussage mit einer Zeichenkette hatte lediglich praktische Gründe. Sie sollte uns mit der Tatsache konfrontieren, dass Aussagen mit Hilfe von Zeichenketten mitgeteilt werden. Sobald man dies verinnerlicht hat, kann man zu der ursprünglichen Sprechweise gefahrlos zurückkehren. Wir bezeichnen Aussagen mit kleinen griechischen Buchstaben, etwa φ, χ, γ, δ, falls wir von ihrem Zeichenkettencharakter absehen möchten.

Definition 46 *Eine Aussage heißt* ***Tautologie****, wenn sie unter jeder Belegung wahr ist. Eine Aussage heißt* ***Kontradiktion****, wenn sie unter keiner Belegung wahr ist; und sie heißt* ***Kontingenz****, wenn sie weder Tautologie noch Kontradiktion ist.*

Es ist also eine Aussage genau dann eine Tautologie, wenn ihre Verneinung eine Kontradiktion ist. Man sagt, eine Aussage sei **erfüllbar**, wenn es eine Belegung gibt, die sie wahr macht. Genau dann ist eine Aussage erfüllbar, wenn sie keine Kontradiktion ist.

Definition 47 *Zwei Aussagen φ und χ heißen* ***äquivalent****, in Zeichen $\varphi \equiv \chi$, falls für jede Belegung β $[\varphi]^\beta = [\chi]^\beta$.*

Offensichtlich ist $\equiv$ eine Äquivalenzrelation auf der Menge der Aussagenterme. Ferner ist φ genau dann eine Tautologie, wenn φ zu $\top$ äquivalent ist.

Wir definieren $var(\varphi)$ als die Menge der in φ auftretenden Variablen. Dann gilt:

Lemma 48 *Es seien β, γ Belegungen mit $\beta \restriction var(\varphi) = \gamma \restriction var(\varphi)$. Dann ist $[\varphi]^\beta = [\varphi]^\gamma$.*

Enthält φ also n Variablen, so gibt es höchstens 2^n verschiedene Fälle, die man wirklich durchgehen muss. Wir werden bald auch Verfahren kennenlernen, wie man systematisch und rationell testet, ob eine Aussage erfüllbar ist oder nicht.

Einer der wichtigsten Begriffe der Logik ist der *Folgerungsbegriff*. Wir folgern aus einer Menge von Tatsachen eine andere Tatsache. Korrektes Schließen geht von wahren Sätzen zu wahren Sätzen.

Definition 49 *Es sei Γ eine Menge von Aussagen und φ eine Aussage. Wir sagen, φ **folgt aus** Γ und schreiben $\Gamma \vDash \varphi$, falls jede Belegung, die jede Aussage aus Γ erfüllt, auch φ erfüllt. Man schreibt anstelle von $\{\gamma_0, \dots, \gamma_{n-1}\} \vDash \varphi$ auch $\gamma_0; \gamma_1; \dots; \gamma_{n-1} \vDash \varphi$. Ferner schreibt man auch $\Delta; \chi$ anstelle von $\Delta \cup \{\chi\}$ sowie $\Gamma; \Delta$ anstelle von $\Gamma \cup \Delta$.*

Wir nennen in $\Gamma \vDash \varphi$, ein Element aus Γ eine **Prämisse** und φ die **Konklusion**. Sei W eine Menge von Variablen. Zu einer Menge H von Belegungen sei $H \restriction W := \{\beta \restriction W : \beta \in H\}$. Es sei H_Γ die Menge aller Belegungen, welche Γ erfüllen, und es sei H_φ die Menge aller Belegungen, welche φ erfüllen. Dann gilt:

Lemma 50 *Genau dann ist $\Gamma \vDash \varphi$, wenn $H_\Gamma \restriction \mathsf{var}(\varphi) \subseteq H_\varphi \restriction \mathsf{var}(\varphi)$.*

Beweis. Setze $W := \mathsf{var}(\varphi)$. Sei $\Gamma \vDash \varphi$. Ferner sei $\beta \in H_\Gamma \restriction W$. Dann existiert eine Belegung $\gamma \in H_\Gamma$ mit $\gamma \restriction W = \beta \restriction W$. Daher ist Γ unter γ erfüllt und damit auch φ. Also $\gamma \in H_\varphi$. Es ist $[\varphi]^\gamma = [\varphi]^\beta$, da $\beta \restriction W = \gamma \restriction W$. Also ist φ auch unter β erfüllt und so $\beta \in H_\varphi \restriction W$. Sei umgekehrt $H_\Gamma \restriction W \subseteq H_\varphi \restriction W$. Sei Γ unter β erfüllt. Dann ist $\beta \in H_\Gamma$ und so $\beta \restriction W \in H_\Gamma \restriction W$. Also ist $\beta \restriction W \in H_\varphi \restriction W$. Also ist $\beta \in H_\varphi$. Q. E. D.

Wir notieren ein paar Eigenschaften von $\vDash$:

Satz 51 *Die Relation $\vDash$ hat folgende Eigenschaften.*

1. $\varphi \vDash \varphi$.
2. *Ist* $\Gamma \vDash \varphi$ *und* $\Gamma \subseteq \Delta$, *so auch* $\Delta \vDash \varphi$.
3. *Ist* $\Gamma \vDash \chi$ *für jedes* $\chi \in \Delta$ *und* $\Delta \vDash \varphi$, *so* $\Gamma \vDash \varphi$.
4. *Ist* $\Gamma \vDash \varphi$ *und* $\Delta; \varphi \vDash \chi$, *so* $\Gamma; \Delta \vDash \chi$.

Beweis. Dies sieht man so: (1) Es sei β eine Belegung, welche φ erfüllt. Dann erfüllt β φ. (2) Es sei β eine Belegung, welche jede Aussage aus Δ erfüllt. Dann erfüllt β jede Aussage aus Γ, da $\Gamma \subseteq \Delta$. Nach Voraussetzung ist dann φ durch β erfüllt. (3) Es sei β eine Belegung, welche jede Aussage aus Γ erfüllt. Dann erfüllt sie wegen $\Gamma \vDash \chi$ für jedes $\chi \in \Delta$ auch jede Aussage aus Δ. Da $\Delta \vDash \varphi$, so erfüllt β auch φ. (4) Es sei $\Gamma \vDash \varphi$ sowie $\Delta; \varphi \vDash \chi$. Sei $\Gamma; \Delta$ unter β erfüllt. Dann ist Γ unter β erfüllt, und deswegen auch φ. Da auch Δ unter β erfüllt ist, ist jetzt χ erfüllt. Q. E. D.

Definition 52 *Eine **Substitution** ist eine beliebige Abbildung $s : \mathit{Var} \to \mathit{Aus}$. Eine Substitution definiert eine homomorphe Fortsetzung $\overline{s}$ auf folgende Weise.*

$$\begin{array}{lcll} \overline{s}(\vec{x}) & := & s(\vec{x}), & \textit{falls } \vec{x} \textit{ Variable} \\ \overline{s}(\texttt{(}\neg\vec{x}\texttt{)}) & := & (\neg\overline{s}(\vec{x})) & \\ \overline{s}(\texttt{(}\vec{x}\wedge\vec{y}\texttt{)}) & := & (\overline{s}(\vec{x})\wedge\overline{s}(\vec{y})) & \\ \overline{s}(\texttt{(}\vec{x}\vee\vec{y}\texttt{)}) & := & (\overline{s}(\vec{x})\vee\overline{s}(\vec{y})) & \end{array}$$

Wir bezeichnen $\overline{s}(\vec{x})$ auch mit $\vec{x}^s$.

Dass $\vec{x}^s$ eindeutig bestimmt ist, hängt wiederum mit der eindeutigen Lesbarkeit zusammen, die wir noch ausführlich beweisen werden.

Lemma 53 *Ist β eine Belegung und s eine Substitution, so ist $\beta \circ \overline{s}$ eine Belegung und es gilt $[\vec{x}]^{\beta\circ\overline{s}} = [\overline{s}(\vec{x})]^\beta$.*

Ist Γ eine Menge von Formeln, so bezeichnen wir mit Γ^s die Menge $\{\gamma^s : \gamma \in \Gamma\}$.

Satz 54 *Es sei s eine Substitution. Dann gilt*

1. *Ist φ Tautologie, so auch φ^s.*
2. *Ist φ Kontradiktion, so auch φ^s.*
3. *Ist $\Gamma \vDash \varphi$, so gilt $\Gamma^s \vDash \varphi^s$.*

7. Teil: Aussagenlogik II: Eindeutige Lesbarkeit

Die alles entscheidene Frage ist nun die, ob $[\vec{x}]^\beta$ eigentlich eindeutig bestimmt ist. Dies ist der Fall, wie wir jetzt in diesem Teil beweisen werden. Was wir beweisen werden, ist, dass jede wohlgeformte Zeichenkette in eindeutiger Weise aus Variablen und Junktoren zusammengesetzt ist. Dies ist nicht selbstverständlich. Falls wir zum Beispiel alle Klammern löschen, so bekommen wir Zeichenketten, deren Erzeugung nicht eindeutig rekonstruierbar ist. Zum Beispiel kann $\texttt{p01} \wedge \texttt{p1} \vee \texttt{p111}$ entweder dem Term $\texttt{(p01} \wedge \texttt{(p1} \vee \texttt{p111))}$ oder dem Term $\texttt{((p01} \wedge \texttt{p1)} \vee \texttt{p111)}$ entsprechen. Man rechnet leicht nach, dass sie unter der Belegung $\beta : \texttt{p1} \mapsto 1$, $\texttt{p01} \mapsto 0$, $\texttt{p111} \mapsto 1$ verschiedene Werte haben.

$$\begin{aligned}
[\texttt{(p01} \wedge \texttt{(p1} \vee \texttt{p111))}]^\beta &= [\texttt{p01}]^\beta \cap [\texttt{(p1} \vee \texttt{p111)}]^\beta \\
&= [\texttt{p01}]^\beta \cap ([\texttt{p1}]^\beta \cup [\texttt{p111}]^\beta) \\
&= 0 \cap (1 \cup 1) \\
&= 0
\end{aligned}$$

Auf der anderen Seite aber ist

$$\begin{aligned}
[\texttt{((p01} \wedge \texttt{p1)} \vee \texttt{p111)}]^\beta &= [\texttt{(p01} \wedge \texttt{p1)}]^\beta \cap [\texttt{p111}]^\beta \\
&= ([\texttt{p01}]^\beta \cap [\texttt{p1}]^\beta) \cup [\texttt{p111}]^\beta \\
&= (0 \cap 1) \cup 1 \\
&= 1
\end{aligned}$$

Das unterschiedliche Ergebnis kommt also zustande, weil wir nicht mehr wissen, wie der Term erzeugt worden ist. Allerdings ist der Prozess der Erzeugung des Terms auch nicht eigentlich das Entscheidende, sondern sein Aufbau aus anderen Termen. Ein Term besteht nämlich aus unmittelbaren Teiltermen, und diese müssen eindeutig bestimmt sein. Wenn das der Fall ist, dann ist der Wahrheitswert eindeutig bestimmt. Das ist einerseits intuitiv klar, kann aber auch rigoros mit Nachfolgerinduktion bewiesen werden. Wir wollen solch einen Beweis jetzt liefern. Dabei werden wir nützliche Kleinarbeit im Hantieren mit Zeichenketten leisten müssen.

Definition 55 *Es seien $\vec{x}$ und $\vec{y}$ wohlgeformte Zeichenketten. $\vec{y}$ heißt **(echter) Teilterm von** $\vec{x}$, falls $\vec{y}$ (echtes) Teilwort von $\vec{x}$ ist. $\vec{y}$ heißt **unmittelbarer Teilterm von** $\vec{x}$, wenn $\vec{y}$ echter Teilterm von $\vec{x}$ ist, es aber keinen echten Teilterm $\vec{u}$ von $\vec{x}$ gibt, dessen echter Teilterm $\vec{y}$ ist.*

Lemma 56 *Die Relation **ist echter Teilterm von** ist eine strikte, fundierte Ordnung.*

Beweis. Falls $\vec{y}$ echter Teilterm von $\vec{x}$ ist, so hat $\vec{y}$ kleinere Länge als $\vec{x}$. Also ist die Relation fundiert. Sie ist offensichtlich auch transitiv und strikt, da es die Ordnung *ist echtes Teilwort von* auch ist. Q. E. D.

Ein Wort kann man mehrfach als Teilwort in einem Wort vokommen. So hat `bab` mehrere Vorkommen in `abababab` (genau drei Stück). Um den Wahrheitswert einer Aussage zu bestimmen, müssen wir nicht nur darauf achten, welche Teilterme darin vorkommen, sondern auch, wo sie vorkommen. Daher müssen wir Teilwortvorkommen betrachten.

Definition 57 *Ein* ***Vorkommen von*** $\vec{y}$ *in* $\vec{x}$ *ist ein Paar Zeichenketten* $\langle \vec{u}, \vec{v} \rangle$ *derart, dass* $\vec{x} = \vec{u}\vec{y}\vec{v}$.

Die Vorkommen von `bab` in `abababab` sind also $C_1 = \langle \texttt{a}, \texttt{abab} \rangle$, $C_2 = \langle \texttt{aba}, \texttt{ab} \rangle$ und $C_3 = \langle \texttt{ababa}, \varepsilon \rangle$.

	a	b	a	b	a	b	a	b
C_1		∘	∘	∘				
C_2				∘	∘	∘		
C_3						∘	∘	∘

Wie man sieht, überlappen C_1 und C_2 in einem Buchstaben (genauer: in einem Buchstabenvorkommen). Genauso überlappen C_2 und C_3 in einem Buchstaben(vorkommen). Dagegen überlappen C_1 und C_3 nicht, und C_1 liegt vor C_3. Die Begriffe des Überlappens und des Nacheinanderfolgens werden wir jetzt exakt definieren. Sie sind nicht daran gebunden, dass wir Vorkommen desselben Wortes nehmen; es können ganz beliebige Teilworte betrachtet werden.

Definition 58 *Es sei* $\vec{x}$ *ein Wort,* $C_1 = \langle \vec{u}_1, \vec{v}_1 \rangle$ *und* $C_2 = \langle \vec{u}_2, \vec{v}_2 \rangle$ *Vorkommen beliebiger Teilworte* $\vec{w}_1$ *und* $\vec{w}_2$ *in* $\vec{x}$. *Wir sagen,* C_2 ***folgt*** C_1, *falls* $\vec{u}_1\vec{w}_1$ *Präfix von* $\vec{u}_2$ *ist.* C_2 *folgt* C_1 ***unmittelbar****, falls* $\vec{u}_2 = \vec{u}_1\vec{w}_1$. C_1 *und* C_2 ***überlappen****, falls* C_2 *nicht auf* C_1 *folgt und* C_1 *nicht auf* C_2. C_1 *ist* ***in*** C_2 ***enthalten****, falls* $\vec{u}_2$ *Präfix von* $\vec{u}_1$ *und* $\vec{v}_2$ *Suffix von* $\vec{v}_1$. C_1 *und* C_2 ***überlappen echt****, wenn sie überlappen, aber weder* C_1 *in* C_2 *noch* C_2 *in* C_1 *enthalten ist.*

	a	b	c	a	d	b	d	a
D_0		∘	∘	∘	∘	∘		
D_1				∘	∘			
D_2							∘	∘

In dem Beispiel gilt: D_2 folgt auf D_1 und auf D_0. Sogar folgt D_2 unmittelbar auf D_0. D_1 ist in D_0 enthalten. Das leere Wort ist in jedem Wort enthalten.

Es hat in einem Wort der Länge n genau $n + 1$ Vorkommen. Keines seiner Vorkommen überlappt echt mit einem Vorkommen irgendeines anderen Wortes.

Als Letztes wollen wir uns mit induzierten Vorkommen befassen. Es sei $\vec{x}$ eine Zeichenkette, und $C = \langle \vec{u}, \vec{v} \rangle$ ein Vorkommen von $\vec{y}$ in $\vec{x}$. Ist dann $D = \langle \vec{u}_1, \vec{v}_1 \rangle$ ein Vorkommen von $\vec{z}$ in $\vec{y}$, so ist $\langle \vec{u}\vec{u}_1, \vec{v}_1\vec{v} \rangle$ ein Vorkommen von $\vec{z}$ in $\vec{x}$. So ist $\langle \mathtt{c}, \mathtt{b} \rangle$ ein Vorkommen von $\mathtt{ad}$ in $\mathtt{cadb}$. Dies induziert das Vorkommen $\langle \mathtt{abc}, \mathtt{bda} \rangle$ in $\mathtt{abcaddba}$. Ist umgekehrt $\langle \vec{u}_2, \vec{v}_2 \rangle$ ein Vorkommen von $\vec{z}$ in $\vec{x}$, so muss dem nicht ein Vorkommen von $\vec{z}$ in $\vec{y}$ entsprechen. Dies ist nur dann der Fall, wenn $\vec{u}$ Präfix von $\vec{u}_2$ und $\vec{v}$ Suffix von $\vec{v}_2$ ist. Denn dann existieren $\vec{u}_1$ und $\vec{v}_1$ mit $\vec{u}_2 = \vec{u}\vec{u}_1$ und $\vec{v}_2 = \vec{v}_1\vec{v}$, und das gesuchte Vorkommen von $\vec{z}$ in $\vec{y}$ ist $\langle \vec{u}_1, \vec{v}_1 \rangle$.

Erinnern wir uns an die Definition von wohlgeformten Zeichenketten. Die Menge Aus der wohlgeformten Zeichenketten ist die kleinste Teilmenge von P^*, für die gilt:

1. $\mathtt{p}\vec{\alpha} \in Aus$, wo $\vec{\alpha}$ eine Binärfolge ist.
2. $\top \in Aus$.
3. Ist $\vec{x} \in Aus$, so auch $(\neg\vec{x})$.
4. Sind $\vec{x}$ und $\vec{y}$ in Aus, so auch $(\vec{x} \wedge \vec{y})$.
5. Sind $\vec{x}$ und $\vec{y}$ in Aus, so auch $(\vec{x} \vee \vec{y})$.

In Fall 3, 4, 5 haben wir ein Vorkommen von $\neg$, $\wedge$ bzw. $\vee$ ausgezeichnet. Dieses Vorkommen nennen wir das **Hauptsymbol** in der konstruierten Zeichenfolge. Es genügt offensichtlich, wenn wir zeigen, dass jede wohlgeformte Zeichenkette, welche nicht eine Variable ist, ein eindeutig bestimmtes Hauptsymbol besitzt. Ist dieses gefunden, so können wir die unmittelbaren Teilterme eindeutig wiederfinden: wir löschen zunächst die äußeren Klammern. Im Falle 3 ist der Teilterm das Ergebnis der Löschung von $\neg$; im Falle 4 und 5 ist der erste Teilterm das Ergebnis der Löschung des Hauptsymbols und der darauffolgenden Zeichenkette, der zweite das Ergebnis der Löschung des Präfixes bis zum Hauptsymbol. Das nächste, was wir festlegen müssen, ist ein Variablenvorkommen.

Definition 59 *Es sei $\vec{x}$ eine wohlgeformte Zeichenkette. Ein **Vorkommen der Variable** $\mathtt{p}\vec{\alpha}$ in $\vec{x}$ ist ein Vorkommen von $\mathtt{p}\vec{\alpha}$ als Teilwort in $\vec{x}$, $\langle \vec{u}, \vec{v} \rangle$,* *dergestalt, dass das erste Symbol von $\vec{v}$ nicht* `0` *oder* `1` *ist.* $\mathtt{p}\vec{z}$ ***kommt in*** $\vec{x}$ ***vor****, falls es ein Vorkommen dieser Variable in* $\vec{x}$ *gibt.*

So ist also $\langle \mathtt{(p01} \wedge \mathtt{(}\neg, \mathtt{))}\rangle$ ein Vorkommen der Variable `p110` in $\vec{x} = \mathtt{(p01} \wedge \mathtt{(}\neg\mathtt{p110))}$. Man beachte, dass die Variable `p11` *nicht* in $\vec{x}$ vorkommt. Aufgrund der Definition ist dies so, weil das einzige Vorkommen dieser Zeichenkette $\langle \mathtt{(p01} \wedge \mathtt{(}\neg, \mathtt{0))}\rangle$ zwar ein Vorkommen der Zeichenkette `p11` ist, aber $\vec{v}$ in diesem Fall mit `0` beginnt. Man mache sich anhand dieses Beispiels klar, dass lediglich die Variablen `p01` und `p110` in diesem Term vorkommen, obwohl die Zeichenketten `p`, `p0`, `p1`, `p11` auch vorkommen, nur eben nicht als Variable.

Wir definieren jetzt für ein Wort über P:

$$\begin{array}{lcll}
\gamma(\varepsilon) & := & 0, & \\
\gamma(\mathtt{(}) & := & 1, & \\
\gamma(\mathtt{)}) & := & -1, & \\
\gamma(x) & := & 0, & x \in P - \{\mathtt{(}, \mathtt{)}\}, \\
\gamma(\vec{x} \cdot \vec{y}) & := & \gamma(\vec{x}) + \gamma(\vec{y}) &
\end{array}$$

Hierbei ist x eine Variable für Symbole aus P, nicht für beliebige Zeichenketten.

Lemma 60 *Es sei $\vec{x}$ eine wohlgeformte Zeichenkette und $\vec{x} = \vec{u}\vec{v}$. Dann gilt $\gamma(\vec{u}) \geq 0$ und $\gamma(\vec{v}) \leq 0$. Ferner ist $\gamma(\vec{x}) = 0$.*

Beweis. Durch Induktion über die Länge von $\vec{x}$. Gewiss gilt die Behauptung für Variablen und für $\top$. Nun sei sie bereits für $\vec{y}$ und $\vec{z}$ gezeigt. Es sei $\vec{x} = \mathtt{(}\neg\vec{y}\mathtt{)}$. Es ist $\gamma(\vec{x}) = \gamma(\mathtt{(}) + \gamma(\neg) + \gamma(\vec{y}) + \gamma(\mathtt{)}) = 1 + 0 + 0 + (-1) = 0$, da $\gamma(\vec{y}) = 0$ nach Induktionsvoraussetzung. Es genügt jetzt zu zeigen, dass für jedes Präfix $\vec{u}$ von $\vec{x}$ gilt: $\gamma(\vec{u}) \geq 0$. Sei $\vec{x} = \vec{u}\vec{v}$. Ist $\vec{u} = \varepsilon$ oder $\vec{u} = \mathtt{(}$, so ist die Behauptung sicher richtig. Ebenfalls für $\vec{u} = \vec{x}$. Andernfalls ist aber $\vec{u} = \mathtt{(}\neg\vec{u}_1$, wo $\vec{u}_1$ ein Präfix von $\vec{y}$ ist. Dann ist nach Induktionsvoraussetzung $\gamma(\vec{u}_1) \geq 0$, und so ist sogar $\gamma(\vec{u}) = 1 + \gamma(\vec{u}_1) > 0$. Ähnlich die Fälle, wo $\vec{x} = \mathtt{(}\vec{u} \wedge \vec{v}\mathtt{)}$ oder $\vec{x} = \mathtt{(}\vec{u} \vee \vec{v}\mathtt{)}$. Q. E. D.

Definition 61 *Es sei $\vec{x}$ eine wohlgeformte Zeichenkette, und $C = \langle \vec{u}, \vec{v}\rangle$ ein Vorkommen eines Junktors oder einer Variable. Wir sagen, die* ***Einbettungstiefe*** *von C in $\vec{x}$ sei n, falls $\gamma(\vec{u}) = n$.*

Wir nennen $\vec{x}$ **zusammengesetzt**, wenn $\vec{x}$ keine Variable ist und $\vec{x} \neq \top$. Dann gilt folgender umittelbar einleuchtender Sachverhalt.

Lemma 62 *Es sei $\vec{x}$ zusammengesetzt, und es sei $\vec{y} \neq \varepsilon, \vec{x}$ ein Präfix von $\vec{x}$. Dann gilt $\gamma(\vec{y}) > 0$.*

Lemma 63 *Es sei $\vec{x}$ eine wohlgeformte Zeichenkette. Ist $\vec{x}$ zusammengesetzt, so existiert ein eindeutig bestimmtes Vorkommen eines Junktors mit der Einbettungstiefe 1. Dieser ist das Hauptsymbol von $\vec{x}$.*

Beweis. Induktion über die Länge von $\vec{x}$. Der Fall, wo $\vec{x}$ unzusammengesetzt ist, ist bereits durch die Annahmen gedeckt. Jetzt sei $\vec{x}$ zusammengesetzt. (a) $\vec{x} = (\neg\vec{y})$. Dann ist $\vec{y}$ entweder unzusammengesetzt oder es besitzt ein Hauptsymbol, welches nach Induktionsvoraussetzung eindeutig ist. Ist $\vec{y}$ unzusammengesetzt, so ist es Variable oder $\top$. In diesem Fall enthält es kein Vorkommen eines Junktors $\neq \top$. Ist $\vec{y}$ zusammengesetzt, so beginnt es mit einer Klammer. Also ist für jedes Vorkommen eines Junktors $\neq \top$ die Einbettungstiefe in $\vec{y}$ mindestens 1, weswegen sie in $\vec{x}$ mindestens 2 ist. Dies bestätigt die Behauptung in diesem Fall. (b) $\vec{x} = (\vec{u} \wedge \vec{v})$. Analog beweisen wir, dass jedes Vorkommen eines Junktors $\neq \top$ in $\vec{u}$ in $\vec{x}$ die Einbettungstiefe ≥ 2 hat. Ebenso gilt dies für Vorkommen in $\vec{v}$. Dazu beachte man nämlich, dass wenn $C = \langle \vec{y}, \vec{z} \rangle$ ein Vorkommen eines Junktors in $\vec{v}$ ist, so ist $\langle (\vec{u} \wedge \vec{y}, \vec{z}) \rangle$ Vorkommen dieses Junktors in $\vec{x}$. Es ist dann $\gamma((\vec{u} \wedge \vec{y}) = \gamma(() + \gamma(\vec{u}) + \gamma(\wedge) + \gamma(\vec{y}) = 1 + \gamma(\vec{y}) \geq 2$, nach Voraussetzung. (c) $\vec{x} = (\vec{u} \vee \vec{v})$. Genauso wie der vorige Fall. Q. E. D.

Damit ist die zentrale Behauptung gezeigt. Wir formulieren sie jetzt explizit.

Lemma 64 (Eindeutige Lesbarkeit) *Es sei $\vec{x} \in Aus$. Dann tritt genau einer der folgenden Fälle auf.*

1. $\vec{x} = \mathrm{p}\vec{x} \in Var$.
2. $\vec{x} = \top$.
3. $\vec{x} = (\neg\vec{y})$ *mit* $\vec{y} \in Aus$.
4. $\vec{x} = (\vec{y} \wedge \vec{z})$ *mit* $\vec{y} \in Aus$ *und* $\vec{z} \in Aus$.
5. $\vec{x} = (\vec{y} \vee \vec{z})$ *mit* $\vec{y} \in Aus$ *und* $\vec{z} \in Aus$.

Im Falle 3 ist das Hauptsymbol wie auch das Vorkommen von $\vec{y}$ eindeutig bestimmt; in den Fällen 4 und 5 sind Hauptsymbol und die Vorkommen von $\vec{y}$ und $\vec{z}$ jeweils eindeutig bestimmt.

Beweis. Wie schon gezeigt, können wir aus $\vec{x}$ das Hauptsymbol eindeutig bestimmen und daraus wiederum die unmittelbaren Teilterme. Q. E. D.

Korollar 65 *Es sei $\vec{x}$ eine wohlgeformte Zeichenkette. Sind dann C und C' zwei verschiedene Vorkommen eines Teilterms, so überlappen sie nicht echt.*

Beweis. Wir skizzieren einen Beweis. Wiederum verwenden wir Induktion. Ist C ein Vorkommen eines Teilterms von $\vec{x}$, so ist C (a) Vorkommen von $\vec{x}$ (also $C = \langle \varepsilon, \varepsilon \rangle$), oder (b) Vorkommen eines echten Teiltermes von $\vec{x}$. Dies ist unmittelbar klar. Sei nun C' ein von C verschiedenes Vorkommen eines Teilterms. Im Falle (a) sind wir fertig. Aus Symmetriegründen dürfen wir jetzt auch $C' = \langle \varepsilon, \varepsilon \rangle$ ausschließen. Sei also jetzt weder C noch C' Vorkommen von $\vec{x}$. Dann sind sie Vorkommen von echten Teiltermen von $\vec{x}$. Nach Eindeutigkeit der Zerlegung von $\vec{x}$ entsprechen den Vorkommen von C und C' in $\vec{x}$ nunmehr Vorkommen D und D' in unmittelbaren Teiltermen von $\vec{x}$. Sind die unmittelbaren Teilterme verschieden, so sind sie disjunkt (da die Zerlegung ja eindeutig ist). Sind sie gleich, so können wie nunmehr die Induktionsbehauptung verwenden. Q. E. D.

Die oben definierte Konvention zur Notation von Termen heißt man **Infixnotation**. Sie existiert in mehreren Varianten, wobei man mehr oder weniger Klammern wegläßt und dabei Prioritäten zwischen den Operationen bestimmt (*Punkt- vor Strichrechnung*). Ferner kann man die Klammern um $(\neg\vec{y})$ ohne Verlust der Eindeutigkeit einsparen. Man setzt jedoch manchmal Klammern, um die Formel augenfällig zu strukturieren. Dies soll verdeutlichen, dass man mit Zeichenketten ziemlich vorsichtig umgehen muss. Es ist der Einsicht von Łukasiewicz zu verdanken, dass wir Aussagen auch anders eindeutig aufschreiben können und dabei die Klammern sparen können. Die von ihm erfundene, sogenannte **Polnische Notation** (**PN**) stellt den Junktor stets vor alle seine Argumente. Wir schreiben also $\wedge\vec{y}\vec{z}$ anstelle von $(\vec{y} \wedge \vec{z})$ und $\vee\vec{y}\vec{z}$ anstelle von $(\vec{y} \wedge \vec{z})$. Und wir schreiben $\neg\vec{y}$ anstelle von $(\neg\vec{y})$. Wir nennen *PAus* die Menge der so gewonnenen Zeichenketten. Dass diese Schreibweise eindeutig ist, beweisen wir so. Zunächst einmal überlegen wir uns, dass wir Vorkommen von Variablen wie vorher definieren. Anschließend behandeln wir Variablen wie eine Einheit. (Dies bedeutet, dass wir ein Vorkommen einer Variable nunmehr als ein Einzelzeichen betrachten.)

Wir definieren nun π induktiv wie folgt.

$$\begin{array}{lcll}
\pi(\varepsilon) & := & 0, & \\
\pi(x) & := & -1, & x \in \mathit{Var}, \\
\pi(\top) & := & -1, & \\
\pi(\neg) & := & 0, & \\
\pi(\wedge) & := & 1, & \\
\pi(\vee) & := & 1, & \\
\pi(\vec{x}\vec{y}) & := & \pi(\vec{x}) + \pi(\vec{y}) \,. &
\end{array}$$

Im allgemeinen Fall ist $\pi(x)$ gerade die Stelligkeit von x vermindert um Eins. Da Variablen und 0–stellige Junktoren die Stelligkeit 0 haben, bekommen sie also -1 zugewiesen.

Proposition 66 *Genau dann ist eine Zeichenkette $\vec{x}$ in PAus, wenn gilt: (a) $\pi(\vec{x}) = -1$, (b) für jedes echte Präfix $\vec{y}$ von $\vec{x}$ ist $\pi(\vec{y}) \geq 0$. Daraus folgt, dass kein echtes Präfix von $\vec{x}$ in PAus ist.*

Auch dies beweist man induktiv. Daraus folgt die eindeutige Lesbarkeit wie folgt. Ist $\vec{x}$ gegeben, so treten vier Fälle ein. (a) $\vec{x}$ ist eine Variable oder $\vec{x} = \top$. Dann ist $\vec{x}$ eindeutig zerlegbar, da Einzelzeichen. (b) $\vec{x}$ ist nicht von dieser Form. Dann beginnt $\vec{x}$ mit einem Junktor. Dieser ist eindeutig bestimmt. Ist er $\neg$, so haben wir $\pi(\neg\vec{y}) = \pi(\vec{y})$, und es folgt, dass $\vec{y}$ aus *PAus* ist. Ist der Junktor $\wedge$, so gilt $\vec{x} = \wedge\vec{y}\vec{z}$, und $\vec{y}$ und $\vec{z}$ sind aus *PAus*. Dann gilt: kein echtes Präfix von $\vec{y}$ ist in *PAus*, also ist $\vec{y}$ eindeutig bestimmt, damit aber auch $\vec{z}$. Ist der Junktor schließlich $\vee$, so sind wir fertig.

In gewissen Rechnern wird wird einer ähnlichen Konvention gearbeitet, die sogenannte **Umgekehrte Polnische Notation**. Sie unterscheidet sich von der PN dadurch, dass das Funktionssymbol hinter seine Argumente gestellt wird. Auch diese ist eindeutig lesbar, wie man zeigen kann. Der Grund für diese Wahl wird einsichtig, wenn man sich überlegt, dass der Rechner die Eingabe von links nach rechts liest, und Zahlen einfach als Ziffernfolgen (oder oft auch etwas kompliziertere Folgen) notiert werden. Das Ende einer Eingabe ist damit nicht eindeutig kenntlich. Tippen wir zuerst `1` ein, so kann dies die Eingabe der Zahl 1 sein oder eben der Beginn der Eingabe einer längeren Ziffernfolge, wie etwa `1342`. Um die Eingabe einer Zahl von der einer anderen Zahl zu trennen, hat man jetzt ja nicht das Operationssymbol zur Verfügung. Es bezeichnet also `+135` in PN ohne Klammern wahlweise `1+35` oder `13+5`. Damit dies nicht geschieht, braucht man ein Trennsymbol. Dies ist bei besagten Rechnern die Taste $\boxed{\texttt{enter}}$. In Polnischer Notation würde unsere Aufgabe `13+5` nun wie folgt aussehen:

$$\texttt{+13}\boxed{\texttt{enter}}\texttt{5}\boxed{\texttt{enter}}$$

In UPN dagegen benötigen wir nur dieses:

$$\texttt{13}\boxed{\texttt{enter}}\texttt{5+}$$

Wir sparen also ein Symbol und damit einen Tastendruck. Der Witz ist, dass wir ja nur das Ende einer Eingabe notieren müssen, nicht ihren Anfang. In PN aber begrenzt das Hauptysmbol den Anfang des ersten Operanden. In UPN begrenzt es das Ende des letzten Operanden.

Die Aktion von $\boxed{\texttt{enter}}$ ist wie folgt: die Eingaben liegen in einem Stapel, und eine Operation verbindet die jeweils oberen Elemente des Stapels. Nennen wie die Register von oben kommend R_0, R_1, und so weiter, so ist $+$ diejenige Funktion, welche den Inhalt von R_0 und R_1 zusammenzählt und das Ergebnis in R_1 schreibt. $\boxed{\texttt{enter}}$ dagegen läßt alle Register eins nach unten gehen (das heißt, für jedes i wird der Inhalt von R_i nach R_{i+1} geschoben) und macht damit das Register R_0 frei für eine neue Eingabe. Man beachte, dass das Ausführen einer Operation automatisch Platz schafft für eine neue Eingabe.

8. Teil: Aussagenlogik III: Boolesche Funktionen

Es bezeichnet M^n für eine Menge M die Menge der n–Tupel über n. Wir beginnen mit einer Definition.

Definition 67 *Eine n–stellige* ***boolesche Funktion*** *ist eine Funktion $f : \{0,1\}^n \to \{0,1\}$. Eine boolesche Funktion ist eine Funktion $f : \{0,1\}^n \to \{0,1\}$ für ein beliebiges n.*

Zum Beispiel ist $-$ eine 1–stellige boolesche Funktion, $\cap$ und $\cup$ 2–stellige boolesche Funktionen. Betrachten wir die Aussage $\varphi = \texttt{((}\mathtt{p}_2 \vee \texttt{(}\neg\mathtt{p}_0\texttt{))} \wedge \texttt{(}\neg\mathtt{p}_1\texttt{))}$. Unter der Belegung $\beta : \mathtt{p}_0 \mapsto 0$, $\mathtt{p}_1 \mapsto 1$, $\mathtt{p}_2 \mapsto 1$ bekommt sie den Wahrheitswert 0. Unter der Belegung $\gamma : \mathtt{p}_0 \mapsto 1$, $\mathtt{p}_1 \mapsto 0$, $\mathtt{p}_2 \mapsto 1$ bekommt sie den Wert 1. Wir können auf natürliche Weise aus der Aussage eine Funktion f_φ definieren. Für sie gilt $f_\varphi(0,1,1) = 0$ und $f_\varphi(0,0,1) = 1$. Den genauen Zusammenhang stellt die folgende Definition dar.

Definition 68 *Es sei φ eine Aussage, welche höchstens die Variablen $\mathtt{p}_i$, $i < n$, enthält, mindestens aber $\mathtt{p}_{n-1}$, und es sei f eine n–stellige boolesche Funktion. Wir sagen, φ* ***repräsentiert*** *f falls für alle Belegungen β gilt $[\varphi]^\beta = f(\beta(\mathtt{p}_0), \beta(\mathtt{p}_1), \ldots, \beta(\mathtt{p}_{n-1}))$. Dies heißt:*

1. *Ist $f(x_0, x_1, \ldots, x_{n-1}) = 0$, so ist $[\varphi]^\beta = 0$ für alle Belegungen β, für welche $\beta(\mathtt{p}_i) = x_i$.*
2. *Ist $f(x_0, x_1, \ldots, x_{n-1}) = 1$, so ist $[\varphi]^\beta = 1$ für alle Belegungen β, für welche $\beta(\mathtt{p}_i) = x_i$.*

Die Aussage $\varphi = \mathtt{((p_2} \vee \mathtt{(}\neg \mathtt{p_0))} \wedge \mathtt{(}\neg\mathtt{p_1))}$ repräsentiert also die folgende Funktion.

x_0	x_1	x_2	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Wir sagen im Übrigen, eine Belegung β sei zu einem n–Tupel $\vec{x}$ **assoziiert**, falls $\beta(\mathtt{p}_i) = x_i$ für $i < n$. Offenkunding sind Aussagen genau dann äquivalent, wenn sie die gleichen booleschen Funktionen repräsentieren. Wir werden nun folgenden Satz beweisen.

Satz 69 *Zu jeder booleschen Funktion existiert eine Aussage, die sie repräsentiert.*

Diesen Satz nennt man die **Vollständigkeit** des Systems der Junktoren $\{\top, \neg, \wedge, \vee\}$. Darauf wollen wir im Anschluß an den Beweis näher eingehen.

Zunächst einmal beginnen wir mit dem Fall $n = 0$. Hier besitzen wir zwei Funktionen, nämlich $f_0 : \varnothing \mapsto 0$ und $f_1 : \varnothing \mapsto 1$. (Man beachte: es ist $M^0 := 1$ für jede Menge; also ist $\varnothing$ einziges Element von $\{0,1\}^0$.) Es repräsentiert $\neg\top$ die Funktion f_0 und $\top$ die Funktion f_1. Hier sei gleich erwähnt, dass, obwohl wir $\top \equiv \mathtt{(p_0} \vee \mathtt{(}\neg\mathtt{p_0))}$ haben, das Symbol $\top$ trotzdem nicht entbehrlich ist. Die Aussage $\mathtt{(p_0} \vee \mathtt{(}\neg\mathtt{p_0))}$ repräsentiert nämlich nicht die Funktion $f_1 : \{0,1\}^0 \to \{0,1\}$, sondern lediglich die konstante Funktion

$g : \{0,1\} \to \{0,1\}$, für die gilt $g(0) = g(1) = 1$. Für diesen (und nur für diesen Fall) wird also das Symbol $\top$ gebraucht. Im Folgenden sei jetzt $n > 0$. Wir sagen, f sei **primitiv**, falls es genau ein Tupel $\mathfrak{z} := \langle x_i : i < n\rangle$ gibt, für das $f(\mathfrak{z}) = 1$ ist. Wir bezeichnen diejenige primitive Funktion f, für welche $f(\mathfrak{z}) = 1$ mit $p_{\mathfrak{z}}$. Wir zeigen unsere Behauptung zunächst für primitive Funktionen. Es sei $\mathfrak{z} = \langle x_i : i < n\rangle$. Wir nehmen zuerst den Fall $n = 1$. Es ist also $\mathfrak{z} = 0$ oder $\mathfrak{z} = 1$. Wir setzen $\varphi(0) := \neg\mathtt{p}_0$ und $\varphi(1) = \mathtt{p}_0$. Dann repräsentiert $\varphi(0)$ die Funktion p_0 und $\varphi(1)$ die Funktion p_1. Im Falle $n = 2$ gibt es schon vier primitive Funktionen: p_{00}, p_{01}, p_{10} und p_{11}. Die sie repräsentierenden Aussagen sind

$$\begin{array}{rclrcl}\varphi(00) & := & \mathtt{((\neg p_0) \wedge (\neg p_1))} & \varphi(01) & := & \mathtt{((\neg p_0) \wedge p_1)} \\ \varphi(10) & := & \mathtt{(p_0 \wedge (\neg p_1))} & \varphi(11) & := & \mathtt{(p_0 \wedge p_1)}\end{array}$$

Nun gehen wir zum allgemeinen Fall über. Es sei $\vec{x}$ von der Länge n. Wir setzen

$$\begin{array}{rcl}\varphi(\vec{x}\mathtt{0}) & := & \mathtt{(}\varphi(\vec{x}) \wedge \mathtt{(\neg p}_n\mathtt{))} \\ \varphi(\vec{x}\mathtt{1}) & := & \mathtt{(}\varphi(\vec{x}) \wedge \mathtt{p}_n\mathtt{)}\end{array}$$

Dann repräsentiert $\varphi(\vec{x}\mathtt{0})$ die Funktion $p_{\vec{x}0}$ und $\varphi(\vec{x}\mathtt{1})$ die Funktion $p_{\vec{x}1}$. Damit ist gezeigt, dass zu jeder primitiven Funktion eine repräsentierende Aussage existiert.

Nun sei f eine beliebige Funktion. Wir nennen den **Träger** von f die Menge $T(f) := \{\mathfrak{z} : f(\mathfrak{z}) = 1\}$. Offensichtlich gilt für jede Funktion f

$$f = \max\, \{p_{\mathfrak{z}} : \mathfrak{z} \in T(f)\}$$

Zu jeder Menge $M \subseteq \{0,1\}^n$ setzen wir

$$\chi_M := \max\, \{p_{\mathfrak{z}} : \mathfrak{z} \in M\}$$

Offensichtlich ist dann $f = \chi_{T(f)}$. Daher sind wir fertig, wenn wir zu jeder Menge $M \subseteq \{0,1\}^n$ eine Aussage $\delta(M)$ finden, welche χ_M repräsentiert. Wir definieren für $M \subseteq \{0,1\}^n$ $\delta(M)$ induktiv über die Anzahl der Elemente in M. Es sei M leer. Dann setzen wir

$$\delta(M) := \neg\top\ .$$

Nun sei $M = N \cup \{\mathfrak{z}\}$ mit $\mathfrak{z} \notin N$. Nach Voraussetzung existiert eine Aussage $\delta(M)$, welche χ_N repräsentiert. Dann setze

$$\delta(M) := \mathtt{(}\delta(N) \vee \varphi(\mathfrak{z})\mathtt{)}\ .$$

Wir behaupten, dass $\delta(M)$ nunmehr χ_M repräsentiert. Sei dazu $\mathfrak{x} \in \{0,1\}^n$ beliebig. Fall 1: $\mathfrak{x} = \mathfrak{z}$. Dann ist $[\varphi(\mathfrak{z})]^\beta = 1$ für jede zu $\mathfrak{x}$ assoziierte Belegung β. Also ist $[\delta(M)]^\beta = [\varphi(N)]^\beta \cup 1 = 1$. Genauso ist $\chi_M(\mathfrak{x}) = 1$. Fall 2: $\mathfrak{x} \in N$. Dann ist $[\varphi(N)]^\beta = 1$ für jede zu $\mathfrak{x}$ assoziierte Belegung β. Also ist $[\delta(M)]^\beta = 1 \cup [\varphi(\mathfrak{z})]^\beta = 1$, und es ist $\chi_M(\mathfrak{x}) = 1$. Fall 3: $\mathfrak{x} \notin M$. Dann ist $[\delta(M)]^\beta = [\delta(N)]^\beta \cup [\varphi(\mathfrak{z})]^\beta = 0 \cup 0 = 0$ für jede zu $\mathfrak{x}$ assoziierte Belegung; ebenso ist $\chi_M(\mathfrak{x}) = 0$. Dies zeigt die Behauptung. Der Satz 69 ist damit bewiesen.

Definition 70 *Eine n–**Basisform** ist eine Aussage von der Form $\varphi(\vec{x})$, wo $\vec{x}$ ein Vektor aus $\{0,1\}^n$. Eine Aussage heißt in **disjunktiver Form**, falls sie eine Disjunktion von Basisformen ist.*

Korollar 71 *Jede Aussage ist äquivalent zu einer Aussage in disjunktiver Form.*

Wir können die Form für jede Funktion sogar eindeutig machen. Ist f primitiv, so ist die Aussage φ tatsächlich eindeutig definiert. Sie hat die Form $\varphi(\vec{x})$ für ein gewisses $\vec{x}$. Die $\vec{x} \in \{0,1\}^n$ sind durch die folgende Ordnung eindeutig geordnet.

$$\vec{x} < \vec{y} \text{ gdw } \text{ es existieren } \vec{x}_1, \vec{x}_2, \vec{y}_2 \text{ mit } \vec{x} = \vec{x}_1 \mathtt{0} \vec{x}_2, \vec{y} = \vec{x}_1 \mathtt{1} \vec{y}_2$$

Nun definieren wir $\delta(M)$ für $M \subseteq \{0,1\}^M$ wie folgt. Ist $M = \varnothing$, so sei $\delta(M) = \neg\top$. Ist aber $M = \{\vec{x}\}$, so sei $\delta(M) := \varphi(\vec{x})$. In allen anderen Fällen setzen wir

$$\delta(M) := (\delta(M - \{\max M\}) \vee \varphi(\max M)) \ .$$

Mithin verstehen wir unter Normalform nur noch Formeln der eben definierten Art.

Definition 72 *Es sei φ eine Aussage und n minimal mit $\mathrm{var}(\varphi) \subseteq \{\mathtt{p}_i : i < n\}$. Dann heißt die **disjunktive Normalform** von φ die eindeutig bestimmte Aussage $\delta(M)$, mit $M \subseteq \{0,1\}^n$ mit $\delta(M) \equiv \varphi$.*

Es ist dann klar, dass je zwei verschiedene Normalformen nicht mehr äquivalent sind. Dies erlaubt uns, die Anzahl der nicht äquivalenten Aussagen in den n Variablen $\{\mathtt{p}_i : i < n\}$ anzugeben. Sie stehen nämlich in 1–1–Beziehung zu den booleschen Funktionen $f : \{0,1\}^n \to \{0,1\}$. Es ist $\{0,1\}^n = 2^n$, und die Mächtigkeit dieser Menge ist genau 2^n. Es gibt also 2^n n–Tupel. Die Menge der Funktionen von 2^n nach 2 hat ihrerseits die Mächtigkeit 2^{2^n}.

Satz 73 *Es gibt genau 2^{2^n} verschiedene Normalformen über der Menge $\{\mathtt{p}_i : i < n\}$.*

Wir ziehen aus diesen Sätzen einige Folgerungen. Zunächst wollen wir jetzt die Begründung nachschieben, warum wir die Funktionen $\rightarrow$ und $\leftrightarrow$ sowie andere nicht schon in die Basissignatur aufgenommen haben. Die Antwort ist: weil sie entbehrlich sind. Wir können sie mit Hilfe der bereits vorhandenen Funktionen definieren. Sehen wir uns an, wie das gehen kann. Hier sind einige 2–stellige Funktionen, die des öfteren verwendet werden.

$$\begin{array}{c|cc} \rightarrow & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 1 \end{array} \qquad \begin{array}{c|cc} \leftrightarrow & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 1 \end{array} \qquad \begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$

$$\begin{array}{c|cc} \uparrow & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & 0 \end{array} \qquad \begin{array}{c|cc} \downarrow & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 0 \end{array}$$

Ein Hinweis: wir werden in Zukunft nicht mehr strikt zwischen den Funktionen und den Junktoren unterscheiden. Dies erlaubt uns, mit weniger Symbolen auszukommen. So kann man zwar zwischen dem Junktor $\rightarrow$ und der oben angegeben Funktion unterscheiden, indem man sie zum Beispiel $\supset$ nennt, aber dieser Unterschied sollte hinreichend geläufig sein, dass er keine weiteren Zwischenfälle verursacht.

Wir nennen $\rightarrow$ die **Implikation**, $\leftrightarrow$ die **Äquivalenz**, $+$ die **Addition** oder auch **exklusives Oder** oder auch **xor**; schließlich ist $\uparrow$ die sogenannte **nand**–Funktion und $\downarrow$ die sogenannte **nor**–Funktion. Wir können die oben angegebene Prozedur benutzen, um die Funktionen durch unsere vier Funktionen 1, $-$, $\cap$ und $\cup$ zu definieren. Man beachte nämlich Folgendes. Falls φ die Funktion f repräsentiert, so lässt sich aus φ geradeswegs eine Funktion $\{\varphi\}$ definieren:

$$\begin{array}{lcl} \{\mathtt{p}_i\} & := & x_i \\ \{\top\} & := & 1 \\ \{\mathtt{(}\neg\vec{x}\mathtt{)}\} & := & -\{\vec{x}\} \\ \{\mathtt{(}\vec{x} \wedge \vec{y}\mathtt{)}\} & := & \{\vec{x}\} \cap \{\vec{y}\} \\ \{\mathtt{(}\vec{x} \vee \vec{y}\mathtt{)}\} & := & \{\vec{x}\} \cup \{\vec{y}\} \end{array}$$

So erhalten wir für $\rightarrow$ etwa

$$\delta_{\rightarrow} = \mathtt{(((}\neg\mathtt{p}_0\mathtt{(} \wedge \mathtt{p}_1\mathtt{))} \vee \mathtt{((}\neg\mathtt{p}_0\mathtt{)} \wedge \mathtt{p}_1\mathtt{))} \vee \mathtt{(p}_0 \wedge \mathtt{p}_1\mathtt{))}$$

Und damit (unter Weglassen einiger Klammern ...)

$$\{\delta_\to\} = (-x_0 \cap -x_1) \cup (-x_0 \cap x_1) \cup (x_0 \cap x_1)$$

Es existiert aber eine kompaktere Darstellung, nämlich

$$\gamma_\to := \texttt{((}\neg\texttt{p}_0\texttt{)} \vee \texttt{p}_1\texttt{)}$$

Ebenso finden wir

$$\begin{array}{lcl}
\gamma_\leftrightarrow & := & \texttt{((p}_0 \wedge \texttt{p}_1\texttt{)} \vee \texttt{((}\neg\texttt{p}_0\texttt{)} \vee \texttt{(}\neg\texttt{p}_1\texttt{)))} \\
\gamma_+ & := & \texttt{((p}_0 \wedge \texttt{(}\neg\texttt{p}_1\texttt{))} \vee \texttt{((}\neg\texttt{p}_0\texttt{)} \wedge \texttt{p}_1\texttt{))} \\
\gamma_\uparrow & := & \texttt{(}\neg\texttt{(p}_0 \wedge \texttt{p}_1\texttt{))} \\
\gamma_\downarrow & := & \texttt{(}\neg\texttt{(p}_0 \vee \texttt{p}_1\texttt{))}
\end{array}$$

Dies zeigt uns, wie sich die Junktoren $\to$, $\leftrightarrow$ und so weiter durch einen komplexen Ausdruck in $\top$, $\neg$, $\wedge$ und $\vee$ ersetzen lassen.

In der Tat kann man wahlweise $\wedge$ und $\vee$ auch noch entbehren. Denn wir haben

$$\begin{array}{lcl}
\texttt{(p}_0 \wedge \texttt{p}_1\texttt{)} & \equiv & \texttt{(}\neg\texttt{((}\neg\texttt{p}_0\texttt{)} \vee \texttt{(}\neg\texttt{p}_1\texttt{)))} \\
\texttt{(p}_0 \vee \texttt{p}_1\texttt{)} & \equiv & \texttt{(}\neg\texttt{((}\neg\texttt{p}_0\texttt{)} \wedge \texttt{(}\neg\texttt{p}_1\texttt{)))}
\end{array}$$

Zum Schluss wollen wir noch auf ein wichtiges Thema eingehen, nämlich den Unterschied zwischen einem Symbol des Alphabets und einem definierten Symbol. Zum Beispiel können wir definieren, dass

$$\varphi \to \chi := \texttt{(}\neg\texttt{(}\varphi \wedge \texttt{(}\neg\chi\texttt{)))}$$

Dann ist $\varphi \to \chi$ kein genuines sprachliches Objekt, sondern es ist lediglich eine Art, sich kürzer oder anders auszudrücken. Auf der anderen Seite hätten wir auch ein anderes Alphabet annehmen können, etwa $Q := P \cup \{\to\}$, sodass jetzt mit zwei Termen $\vec{x}$, $\vec{y}$ auch $\texttt{(}\vec{x} \to \vec{y}\texttt{)}$ ein Term ist, und dessen Wahrheit wie oben definiert wird. Dann haben wir anstelle der definitorischen Setzung jetzt eine Äquivalenz

$$\texttt{(}\varphi \to \chi\texttt{)} \equiv \texttt{(}\neg\texttt{(}\varphi \wedge \texttt{(}\neg\chi\texttt{)))}$$

Dies ist eine nicht banale Tatsache, denn sie muss durch Vergleich der Wahrheitsdefinitionen ermittelt werden. Ist $\to$ dagegen definiert, dann ist die Äquivalenz völlig trivial: es gilt sogar Gleichheit der Zeichenketten.

9. Teil: Aussagenlogik IV: Ein Gleichungskalkül

In diesem Teil wollen wir einen Gleichungskalkül vorstellen, welcher uns vollständige Auskunft darüber gibt, ob für gegebene Aussagen φ und χ gilt $\varphi \equiv \chi$. Es bezeichnet H die Menge der folgenden Paare, wobei $\vec{x}$, $\vec{y}$ und $\vec{z}$ alle wohlgeformten Zeichenfolgen durchlaufen, und $\vec{x} \approx \vec{y}$ anstelle von $\langle \vec{x}, \vec{y} \rangle \in H$ steht.

$$
\begin{array}{lllll}
(\mathrm{as}\wedge) & (\vec{x} \wedge (\vec{y} \wedge \vec{z})) & \approx & ((\vec{x} \wedge \vec{y}) \wedge \vec{z}) \\
(\mathrm{as}\vee) & (\vec{x} \vee (\vec{y} \vee \vec{z})) & \approx & ((\vec{x} \vee \vec{y}) \vee \vec{z}) \\
(\mathrm{ko}\wedge) & (\vec{x} \wedge \vec{y}) & \approx & (\vec{y} \wedge \vec{x}) \\
(\mathrm{ko}\vee) & (\vec{x} \vee \vec{y}) & \approx & (\vec{y} \vee \vec{x}) \\
(\mathrm{id}\wedge) & (\vec{x} \wedge \vec{x}) & \approx & \vec{x} \\
(\mathrm{id}\vee) & (\vec{x} \vee \vec{x}) & \approx & \vec{x} \\
(\mathrm{ds}\vee) & (\vec{x} \wedge (\vec{y} \vee \vec{z})) & \approx & ((\vec{x} \wedge \vec{y}) \vee (\vec{x} \wedge \vec{z})) \\
(\mathrm{ds}\wedge) & (\vec{x} \vee (\vec{y} \wedge \vec{z})) & \approx & ((\vec{x} \vee \vec{y}) \wedge (\vec{x} \vee \vec{z})) \\
(\mathrm{dm}\wedge) & (\neg(\vec{x} \wedge \vec{y})) & \approx & ((\neg\vec{x}) \vee (\neg\vec{y})) \\
(\mathrm{dm}\vee) & (\neg(\vec{x} \vee \vec{y})) & \approx & ((\neg\vec{x}) \wedge (\neg\vec{y})) \\
(\top\wedge) & (\vec{x} \wedge \top) & \approx & \vec{x} \\
(\top\vee) & (\vec{x} \vee \top) & \approx & \top \\
(\mathrm{dn}) & (\neg(\neg\vec{x})) & \approx & \vec{x} \\
(\mathrm{cn}) & (\vec{x} \vee (\neg\vec{x})) & \approx & \top
\end{array}
$$

Man nennt (as$\wedge$) das **Assoziativgesetz bezüglich** $\wedge$, (ko$\wedge$) das **Kommutativgesetz bezüglich** $\wedge$, (id$\wedge$) das **Idempotenzgesetz bezüglich** $\wedge$, (ds$\wedge$) das **Distributivgesetz bezüglich** $\wedge$ **und** $\vee$, (dm$\wedge$) das **de Morgansche Gesetz bezüglich** $\wedge$, (dn) das **Gesetz der doppelten Negation**.

Es ist leicht zu sehen, dass diese Gleichungen gültig sind in dem Sinne, dass aus $\vec{x} \approx \vec{y}$ auch folgt $\vec{x} \equiv \vec{y}$. Wir wollen in diesem Abschnitt beweisen, dass sie aber auch vollständig sind. Dabei muss noch geklärt werden, in welchem Sinne diese Gleichungen zu verwenden sind. Dazu machen wir uns klar, wie Gleichungen im Rechnen mit Zahlen verwendet werden dürfen. Sie erlauben uns nicht nur, einen Term auf der einen Seite durch den entsprechenden Term auf der anderen Seite zu ersetzen, sondern wir dürfen dies auch bei Teiltermen tun. Wir haben also nicht nur $(5 + 7) + 1 = 1 + (5 + 7)$, weil dies eine Instanz des Kommutativgesetzes der Addition ist (welches lautet $x+y = y+x$), sondern wir dürfen auch $(5+7)+1 = (7+5)+1$ schließen, weil auch die Gleichung $5 + 7 = 7 + 5$ eine Instanz des Kommutativgesetzes ist. Also dürfen wir auch Teilterme durch entsprechende gleiche andere Teilterme ersetzen. Wir nennen die Gleichung $(5 + 7) + 1 = (7 + 5) + 1$ eine *Subinstanz* der Kommutativitätsgleichung. Dies gießen wir in eine Definition.

Definition 74 *Wir sagen, ein Paar $\langle\vec{x},\vec{y}\rangle$ sei* ***Subinstanz*** *einer Gleichung $\vec{u} \approx \vec{v}$ aus H, falls gilt: (a) es existiert ein Teiltermvorkommen von $\vec{u}$ in $\vec{x}$, welches auch Teiltermvorkommen von $\vec{v}$ in $\vec{y}$ ist oder (b) es existiert ein Teiltermvorkommen von $\vec{v}$ in $\vec{x}$, welches auch Teiltermvorkommen von $\vec{u}$ in $\vec{y}$ ist. Wir schreiben $\vec{x} \approx \vec{y}$ auch in dem Fall, wo $\langle\vec{x},\vec{y}\rangle$ lediglich Subinstanz einer Gleichung von H ist.*

Wir wollen uns davon überzeugen, dass die gegebene Definition das Gewünschte leistet. Im Falle (a) haben wir eine Zerlegung $\vec{x} = \vec{x}_1\vec{u}\vec{x}_2$, wo $\vec{u}$ ein Term ist und das Vorkommen $\langle\vec{x}_1,\vec{x}_2\rangle$ ein Vorkommen von $\vec{u}$ in $\vec{x}$als Teilterm. In diesem Fall sagt (a), dass $\vec{y} = \vec{x}_1\vec{v}\vec{x}_2$, und das Paar $\langle\vec{x}_1,\vec{x}_2\rangle$ ist nunmehr ein Vorkommen von $\vec{v}$ als Teilterm in $\vec{y}$. Analog für Fall (b). In der Tat überlege man sich, dass die Ersetzung eines Vorkommens eines Terms durch einen beliebigen anderen Term wieder einen Term liefert.

Definition 75 *Wir sagen, $\langle\vec{x},\vec{y}\rangle$ ist* ***ableitbar****, falls es eine endliche Folge $\vec{x}_i$, $i < n+1$, gibt derart, dass $\vec{x}_0 = \vec{x}$, $\vec{x}_n = \vec{y}$, und für alle $i < n$ ist $\vec{x}_i \approx \vec{x}_{i+1}$ eine Subinstanz einer Gleichung aus H. Ist $\langle\vec{x},\vec{y}\rangle$ ableitbar, so schreiben wir $\vec{x} \approx \vec{y}$.*

In der Mathematik hat man eine eigene Begriffsbildung geschaffen für Äquivalenzrelationen, die von Gleichungen in dieser Weise erzeugt werden. Obwohl dieser Begriff sehr allgemein definiert werden kann, wollen wir hier nur den Spezialfall der Junktoren $\top$, $\neg$, $\wedge$ und $\vee$ betrachten.

Definition 76 *Eine Relation $\Theta \subseteq Aus \times Aus$ heißt* ***Kongruenzrelation****, falls gilt:*

1. *Θ ist eine Äquivalenzrelation.*
2. *Ist $\vec{x}\,\Theta\,\vec{y}$, so auch $(\neg\vec{x})\;\Theta\;(\neg\vec{y})$.*
3. *Ist $\vec{x}_1\,\Theta\,\vec{y}_1$ und $\vec{x}_2\,\Theta\,\vec{y}_2$, so auch $(\vec{x}_1 \wedge \vec{x}_2)\;\Theta\;(\vec{y}_1 \wedge \vec{y}_2)$.*
4. *Ist $\vec{x}_1\,\Theta\,\vec{y}_1$ und $\vec{x}_2\,\Theta\,\vec{y}_2$, so auch $(\vec{x}_1 \vee \vec{x}_2)\;\Theta\;(\vec{y}_1 \vee \vec{y}_2)$.*

Proposition 77 *$\equiv$ ist eine Kongruenzrelation.*

Beweis. $\equiv$ ist sicher eine Äquivalenzrelation. Dass sie auch eine Kongruenzrelation ist, wollen wir jetzt zeigen. Es sei $\vec{x} \equiv \vec{y}$ und β eine Belegung. Wir haben dann

$$[(\neg\vec{x})]^\beta = -[\vec{x}]^\beta = -[\vec{y}]^\beta = [(\neg\vec{y})]^\beta \ .$$

β war beliebig. Also gilt $(\neg\vec{x}) \equiv (\neg\vec{y})$. Nun sei $\vec{x}_1 \equiv \vec{y}_1$ und $\vec{x}_2 \equiv \vec{y}_2$. Dann ist

$$[(\vec{x}_1 \wedge \vec{x}_2)]^\beta = [\vec{x}_1]^\beta \cap [\vec{x}_2]^\beta = [\vec{y}_1]^\beta \cap [\vec{y}_2]^\beta = [(\vec{y}_1 \wedge \vec{y}_2)]^\beta \ .$$

β war beliebig. Also gilt $(\vec{x}_1 \wedge \vec{x}_2) \equiv (\vec{y}_1 \wedge \vec{y}_2)$. Ebenso für $\vee$. Q. E. D.

Es ist $\approx$ nach Definition die kleinste Kongruenzrelation, welche H enthält. Da jede der obigen Gleichungen gültig ist und $\equiv$ auch Kongruenzrelation ist, haben wir $\approx \subseteq \equiv$. Nun gilt es zu zeigen, dass $\equiv \subseteq \approx$. Dazu genügt es zu zeigen, dass für eine beliebige disjunktive Normalform $\vec{y}$ von $\vec{x}$ gilt $\vec{x} \equiv \vec{y}$. Dies wollen wir allerdings nur skizzieren.

Wir beginnen mit einem einfachen Spezialfall. Es sei von jetzt ab $\bot := \neg\top$.

Lemma 78 *Es sei $\vec{x}$ **konstant**, das heißt eine Formel ohne Variable. Dann gilt $\vec{x} \approx \top$ oder $\vec{x} \approx \bot$.*

Lemma 79 *Es gilt $(\vec{x} \wedge \bot) \approx \bot$ sowie $(\vec{x} \vee \bot) \approx \vec{x}$.*

Die Beweise sind einfach und als Übung überlassen. Dies dient zu einer ersten Vereinfachung. Es heiße $\vec{x}$ **konstantenfrei**, falls $\vec{x}$ $\top$ nicht enthält. Dann gilt:

Proposition 80 *Zu jedem $\vec{x}$ gilt:*

1. *Es existiert ein konstantenfreies $\vec{y}$ mit $\vec{y} \approx \vec{x}$ oder*
2. *$\vec{x} \approx \top$ oder*
3. *$\vec{x} \approx \bot$.*

Wir gehen deswegen im Folgenden davon aus, dass $\vec{x}$ entweder $\top$ oder $\bot$ oder aber konstantenfrei ist.

Zunächst betrachten wir die Gesetze (dm$\wedge$) und (dm$\vee$). Es heißt $\vec{x}$ ein Literal, falls $\vec{x} = \top$, $\vec{x} = (\neg\top)$, falls $\vec{x}$ eine Variable ist oder von der Form $(\neg\vec{y})$, wo $\vec{y}$ eine Variable. $\vec{x}$ heißt **halbeinfach**, falls die einzigen Teilformeln mit $\neg$ als Hauptsymbol Literale sind.

Lemma 81 *Für jede wohlgeformte Zeichenkette $\vec{x}$ existiert eine halbeinfache Formel $\vec{y}$ mit $\vec{x} \approx \vec{y}$.*

Beweis. Wir zeigen: zu jeder Formel $\vec{x}$ der Form $(\neg\vec{z})$ existiert ein halbeinfaches $\vec{y}$ mit $\vec{x} \approx \vec{y}$. Dies genügt als Nachweis. Denn sei $\vec{x}$ beliebig. Ist dann $\vec{x}$ nicht halbeinfach, wähle ein Vorkommen $\langle \vec{v}, \vec{w} \rangle$ von einem maximalen Teilterm der Form $(\neg\vec{z})$ für ein $\vec{z}$. Dann existiert ein halbeinfaches $\vec{y}$ mit $(\neg\vec{z}) \approx \vec{y}$. Dann ist $\vec{x}_1 = \vec{v}\vec{y}\vec{w}$ eine wohlgeformte Zeichenkette, $\vec{x}_1 \approx \vec{x}$, und es enthält $\vec{x}_1$ weniger nicht–halbeinfache Teilterme als $\vec{x}$. Diese Ersetzung nehmen wir so oft vor, wie wir können. Dies liefert einen halbeinfachen Term.

Nun beweisen wir die Behauptung durch Induktion über die Länge von $\vec{z}$. Ist $\vec{z}$ eine Variable oder $\vec{x} = \top$, so ist $\vec{x}$ halbeinfach, und da $\vec{x} \approx \vec{x}$, ist die Behauptung gezeigt. Ist $\vec{z} = (\neg\vec{u})$, so existiert nach Induktionsvoraussetzung ein halbeinfaches $\vec{w}$ mit $\vec{w} \approx \vec{u}$. Nun gilt gleichfalls $\vec{x} \approx \vec{u}$, wegen (dn). Ist nun $\vec{z} = (\vec{u}_1 \vee \vec{u}_2)$, so gilt $\vec{x} \approx ((\neg\vec{u}_1) \wedge (\vec{u}_1))$, wegen (dm$\vee$). Nach Induktionsvoraussetzung existieren halbeinfache $\vec{w}_1$ und $\vec{w}_2$ mit $(\neg\vec{u}_1) \approx \vec{w}_1$ und $(\neg\vec{u}_2) \approx \vec{w}_2$. Dann ist $(\vec{w}_1 \wedge \vec{w}_2)$ ebenfalls halbeinfach, und $(\vec{w}_1 \wedge \vec{w}_2) \approx \vec{x}$. Analog der Fall, wo $\vec{x}$ das Hauptsymbol $\wedge$ hat. Q. E. D.

Nun betrachten wir die Gesetze (as$\wedge$) sowie (as$\vee$). Wie in Teil 5 bereits geschildert, folgt aus diesen Gesetzen, dass wir die Klammerung bei gleichartigen Operationen fallenlassen können. Wir schreiben also ab jetzt

$$(\vec{x}_0 \wedge \vec{x}_1 \wedge \ldots \wedge \vec{x}_{n-1})$$

anstelle einer beliebig geklammerten Version. Als Zweites betrachten wir (ko$\wedge$) und (id$\wedge$). Wir behaupten hier ohne Beweis, dass aus (ko$\wedge$) schon folgt, dass für eine beliebige Umnummerierung $\sigma : n \to n$ gilt:

$$(\vec{x}_0 \wedge \vec{x}_1 \wedge \ldots \wedge \vec{x}_{n-1}) \approx (\vec{x}_{\sigma(0)} \wedge \vec{x}_{\sigma(1)} \wedge \ldots \wedge \vec{x}_{\sigma(n-1)})$$

(id$\wedge$) bedeutet, dass wir mehrfache Vorkommen desselben Terms eliminieren dürfen.

Wir weisen darauf hin, dass Abkürzungen wie die folgenden davon Gebrauch machen:

$$\bigwedge_{i<n} \vec{x}_i := (\vec{x}_0 \wedge \vec{x}_1 \wedge \ldots \wedge \vec{x}_{n-1})$$

Ferner definiert man auch gerne so

$$\bigwedge_{\sigma \in A} \vec{x}_\sigma$$

Dies ist als Zeichenkette nicht eindeutig definiert. Aber wenn wir nur bis Äquivalenz rechnen wollen, so genügt diese Darstellung.

Nun nehmen wir uns die Regeln (ds$\wedge$) und (ds$\vee$) vor. Es heiße eine Formel **einfach**, falls sie halbeinfach ist und jede Formel, welche $\wedge$ als Hauptymbol besitzt, kein Vorkommen von $\vee$ enthält. Unter Verwendung (ds$\wedge$) und (ds$\vee$) zeigt man (durch Induktion), dass zu jeder halbeinfachen Zeichenkette $\vec{x}$ eine einfache Zeichenkette $\vec{y}$ existiert mit $\vec{x} \approx \vec{y}$. Dazu erst einmal Folgendes, allgemeines Distributivgesetz:

Lemma 82

$$\begin{array}{rcl} (\bigwedge_{i<n} \varphi_i) \vee \chi & \approx & \bigwedge_{i<n}(\varphi \vee \chi) \\ (\bigvee_{i<n} \varphi_i) \wedge \chi & \approx & \bigvee_{i<n}(\varphi \wedge \chi) \end{array}$$

Nun zeigen wir

Satz 83 *Zu jeder Formel $\vec{x}$ existiert ein einfaches $\vec{z}$ mit $\vec{x} \approx \vec{z}$.*

Beweis. Nach Lemma 81 können wir uns auf den Fall beschränken, wo $\vec{x}$ bereits halbeinfach ist. Nun sei also $\vec{x}$ halbeinfach. Ist $\vec{x}$ Literal, so ist nichts zu zeigen. Sei $\vec{x} = (\vec{x}_1 \vee \vec{x}_2)$. Dann existieren nach Induktionsvoraussetzung einfache $\vec{y}_1$ und $\vec{y}_2$ mit $\vec{x}_1 \approx \vec{y}_1$ und $\vec{x}_2 \approx \vec{y}_2$. Dann gilt $(\vec{x}_1 \vee \vec{x}_2) \approx (\vec{y}_1 \vee \vec{y}_2)$. Nun sei $\vec{x} = (\vec{x}_1 \wedge \vec{x}_2)$. Gemäß Induktionsvoraussetzung ist

$$\begin{array}{l} \vec{x}_1 \approx \bigvee_{i<m} \vec{u}_i \\ \vec{x}_2 \approx \bigvee_{i<n} \vec{v}_i \end{array}$$

für gewisse einfache $\vec{u}_i$, $i < m$, und $\vec{v}_i$, $i < n$, welche frei von $\vee$ sind. Dann gilt nach dem allgemeinen Distributivgesetz

$$(\vec{x}_1 \wedge \vec{x}_2) \approx \bigvee_{i<m,j<n} (\vec{u}_i \wedge \vec{v}_j)$$

Diese Formel ist einfach. Q. E. D.

Nun haben wir aber immer noch nicht unsere gewünschte Normalform. Wir werden jetzt zeigen, dass zu jedem $\vec{x}$ eine Normalform $\vec{y}$ existiert mit $\vec{y} \approx \vec{x}$. Die Form die wir bisher erreicht haben, ist eine Disjunktion von Konjunktionen von Literalen. Sei $\vec{x}$ eine Aussage in den Variablen $\{\mathsf{p}_i : i < n\}$ n minimal. Wir bringen $\vec{x}$ zunächst auf einfache Form, $\vec{x} = \bigvee_{j<m} \vec{z}_j$, wo $\vec{z}_j$ eine Konjunktion von Literalen ist für jedes $j < m$. Wir nehmen uns $\vec{z}_j$ vor. Wir wissen schon, dass wir mehrfach auftretende Konjunktionsglieder weglassen dürfen (wegen (ko$\wedge$) und (id$\wedge$)), sowie, dass wir die Konjunktionsglieder beliebig ordnen dürfen. Als Letztes überlegen wir noch, dass gilt $(\vec{x} \wedge (\neg\vec{x})) \approx \bot$. Deswegen bekommen wir jetzt, dass $\vec{z}_j \approx \bigwedge_{j<m} L_j$, wo $L_j = \mathtt{p}_{f(j)}$ oder $L_j = (\neg\mathtt{p}_{f(j)})$, für eine gewisse, echt monoton ansteigende Funktion $f : m \to n$. Wir hätten allerdings gerne $m = n$. Dazu überlegen wir Folgendes. Es gilt $\varphi \wedge \chi \approx (\varphi \wedge \mathtt{p}_j \wedge \chi) \vee (\varphi \wedge \neg\mathtt{p}_j \wedge \chi)$ (wegen ($\top\wedge$), (cn), und den Distributivgesetzen). Wir können also eine nicht auftretende Variable einschmuggeln. Das bedeutet, dass wir tatsächlich annehmen dürfen, dass sie in mindestens einem Literal auftritt. Alles in allem besitzen jetzt die Disjunktionsglieder die richtige Form. (Wir dürfen sie beliebig klammern, wegen (as$\wedge$), und wir wählen die Linksklammerung.) Nun ordnen wir die Disjunktionsglieder in aufsteigender Reihenfolge, wie in der Definition der Normalform gefordert (dies ist erlaubt, wie wir schon gesehen haben). Wir klammern wir sie nach links, und erhalten nun — endlich — unsere gewünschte disjunktive Normalform.

Satz 84 *Es sei $\vec{x}$ eine Aussage, und $\vec{y}$ ihre disjunktive Normalform. Dann gilt $\vec{x} \approx \vec{y}$.*

Satz 85 *Es seien $\vec{x}$ und $\vec{y}$ Aussagen. Dann gilt $\vec{x} \approx \vec{y}$ genau dann, wenn $\vec{x} \equiv \vec{y}$.*

Beweis. Sicher folgt aus aus $\vec{x} \approx \vec{y}$ schon $\vec{x} \equiv \vec{y}$. Nun sei umgekehrt $\vec{x} \equiv \vec{y}$. Sei $\vec{u}$ die Normalform von $\vec{x}$. Diese ist dann auch Normalform von $\vec{y}$, sofern $\vec{x}$ und $\vec{y}$ denselben Variablenabschnitt belegen (das bedeutet, das größte n, für das $\mathtt{p}_{n-1}$ in $\vec{x}$ auftritt, ist gleichzeitig das größte n, sodass $\mathtt{p}_{n-1}$ in $\vec{y}$ auftritt). In diesem Fall ist $\vec{u}$ auch die Normalform von $\vec{y}$ und deswegen haben wir $\vec{x} \approx \vec{u} \approx \vec{y}$, und die Behauptung folgt. Falls nicht, so bemerken wir nur, dass $\vec{x} \approx \vec{x} \wedge (\mathtt{p}_m \vee (\neg\mathtt{p}_m))$ (ebenso für $\vec{y}$, sodass wir uns auf den ersten Fall zurückziehen können. Q. E. D.

Kommen wir noch einmal auf Kongruenzrelationen zurück. Es bezeichne

$$[\vec{x}]_\Theta := \{\vec{y} : \vec{y}\,\Theta\,\vec{x}\}$$

die sogenannte **Kongruenzklasse von $\vec{x}$**. Dann können wir anstelle mit wohlgeformten Zeichenketten auch mit deren Kongruenzklassen rechnen. Wir setzen nämlich wie folgt:

$$\begin{array}{lcl} -[\vec{x}]_\Theta & := & [(\neg\vec{x})]_\Theta \\ [\vec{x}]_\Theta \cap [\vec{y}]_\Theta & := & [(\vec{x} \wedge \vec{y})]_\Theta \\ [\vec{x}]_\Theta \cup [\vec{y}]_\Theta & := & [(\vec{x} \vee \vec{y})]_\Theta \end{array}$$

Dies ist wohldefiniert, da nach Wahl von Θ als Kongruenzrelation die Definition der Operationen nicht von den Vertretern abhängen. Nun wählen wir speziell $\Theta := \approx$. Dann ist auch $\Theta = \equiv$. Wir wissen dann, dass jede Äquivalenzklasse genau eine Normalform enthält. Es gibt also in Bezug auf n Variable jeweils 2^{2^n} verschiedene Restklassen. Diese sind auch in eineindeutiger Korrespondenz mit den n–stelligen booleschen Funktionen. Hierbei definieren wir folgende Operationen auf der Menge der booleschen Funktionen. Ist $M \subseteq \{0,1\}^n$ beliebig, so ist χ_M die Funktion, welche Elemente von M auf 1 wirft und alle anderen Elemente auf 0.

$$\begin{array}{lcl} -\chi_M & := & \chi_{-M} \\ \chi_M \cap \chi_N & := & \chi_{M\cap N} \\ \chi_M \cup \chi_N & := & \chi_{M\cup N} \end{array}$$

10. Teil: Aussagenlogik V: Ein Tableaukalkül

In diesem Abschnitt wollen wir ein handliches Verfahren vorstellen, mit dem man prüfen kann, ob und durch welche Belegungen eine Aussage erfüllt wird. Natürlich kann man dies durch Ausprobieren sämtlicher Fälle tun, aber es gibt Verfahren, welche in den meisten Fällen schneller zum Ziel führen. Ein solches Verfahren ist das Tableau. Ein Tableau ist in einer Interpretation ein Baum, dessen Knoten Mengen von Aussagen enthalten, welche mit fortschreitender Berechnung immer einfacher werden. Wir wählen hier eine etwas andere Sichtweise. Ein Tableau ist eine Berechnung auf Mengen von Formelmengen, die die Erfüllbarkeit erhält und bei der die Komplexität dieser Mengen mit jedem Schritt abnimmt.

Zur besseren Übersicht wählen wir jetzt folgende Konvention. (a) Äußere Klammern werden nicht notiert, (b) Klammern werden bei einstelligen Junktoren nicht gesetzt, (c) Klammern und `p` erscheinen in einfachem Schrifttyp, nicht im Schreibmaschinentyp, (d) anstelle von $\vec{x}$ und $\vec{y}$ treten jetzt kleine griechische Buchstaben: φ, χ, ψ, (e) für Mengen von Formeln nehmen wir große griechische Buchstaben: Γ, Δ, Σ. Wir schreiben $\varphi;\psi$, $\varphi;\Gamma$ sowie $\Gamma;\Delta$, anstelle von $\{\varphi\} \cup \{\psi\}$, $\{\varphi\} \cup \Gamma$ sowie $\Gamma \cup \Delta$. Die Regeln des Kalküls sind die folgenden.

$$
\begin{array}{llll}
(\wedge\mathrm{E}) & \dfrac{\varphi \wedge \chi; \Gamma}{\varphi; \chi; \Gamma} & (\neg\wedge\mathrm{E}) & \dfrac{\neg(\varphi \wedge \chi); \Gamma}{\neg\varphi; \Gamma \mid \neg\chi; \Gamma} \\[2ex]
(\vee\mathrm{E}) & \dfrac{\varphi \vee \chi; \Gamma}{\varphi; \Gamma \mid \chi; \Gamma} & (\neg\vee\mathrm{E}) & \dfrac{\neg(\varphi \vee \chi); \Gamma}{\neg\varphi; \neg\chi; \Gamma} \\[2ex]
(\neg\mathrm{E}) & \dfrac{\neg\neg\varphi; \Gamma}{\varphi; \Gamma} & (\top\mathrm{E}) & \dfrac{\top; \Gamma}{\Gamma} \\[2ex]
(\neg\top\mathrm{E}) & \dfrac{\neg\top; \Gamma}{\surd} & (\surd) & \dfrac{\varphi; \neg\varphi; \Gamma}{\surd}
\end{array}
$$

Die Idee hinter diesem Verfahren ist recht einfach. Wir wollen herausfinden, ob eine Menge von Formeln erfüllbar ist. Falls sie nicht nur Literale enthält, so können wir eine der angegebenen Regeln verwenden und das Problem in ein, höchstens zwei Teilprobleme verwandeln. Nehmen wir zum Beispiel die Menge $p_0 \vee (\neg p_1 \wedge p_2); \neg(p_1 \vee p_0)$. Wenden wir auf sie die Regel $(\neg\vee\mathrm{E})$ an, so erhalten wir die Menge $p_0 \vee (\neg p_1 \wedge p_2); \neg p_1; \neg p_0$. Wenden wir darauf wiederum die Regel $(\vee\mathrm{E})$ an, so erhalten wir zwei Mengen: $p_0; \neg p_1; \neg p_0$ und $\neg p_1 \wedge p_2; \neg p_1; \neg p_0$. Die erste Menge enthält eine Formel und ihre Negation. Auf sie ist die Regel $(\surd)$ anwendbar und liefert $\surd$. Die zweite Menge kann noch mit Hilfe von $(\wedge\mathrm{E})$ zu $\neg p_1; p_2; \neg p_0$ weiterverarbeitet werden. Diese Menge enthält nur noch Literale. Sie ist erfüllbar, wie man unmittelbar sieht. Wir müssen einfach $\beta(p_0) = 0$, $\beta(p_1) = 0$ sowie $\beta(p_2) = 1$ haben.

Zunächst einmal gilt für die Regeln Folgendes.

1. Hat eine Regel die Form $\dfrac{\Gamma}{\surd}$, so ist Γ nicht erfüllbar.
2. Hat eine Regel die Form $\dfrac{\Gamma}{\Delta}$, so erfüllt eine Belegung β genau dann Γ, wenn sie Δ erfüllt.
3. Hat eine Regel die Form $\dfrac{\Gamma}{\Delta_1 \mid \Delta_2}$, so erfüllt eine Belegung β genau dann Γ, wenn sie Δ_1 oder Δ_2 erfüllt.

Das bedeutet, dass wir beim Übergang von der Menge Γ zu den Mengen unter dem Strich keinerlei Verlust haben. Genau dann ist Γ durch eine Belegung erfüllt, wenn eine der unteren Mengen durch β erfüllt ist. Ferner gilt $\surd$ als nicht erfüllbar. Das Tableau schreitet also fort, indem es zu der Ausgangsmenge Γ_0 immer neue Mengen erzeugt, wobei wir normalerweise nicht eine Menge, sondern mehrere haben, welche wir wahlweise erfüllen können. Wir schreiben dann $\Delta_0 \mid \Delta_1 \mid \ldots | \Delta_{n-1}$ um zu signalisieren, dass wahlweise Δ_i, $i < n$, erfüllt werden darf. Diese Schreibweise ist allerdings nur eine optische Variante von

$$\{\Delta_0, \Delta_1, \ldots, \Delta_{n-1}\}$$

Eine Tableauregel wird dann stets auf eines der Δ_i angewendet und erzeugt ein oder zwei neue Mengen, die an die Stelle von Δ_i treten. Nun wollen wir uns als erstes vergewissern, dass der Kalkül stets terminiert, und zwar egal in welcher Reihenfolge man die Regeln anwendet.

Definition 86 *Es sei $\sharp\Delta$ die Anzahl der Junktoren in Δ. Dies ist definiert durch*

$$\begin{array}{lcl}
\sharp(p_i) & := & 0 \\
\sharp(\top) & := & 1 \\
\sharp(\neg\varphi) & := & \sharp(\varphi) + 1/2 \\
\sharp(\varphi \wedge \chi) & := & \sharp(\varphi) + \sharp(\chi) + 1 \\
\sharp(\varphi \vee \chi) & := & \sharp(\varphi) + \sharp(\chi) + 1 \\
\sharp(\Gamma) & := & \sum_{\gamma \in \Gamma} \sharp(\gamma)
\end{array}$$

Lemma 87 *Es sei $\dfrac{\Gamma}{\Delta}$ eine Regel. Dann ist $\sharp(\Gamma) > \sharp(\Delta)$. Es sei $\dfrac{\Gamma}{\Delta_1 \mid \Delta_2}$ eine Regel. Dann ist $\sharp(\Gamma) > \sharp(\Delta_1), \sharp(\Delta_2)$.*

Nun definieren wir für $\mathfrak{D} := \Delta_0 \mid \Delta_1 \ldots \mid \Delta_{n-1}$:

$$\clubsuit\,\mathfrak{D} := \sum_{i<n} 3^{\sharp(\Delta_i)}$$

Wir behaupten: in jedem Schritt nimmt $\clubsuit\,\mathfrak{D}$ echt ab. Dies läßt sich einfach zeigen. Entweder es wird Δ_i durch eine Menge Δ_i' ersetzt, und dann ist $\sharp(\Delta_i') < \sharp(\Delta_i)$, und so erst recht $3^{\sharp(\Delta_i')} < 3^{\sharp(\Delta)}$. Oder es wird Δ_i durch zwei Mengen Δ_i' und Δ_i'' ersetzt, und dann ist $3^{\sharp(\Delta_i')} + 3^{\sharp(\Delta_i'')} \leq 2 \cdot 3^{\sharp(\Delta_i)-1} < 3^{\sharp(\Delta_i)}$.

Lemma 88 *Genau dann ist keine Tableauregel auf $\mathfrak{D}$ anwendbar, wenn $\mathfrak{D}$ nur aus Literalen besteht und nicht gleichzeitig p_i und $\neg p_i$ für eine Variable p_i enthält.*

Ist Γ eine Folge von Literalen dieser Art, so können wir sofort eine Bedingung an die Belegung angeben, welche Γ erfüllt. Es muss $\beta(p_i) = 1$ sein, wenn $p_i \in \Gamma$, und es muss $\beta(p_i) = 0$ sein, wenn $\neg p_i \in \Gamma$. Ist $\Gamma \neq \surd$ und ist keine Regel mehr anwendbar, so ist Γ erfüllbar. Für $\mathfrak{D} = \Delta_0 \mid \Delta_1 \mid \dots \mid \Delta_{n-1}$ sagen wir, β **erfülle** $\mathfrak{D}$, falls es ein Δ_i, $i < n$, erfüllt. Geht also $\mathfrak{D}'$ aus $\mathfrak{D}$ durch Anwendung einer Tableauregel hervor, so erfüllt β $\mathfrak{D}'$ genau dann, wenn es $\mathfrak{D}$ erfüllt. Falls keine Regel mehr anwendbar ist, enthalten alle $\mathfrak{D}_i$ nur noch Literale, und die Bedingung an β ist leicht zu finden. Insbesondere ist $\mathfrak{D}$ genau dann erfüllbar, falls $\mathfrak{D} \neq \surd$.

Eine **Berechnung** ist eine Folge $\langle \mathfrak{D}_i : i < \ell + 1 \rangle$ derart, dass $\mathfrak{D}_{i+1}$ aus $\mathfrak{D}_i$ $(i < \ell)$ durch Anwendung einer Tableauregel auf einer der in $\mathfrak{D}_i$ enthaltenen Mengen entsteht. Wir sagen dann, die Berechnung habe die **Länge** ℓ.

Satz 89 *Es sei* Γ *eine beliebige Formelmenge. Dann hat jede Berechnung, welche mit* Γ *beginnt, höchstens die Länge* $3^{\sharp(\Gamma)}$.

Die Schranke ist zwar sehr schlecht und lässt sich erheblich verbessern, sie soll aber erst einmal für unsere Zwecke genügen. Denn sie beweist immerhin, dass nicht nur irgendeine terminierende Berechnung existiert, sondern auch, dass jede Berechnung terminiert. Natürlich gibt es schnellere und langsamere Berechnungen, und man kann sich leicht überlegen, dass es sinnvoll ist, verzweigende Regeln so lange wie möglich zu verzögern. Aber dennoch terminiert jede Berechnung.

Satz 90 *Genau dann ist* Γ *nicht erfüllbar, wenn es eine Berechnung gibt, die mit* Γ *beginnt und in* $\surd$ *endet.*

Anstelle von Tableaukalkülen axiomatisiert man auch gerne die Beziehung

$\vDash$. Es gilt, wie man sich leicht überzeugt, Folgendes.

$$
(\mathrm{I}\wedge)\quad \frac{\Gamma \vDash \varphi \qquad \Gamma \vDash \chi}{\Gamma \vDash \varphi \wedge \chi} \qquad\qquad (\wedge\mathrm{I})\quad \frac{\Gamma; \varphi; \chi \vDash \delta}{\Gamma; \varphi \wedge \chi \vDash \delta}
$$

$$
(\mathrm{I}\vee)\quad \frac{\Gamma \vDash \varphi}{\Gamma \vDash \varphi \vee \chi} \qquad \frac{\Gamma \vDash \chi}{\Gamma \vDash \varphi \vee \chi} \qquad\qquad (\vee\mathrm{I})\quad \frac{\Gamma; \varphi \vDash \delta \quad \Gamma; \chi \vDash \delta}{\Gamma; \varphi \vee \chi \vDash \delta}
$$

$$
(\mathrm{I}\neg)\quad \frac{\Gamma; \varphi \vDash \bot}{\Gamma \vDash \neg\varphi} \qquad\qquad (\neg\mathrm{I})\quad \frac{\Gamma \vDash \varphi}{\Gamma; \neg\varphi \vDash \bot}
$$

$$
(\text{schnitt})\quad \frac{\Gamma \vDash \varphi \qquad \Delta; \varphi \vDash \chi}{\Gamma; \Delta \vDash \chi} \qquad\qquad (\text{mon})\quad \frac{\Gamma \vDash \varphi}{\Gamma; \Delta \vDash \varphi}
$$

$$
(\text{axiom})\quad \varphi \vDash \varphi \qquad\qquad (\bot\mathrm{I})\quad \bot \vDash \varphi
$$

Diese Regeln charakterisieren die Beziehung $\Gamma \vDash \varphi$ vollständig:

Satz 91 *Genau dann ist $\Gamma \vDash \varphi$, wenn wir dies mit Hilfe der obenstehenden Regeln herleiten können.*

Dabei dürfen wir also mit allen Paaren der Form $\varphi \vDash \varphi$ oder $\bot \vDash \varphi$ beginnen, und wir dürfen nur die gegebenen Regeln verwenden. Man kann zeigen, dass der Tableaukalkül und der just gegebene Kalkül aufeinander reduzierbar sind.

11. Teil: Aussagenlogik VI: Ein Hilbert–Kalkül

Wir wollen uns in diesem Teil mit der Relation $\vDash$ beschäftigen, die wir schon in Teil 6 eingeführt hatten. Es bezeichnete $\Gamma \vDash \varphi$ die Tatsache, dass jede Belegung, welche Γ erfüllt, auch φ erfüllt. Dies ist im Übrigen gleichwertig mit der Tatsache, dass $\Gamma; \neg\varphi$ nicht erfüllbar ist. Wir wollen nun die Notation noch weiter auflockern und uns frei der Junktoren $\to$, $\leftrightarrow$ und anderer bedienen, die wir ja schon eingeführt hatten. Zunächst einige Eigenschaften von $\vDash$.

Satz 92 (Deduktionstheorem) $\Gamma \vDash \varphi \to \chi$ *genau dann, wenn* $\Gamma; \varphi \vDash \chi$.

Der Beweis ist ganz leicht. Angenommen, $\Gamma; \varphi \vDash \chi$. Sei ferner β eine Belegung, die Γ erfüllt. Dann treten zwei Fälle ein. (a) β erfüllt φ nicht. Dann erfüllt β

die Formel $\varphi \to \chi$. (b) β erfüllt φ. Dann erfüllt β auch χ, nach Voraussetzung. Also erfüllt β die Formel $\varphi \to \chi$. Also gilt $\Gamma \vDash \varphi \to \chi$. Nehmen wir umgekehrt an, $\Gamma \vDash \varphi \to \chi$. Sei β eine Belegung, welche $\Gamma; \varphi$ erfüllt. Dann erfüllt sie auch χ, da sie $\varphi \to \chi$ erfüllt. Also gilt $\Gamma; \varphi \vDash \chi$.

Wir notieren einen interessanten Spezialfall. Zunächst sei bemerkt, dass $\Gamma \vDash \bot$ gleichwertig mit der Tatsache ist, dass Γ nicht erfüllbar ist. Ist $\chi = \bot$, so gilt $\neg\varphi \equiv \varphi \to \bot = \varphi \to \chi$. Also haben wir aufgrund des Deduktionstheorems: $\Gamma \vDash \neg\varphi$ ist gleichwertig mit $\Gamma; \varphi \vDash \bot$. Letzteres bedeutet schlicht, dass $\Gamma; \varphi$ nicht erfüllbar ist. Da dies für jedes φ gilt, so können wir jetzt schließen, dass $\Gamma; \neg\varphi \vDash \bot$ gleichwertig ist mit $\Gamma \vDash \neg\neg\varphi$, was wiederum nichts anderes ist als $\Gamma \vDash \varphi$. Dies ist eine Ableitung des oben schon erwähnten Sachverhalts. Wir haben dabei stillschweigend von folgendem evidenten Prinzip Gebrauch gemacht.

Proposition 93 *Es sei $\varphi \equiv \varphi'$ und $\chi \equiv \chi'$. Dann gilt $\Gamma; \varphi \vDash \chi$ genau dann, wenn $\Gamma; \varphi' \vDash \chi'$.*

Mit anderen Worten: die Folgerungsbeziehung unterscheidet nicht zwischen äquivalenten Formeln.

Ein **Hilbert–Kalkül** ist ein Axiom–Regel Kalkül, der auf einem sequentiellen Beweisverfahren beruht. Mittels gewisser Regeln wird eine Formel φ aus Γ geschlossen. Der originäre Hilbert–Kalkül benutzt nur eine Schlussregel, den sogenannten **Modus Ponens**. Eine Regel ist im allgemeinen ein Paar $\langle \Delta, \varphi \rangle$, wobei Δ eine Menge von Formeln und φ eine einzelne Formel ist. Man schreibt auch gerne $\frac{\Delta}{\varphi}$. Bespiele sind

$$\text{Modus Ponens } \frac{\varphi \to \chi \quad \varphi}{\chi} \qquad \text{Modus Tollens } \frac{\varphi \to \chi \quad \neg\chi}{\neg\varphi}$$

Diese Notation besagt nichts weiter, als dass man mittels dieser Regel von den Prämissen (den Mengen über dem Strich) auf die Konklusion (die Formel unter dem Strich) schließt. Eine Regel $\langle \Delta, \varphi \rangle$ heißt **korrekt**, falls $\Delta \vDash \varphi$. Modus Ponens und Modus Tollens sind korrekt. Ist $\Delta = \varnothing$, so spricht man von einem **Axiom**. Axiome sind also spezielle Regeln. Der Hilbert Kalkül ist ein Kalkül, in dem Modus Ponens die einzige Schlussregel ist, welche nicht Axiom ist.

Wir stellen hier einen speziellen Hilbert–Kalkül für die Aussagenlogik vor. Wir benutzen dabei $\to$, $\wedge$, $\vee$, $\neg$ und $\bot$ als Basisjunktoren. Man mache sich klar, dass die zu wählende Axiomenmenge auch von der gewählten Menge der Basisjunktoren abhängt, wir diese also explizit nennen müssen. Ferner ist die Menge Aus jetzt nicht mehr dieselbe Menge wie in der vorangegangenen Abschnitten. Sie ist die Menge der wohlgeformten Zeichenkette über der jetzt verabredeten Junktorenmenge, wobei Definitionen analog übertragen werden.

$$
\begin{array}{ll}
\text{(a0)} & p_0 \to (p_1 \to p_0) \\
\text{(a1)} & (p_0 \to (p_1 \to p_2)) \to ((p_0 \to p_1) \to (p_0 \to p_2)) \\
\text{(a2)} & ((p_0 \to p_1) \to p_0) \to p_0 \\
\text{(a3)} & \bot \to p_0 \\
\text{(a4)} & \neg p_0 \to (p_0 \to \bot) \\
\text{(a5)} & (p_0 \to \bot) \to \neg p_0 \\
\text{(a6)} & p_0 \to (p_1 \to (p_0 \wedge p_1)) \\
\text{(a7)} & (p_0 \wedge p_1) \to p_0 \\
\text{(a8)} & (p_0 \wedge p_1) \to p_1 \\
\text{(a9)} & p_0 \to (p_0 \vee p_1) \\
\text{(a10)} & p_1 \to (p_0 \vee p_1) \\
\text{(a11)} & ((p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2))
\end{array}
$$

Die Formeln (a0) – (a11) sind aus Aus. Wir definieren zunächst den Begriff einer Substitutionsinstanz.

Definition 94 *Es sei $\sigma : \mathrm{Var} \to Aus$ eine beliebige Funktion. Dann wird für $\varphi \in Var$, φ^σ induktiv wie folgt definiert.*

1. *Ist $\varphi \in Var$, so sei $\varphi^\sigma := \sigma(\varphi)$.*
2. *Ist $\varphi = \bot$, so sei $\varphi^\sigma := \bot$.*
3. *Ist $\varphi = \neg\chi$, so sei $\varphi^\sigma := \neg(\chi^\sigma)$.*
4. *Ist $\varphi = \chi_1 \wedge \chi_2$, so sei $\varphi^\sigma := \chi_1^\sigma \wedge \chi_2^\sigma$.*
5. *Ist $\varphi = \chi_1 \vee \chi_2$, so sei $\varphi^\sigma := \chi_1^\sigma \vee \chi_2^\sigma$.*
6. *Ist $\varphi = \chi_1 \to \chi_2$, so sei $\varphi^\sigma := \chi_1^\sigma \to \chi_2^\sigma$.*

*Wir sagen, χ sei eine **(Substitutions)Instanz** von φ, falls ein σ existiert mit $\chi = \varphi^\sigma$.*

In der Sprache der Algebra sagt man, die Abbildung $\varphi \mapsto \varphi^\sigma$ sei die **homomorphe Fortsetzung von σ**.

Lemma 95 *Ist φ eine Tautologie, so ist jede Instanz von φ eine Tautologie.*

Definition 96 *Ein **Beweis von** φ **aus** Γ ist eine endliche Folge $\Pi = \langle \chi_i : i < n+1 \rangle$ derart, dass (a) $\chi_n = \varphi$ und (b) für jedes $i < n+1$ ist (b1) χ_i eine Instanz einer der Formeln (a0) – (a11) oder (b2) $\chi_i \in \Gamma$ oder (b3) es gibt $j, k < i$ mit $\chi_k = \chi_j \to \chi_i$. Es ist n die **Länge** von Π. Wir schreiben $\Gamma \vdash \varphi$, falls ein Beweis von φ aus Γ existiert. Γ heißt die Menge der **Prämissen**, φ die **Konklusion**.*

Folgendes ist ein Beweis von $\neg p_0$ aus $p_4; \neg(p_0 \to p_4)$.

$$\begin{array}{lll}
(0) & p_4, & \text{Präm}, \\
(1) & \neg(p_0 \to p_4), & \text{Präm}, \\
(2) & \neg(p_0 \to p_4) \to ((p_0 \to p_4) \to \bot), & \text{(a4)}, \\
(3) & (p_0 \to p_4) \to \bot, & \text{MP} : 1, 2, \\
(4) & p_4 \to (p_0 \to p_4), & \text{(a1)}, \\
(5) & (p_0 \to p_4), & \text{MP} : 0, 4, \\
(6) & \bot, & \text{MP} : 3, 5, \\
(7) & \bot \to \neg p_0, & \text{(a3)} \\
(8) & \neg p_0, & \text{MP} : 6, 7.
\end{array}$$

Man mache sich klar, dass jedes Anfangsstück eines Beweises wieder ein Beweis ist. Dabei haben beide dieselbe Prämissenmenge, lediglich die Konklusion ist eine andere. Ziel ist folgender

Satz 97 *Genau dann ist $\Gamma \vDash \varphi$, wenn $\Gamma \vdash \varphi$, das heißt, wenn ein Beweis von φ aus Γ existiert.*

Daraus folgt unmittelbar folgende Tatsache.

Korollar 98 (Endlichkeitssatz) *Falls $\Gamma \vDash \varphi$, so existiert ein endliches $\Gamma_0 \subseteq \Gamma$ derart, dass $\Gamma_0 \vDash \varphi$.*

Beweis. Aufgrund von Satz 97 ist $\Gamma \vdash \varphi$. Dies bedeutet, dass ein Beweis Π von φ aus Γ existiert. Es sei Γ_0 die Menge aller $\gamma \in \Gamma$, welche in Π vorkommen. Γ_0 ist endlich. Ferner ist, wie man leicht sieht, Π auch ein Beweis von φ aus Γ_0, also $\Gamma_0 \vdash \varphi$. Daraus folgt nun wiederum mit Satz 97 $\Gamma_0 \vDash \varphi$. Q. E. D.

Der Beweis von Satz 97 zerfällt in zwei Teile. Der erste ist die *Korrektheit*: $\vdash \subseteq \vDash$. Diese besagt, dass alles, was im Kalkül aus Γ beweisbar ist, auch aus Γ folgt. Die zweite ist die *Vollständigkeit*: $\vDash \subseteq \vdash$. Diese besagt, dass alles, was aus Γ folgt, bereits im Kalkül ableitbar ist.

Die Relation $\vdash$ besitzt folgende Eigenschaften, welche wir schon von $\vDash$ nachgewiesen haben:

Lemma 99 *Es gilt*

1. *Ist* $\varphi \vdash \varphi$.
2. *Ist* $\Gamma \vdash \varphi$ *und* $\Gamma \subseteq \Delta$, *so auch* $\Delta \vdash \varphi$.
3. *Ist* $\Gamma \vdash \delta$ *für jedes* $\delta \in \Delta$ *und ist* $\Delta \vdash \varphi$, *so ist auch* $\Gamma \vdash \varphi$.

Beweis. (1) Setze $\Pi := \langle \varphi \rangle$. Dies ist ein Beweis von φ auf $\{\varphi\}$. (2) Ist Π ein Beweis von φ aus Γ, so auch aus jeder Menge, die Γ enthält, wie man leicht nachprüft. (3) Es seien Π_δ, $\delta \in \Delta$, Beweise von δ aus Γ und $\Xi = \langle \xi_i : i < n+1 \rangle$ ein Beweis von φ aus Δ. Ersetze in Ξ jedes Vorkommen einer Formel $\delta \in \Delta$ durch die Folge Π_δ. Nenne das Ergebnis $\Theta = \langle \vartheta_i : i < p+1 \rangle$. Θ ist ein Beweis von φ aus Γ. Denn sei ϑ_i eine Formel aus Θ. Ist $\vartheta_i \notin \Gamma$, und trifft (b1) und (b3) nicht auf sie zu, so ist sie in einer Folge der Form Π_δ, $\delta \in \Delta$, enthalten. Dann trifft auf diese Formel in Π_δ auch nicht (b1) oder (b3) zu, sodass (b2) zutrifft. Das bedeutet aber $\vartheta_i \in \Gamma$. Q. E. D.

Die letzte Eigenschaft führt im Verbund mit dem weiter unten bewiesenen Deduktionstheorem zur Transitivität des Schließens. Haben wir einmal $\Gamma \vdash \varphi$ etabliert, so dürfen wir $\Gamma \vdash \chi$ schließen, sofern $\Gamma; \varphi \vdash \chi$ gilt. Denn Letzteres hat zur Folge, dass $\Gamma \vdash \varphi \to \chi$. Und so haben wir: $\Gamma \vdash \varphi; \varphi \to \chi$ und andererseits $\varphi; \varphi \to \chi \vdash \chi$. Setze also $\Delta := \{\varphi, \varphi \to \chi\}$, und die Behauptung folgt aus dem eben bewiesenen Satz. Setze nun $\varphi \simeq \varphi'$, falls $\varphi \vdash \varphi'$ und $\varphi' \vdash \varphi$ gilt. (Es wird sich erweisen dass $\simeq = \equiv$ ist, aber das dürfen wir nicht voraussetzen.) Dann kann man Folgendes zeigen.

Lemma 100 *Es gelte* $\varphi \simeq \varphi'$ *und* $\chi \simeq \chi'$. *Dann ist* $\Gamma; \varphi \vdash \chi$ *äquivalent mit* $\Gamma; \varphi' \vdash \chi'$.

Beweis. Es sei $\Gamma; \varphi \vdash \chi$. Da $\chi \vdash \chi'$, so haben wir $\Gamma; \varphi \vdash \chi'$. Da $\varphi' \vdash \varphi$, so haben wir $\Gamma; \varphi' \vdash \varphi$, mithin $\Gamma; \varphi' \vdash \chi'$. Q. E. D.

Es gilt zum Beispiel $\neg\varphi \simeq \varphi \to \bot$, aufgrund von (a4) und (a5). Dies ist sehr nützlich.

Wir zeigen zunächst die Korrektheit. Dies tun wir durch Induktion über die Länge eines Beweises. Genauer zeigen wir durch Ordnungsinduktion: ist Π ein Beweis der Länge n von φ aus Γ, so gilt $\Gamma \vdash \varphi$. Der Fall $n = 0$ ist trivial, weil kein solcher Beweis existiert. Der Fall $n = 1$ bedeutet, dass $\varphi \in \Gamma$ oder φ Instanz eines Axioms ist. Wir überlassen an dieser Stelle dem Leser den Nachweis, dass jede Instanz eines Axioms (a0) – (a11) eine Tautologie ist. Denn dann gilt $\Gamma \vDash \varphi$ für jedes Γ. Dazu genügt aufgrund von Lemma 95 der Nachweis, dass alle (a0) – (a11) Tautologien sind. Nun sei die Behauptung für alle $i < n$ gezeigt. Wir zeigen sie nun für n. Da ein Anfangsstück eines Beweises wieder ein Beweis ist, gilt bereits nach Induktionsvoraussetzung $\Gamma \vDash \chi_i$ für alle $i < n$. Nun werden wir $\Gamma \vDash \chi_n$ zeigen. Falls der Fall (b1) wie auch der Fall (b2) eintritt, so argumentieren wir wie eben gesehen. Im Falle (b3) aber haben wir $j, k < n + 1$ derart, dass $\chi_k = \chi_j \to \chi_n$. Dann ist nach Induktionsvoraussetzung $\Gamma \vDash \chi_j; \chi_k$. Sei nun β eine Belegung, welche Γ erfüllt. Dann erfüllt sie χ_j und ferner auch $\chi_k = \chi_j \to \chi_n$ und deswegen auch χ_n. Dies zeigt $\Gamma \vDash \chi_n$. Der Kalkül ist also korrekt. Bevor wir seine Vollständigkeit beweisen, wollen wir einige weitere elementare Eigenschaften zeigen.

Lemma 101 (Deduktionstheorem) *Es gilt* $\Gamma \vdash \varphi \to \chi$ *genau dann, wenn* $\Gamma; \varphi \vdash \chi$.

Beweis. Es sei $\Gamma \vdash \varphi \to \chi$. Dann existiert ein Beweis $\Pi = \langle \sigma_i : i < n+1 \rangle$ von $\varphi \to \chi$ aus Γ. Nun setzen wir $\sigma_{n+1} := \varphi$ und $\sigma_{n+2} := \chi$. Dann ist $\Delta := \langle \sigma_i : i < n+3 \rangle$ ein Beweis von χ aus $\Gamma \vdash \varphi$. Sei umgekehrt $\Pi = \langle \sigma_i : i < n+1 \rangle$ ein Beweis von χ aus $\Gamma; \varphi$. Wir führen eine Ordnungsinduktion über n durch. Es treten drei Fälle ein. (b1) χ ist ein Axiom. Dann ist $\langle \chi, \chi \to (\varphi \to \chi), \varphi \to \chi \rangle$ ein Beweis von $\varphi \to \chi$ aus Γ. (b2) $\chi \in \Gamma$. Wie (b1). (b3) Es gibt $j, k < n+1$ derart, dass $\sigma_j = \sigma_k \to \chi$. In diesem Fall ist der Abschnitt $\Pi \restriction j + 1$ von Π (der die Formeln σ_0 bis σ_j enthlält) ein Beweis von $\Gamma; \varphi \vdash \sigma_j$ und der Abschnitt $\Pi \restriction k + 1$ (der die Formeln von σ_0 bis σ_k enthält) ein Beweis von $\sigma_k = \sigma_j \to \chi$. Jetzt existieren nach Induktionsvoraussetzungen ein Beweis Δ von $\varphi \to \sigma_j$ aus Γ und ein Beweis Σ von $\varphi \to (\sigma_j \to \chi)$ aus Γ. Jetzt bilden wir folgenden Beweis:

$$\begin{gathered}\Delta^\frown\Sigma^\frown((\varphi \to (\sigma_j \to \chi)) \to ((\varphi \to \sigma_j) \to (\varphi \to \chi))^\frown \\ (\varphi \to \sigma_j) \to (\varphi \to \chi)^\frown\varphi \to \chi\end{gathered}$$

Dies ist ein Beweis von $\varphi \to \chi$ aus Γ, wie man leicht bestätigt. Q. E. D.

Lemma 102 *1. Es gilt* $\Gamma \vdash \varphi$ *genau dann, wenn* $\Gamma; \neg\varphi \vdash \bot$

2. Es gilt $\Gamma \vdash \neg\varphi$ *genau dann, wenn* $\Gamma; \varphi \vdash \bot$.

3. Es gilt $\Gamma \vdash \varphi \wedge \chi$ *genau dann, wenn* $\Gamma \vdash \varphi$ *und* $\Gamma \vdash \chi$.

4. Es gilt $\Gamma; \varphi \vee \chi \vdash \psi$ *genau dann, wenn* $\Gamma; \varphi \vdash \psi$ *und* $\Gamma; \chi \vdash \psi$.

Beweis. (1) Hat man einen Beweis Π von φ, hänge man daran noch folgende Sequenz an: $\langle \neg\varphi \to (\varphi \to \bot), \neg\varphi, \varphi \to \bot, \bot\rangle$. Dies bildet einen Beweis von $\bot$ aus $\Gamma; \neg\varphi$. Sei umgekehrt $\Gamma; \neg\varphi \vdash \bot$, dann gilt nach dem Deduktionstheorem $\Gamma \vdash \neg\varphi \to \bot$ und so mit (a5) $\Gamma \vdash \neg\neg\varphi$. Wir zeigen jetzt $\neg\neg\varphi \vdash \varphi$. Es gilt $\neg\neg\varphi; \neg\varphi \vdash \bot$, und da $\bot \vdash \varphi$, so gilt auch $\neg\neg\varphi \vdash \neg\varphi \to \varphi$. Schließlich ist auch noch $(\varphi \to \bot) \to \neg\varphi$ eine Instanz eines Axioms, und so haben wir $\neg\neg; \varphi \to \bot \vdash \varphi$. Es ist aber $((\varphi \to \bot) \to \varphi) \to \varphi$ ebenfalls eine Instanz eines Axioms, (a2), und so $\neg\neg\varphi \vdash \varphi$. (2) Hat man $\Gamma \vdash \neg\varphi$, so auch mit (a4) $\Gamma \vdash \varphi \to \bot$, woraus mit dem Deduktionstheorem sofort $\Gamma; \varphi \vdash \bot$ folgt. Sei umgekehrt $\Gamma; \varphi \vdash \bot$. Dann gilt nach dem Deduktionstheorem $\Gamma \vdash \varphi \to \bot$ und so mit (a5) $\Gamma \vdash \neg\varphi$. (3) Es sei $\Gamma \vdash \varphi \wedge \chi$. Wir haben $\varphi \wedge \chi \vdash \varphi$, also $\Gamma; \varphi \wedge \chi \vdash \varphi$. Ebenso zeigt man $\Gamma; \varphi \wedge \chi \vdash \chi$. Q. E. D.

Nun zum Nachweis der Vollständigkeit. Dies kann auf zwei Weisen etabliert werden. Ein besteht in dem Nachweis der Regeln (I$\wedge$), ($\wedge$I), (I$\vee$), ($\vee$I), (I$\neg$), ($\neg$I), (schnitt), (mon), (axiom) und ($\bot$I) für $\vdash$. Dies ist angesichts des eben Bewiesenen leicht.

Eine andere Möglichkeit ist die Reduktion auf den Tableauklkül. Es sei $\Gamma \nvdash \varphi$. Wir werden zeigen, dass dann auch $\Gamma \nvDash \varphi$. Zunächst einmal ist $\Gamma \nvdash \varphi$ äquivalent mit $\Gamma; \neg\varphi \nvdash \bot$.

Definition 103 *Eine Menge von Formeln heißt* ***konsistent****, falls nicht jede Formel aus ihr herleitbar ist.*

Offensichtlich ist eine Menge genau dann konsistent, wenn sich $\bot$ nicht aus ihr ableiten lässt. Es ist also zu zeigen: ist eine Menge konsistent, so ist sie auch erfüllbar. Dazu wiederum zeigen wir:

1. $\Gamma; \varphi; \neg\varphi$ ist nicht konsistent.

2. Ist $\Gamma; \neg\neg\varphi$ konsistent, so auch $\Gamma; \varphi$.

3. Ist $\Gamma; \varphi \wedge \chi$ konsistent, so auch $\Gamma; \varphi; \chi$.

4. Ist $\Gamma; \neg(\varphi \wedge \chi)$ konsistent, so ist $\Gamma; \neg\varphi$ oder $\Gamma; \neg\chi$ konsistent.

5. Ist $\Gamma; \varphi \vee \chi$ konsistent, so ist $\Gamma; \varphi$ oder $\Gamma; \chi$ konsistent.

6. Ist $\Gamma; \neg(\varphi \vee \chi)$ konsistent, so auch $\Gamma; \neg\varphi; \neg\chi$.

7. Ist $\Gamma; \varphi \to \chi$ konsistent, so auch $\Gamma; \neg\varphi$ oder $\Gamma; \chi$.

8. Ist $\Gamma; \neg(\varphi \to \chi)$ konsistent, so auch $\Gamma; \varphi; \neg\chi$.

Offensichtlich haben wir es hier mit einem getreuen Abbild der Tableauregeln zu tun. Also können wir daraus schließen, dass eine konsistent Menge auch erfüllbar ist.

Wir werden den Beweis nicht in allen Einzelheiten durchführen. Dies folgt nämlich im Wesentlichen aus dem, was wir bereits gezeigt haben. Zum Beispiel die erste Behauptung: wir haben $\Gamma; \neg\varphi \vdash \neg\varphi$, und so $\Gamma; \neg\varphi \vdash \varphi \to \bot$ und daraus wiederum $\Gamma; \neg\varphi; \varphi \vdash \bot$.

12. Teil: Prädikatenlogik I: Sprachen und Modelle

Die Prädikatenlogik unterscheidet sich von Aussagenlogik dadurch, dass sie Aussagen stets als zusammengesetzte Ausdrücke behandelt. Es gibt in der Prädikatenlogik nunmehr verschiedene Sorten von Ausdrücken: Ausdrücke, welche Objekte bezeichnen (sie werden *Terme* heißen), und Ausdrücke, welche Aussagen bezeichnen (sie werden *Formeln* heißen). Die Aussagen werden ein Spezialfall der Ausdrücke letzterer Art sein. Wir werden ansonsten wie in der Aussagenlogik vergehen. Zunächst definieren wir unsere wohlgeformten Ausdrücke; anschließend definieren wir einen Modellbegriff und sagen, wann eine Formel unter einer Belegung erfüllt ist. Dies führt zu einer Logik, die wir axiomatisieren werden. Den Beweis, dass diese Axiomatisierung vollständig ist, werden wir dagegen nicht in allen Einzelheiten führen. Er ist zwar nicht übermäßig schwer, aber denoch zu umfangreich.

In der Prädikatenlogik haben wir Funktionen und Relationen und zwar so viele, wie wir möchten. Das bedeutet, dass es nicht wie in der Aussagenlogik im Wesentlichen *eine* Sprache gibt sondern sehr viele. Sie unterscheiden sich in der sogenannten *Signatur*. Eine Signatur besteht in der Angabe der Stelligkeit der Ausdrücke. Dies ist aus Programmiersprachen hinlänglich bekannt.

Wer ein neues Funktions- oder Relationssymbol einführt, muss als erstes angeben, wie viele Argumente dieses Symbol braucht (und, falls nötig, auch, welchen Typ diese Argumente haben).

Definition 104 *Eine **Signatur** ist ein Paar $\langle F, \Omega\rangle$, wobei F eine Menge ist und $\Omega : F \to \omega$ eine Funktion, die sogenannte **Signaturfunktion**. Wir nennen $\Omega(f)$, $f \in F$, die **Stelligkeit** des Symbols f.*

Um die Notation nicht zu überfrachten, werden wir in Zukunft auch das Paar $\langle F, \Omega\rangle$ mit Ω bezeichnen.

Definition 105 *Es seien F und R disjunkte Mengen und $\langle F, \Omega\rangle$ und $\langle R, \Xi\rangle$ Signaturen. Dann ist $A_{\Omega,\Xi} := \{\mathtt{x}, \mathtt{0}, \mathtt{1}, \mathtt{(}, \mathtt{)}, \mathtt{,}, \doteq, \forall, \exists, \wedge, \neg, \vee\}$. Ein **Term** über der Signatur Ω ist wie folgt definiert.*

1. *Ist $\vec{y}$ eine Binärfolge, so ist $\mathtt{x}\vec{y}$ ein Term.*
2. *Ist $f \in F$ und sind t_i, $i < \Omega(f)$, Terme, so auch $f\mathtt{(}t_0\mathtt{,}t_1\mathtt{,}\cdots\mathtt{,}t_{\Omega(f)-1}\mathtt{)}$.*

*Die Menge der Terme wird mit Tm_Ω bezeichnet. Eine **Variable** ist ein Term der Form $\mathtt{x}\vec{y}$. Var bezeichnet die Menge der Variablen. Die Menge der **Formeln** wird wie folgt definiert.*

1. *Ist $\sigma \in R$ und sind t_i, $i < \Xi(\sigma)$, Terme, so auch $\sigma\mathtt{(}t_0\mathtt{,}t_1\mathtt{,}\cdots\mathtt{,}t_{\Xi(\sigma)-1}\mathtt{)}$.*
2. *Sind t_0 und t_1 Terme, so ist $\mathtt{(}t_0 \doteq t_1\mathtt{)}$ eine Formel.*
3. *Ist φ eine Formel, so auch $\mathtt{(}\neg\varphi\mathtt{)}$.*
4. *Sind φ_0 und φ_1 Formeln, so auch $\mathtt{(}\varphi_0 \wedge \varphi_1\mathtt{)}$.*
5. *Sind φ_0 und φ_1 Formeln, so auch $\mathtt{(}\varphi_0 \vee \varphi_1\mathtt{)}$.*
6. *Ist φ eine Formel und $\mathtt{x}\vec{y}$ eine Variable, so ist auch $\mathtt{(}\exists\mathtt{x}\vec{y}\mathtt{)}\varphi$ eine Formel.*
7. *Ist φ eine Formel und $\mathtt{x}\vec{y}$ eine Variable, so ist auch $\mathtt{(}\forall\mathtt{x}\vec{y}\mathtt{)}\varphi$ eine Formel.*

*Die Menge der Formeln wird mit $\mathrm{Fml}_{\Omega,\Xi}$ bezeichnet. In diesem Zusammenhang heißt Ω die **funktionale Signatur** und Ξ die **relationale Signatur** von $\mathrm{Fml}_{\Omega,\Xi}$.*

Man beachte also, dass die Subskripte Ω und Ξ die Abhängigkeit von der Signatur andeuten. Wir weisen ausdrücklich darauf hin, dass ein und dasselbe Symbol in der einen Sprache ein Funktionssymbol sein kann, in der anderen ein Relationssymbol, und dass die Stelligkeit sich von Sprache zu Sprache unterscheiden kann. Ist $\Omega(\mathtt{g}) = 1$ und $\Xi(\mathtt{r}) = 2$, so ist Folgendes eine Formel

$$\mathtt{(\exists x01)(r(g(x11),g(g(x01)))} \wedge \mathtt{r(g(x01),g(g(x11))))}$$

Definition 106 *Es seien Ω und Ξ Signaturen. Dann ist eine $\langle\Omega, \Xi\rangle$–**Struktur** ein Paar $\mathfrak{M} = \langle M, \mathfrak{I}\rangle$ derart, dass M eine Menge ist und $\mathfrak{I}$ eine Funktion, welche definiert ist auf $F \cup R$, und welche Folgendes erfüllt.*

1. *Ist $f \in F$, so ist $\mathfrak{I}(f) : M^{\Omega(f)} \to M$.*

2. *Ist $\sigma \in R$, so ist $\mathfrak{I}(\sigma) \subseteq M^{\Xi(\sigma)}$.*

*M heißt hier auch der **Bereich** der Struktur $\mathfrak{M}$ und $\mathfrak{I}$ die **Interpretationsfunktion**. $\mathfrak{I}(f)$ bzw. $\mathfrak{I}(\sigma)$ heißt die **Interpretation** des Funktionssymbols f bzw. des Relationssymbols σ.*

Mit anderen Worten, ist f ein Funktionssymbol, so ist die Interpretation des Symbols f in M eine $\Omega(f)$–stellige Funktion auf M. Ist σ hingegen ein Relationssymbol, so ist die Interpretation von σ eine $\Xi(\sigma)$–stellige Relation auf M. Man mache sich klar, dass diese Definition nur deswegen eindeutig ist, weil F und R disjunkt sind. Nehmen wir zum Beispiel folgende Signaturen: $\Omega : + \mapsto 2, \cdot \mapsto 2, \mathtt{s} \mapsto 1$, $\Xi : < \mapsto 2$. Dann ist zum Beispiel eine Struktur über dieser Signatur das Tripel $\langle\omega, \mathfrak{I}\rangle$, wo $\mathfrak{I}(\mathtt{s})$ die Nachfolgerfunktion, $\mathfrak{I}(+)$ die Addition, $\mathfrak{I}(\cdot)$ die Multiplikation, und $\mathfrak{I}(<)$ die Kleinerrelation auf Ω ist. Natürlich ist es nicht zwingend, dass $\mathfrak{I}(+)$ durch die Addition auf ω interpretiert wird; jede andere zweistellige Funktion täte es genauso gut.

Wir notieren noch ein paar wichtige Sonderfälle. Ist $\Omega(f) = 0$, so ist f ein nullstelliges Symbol, mithin $\mathfrak{I}(f) : M^0 \to M$. Wir haben schon gesagt, dass $M^0 = \{\varnothing\}$. Folglich ist $\mathfrak{I}(f) : \varnothing \mapsto m$ für ein gewisses $m \in M$. $\mathfrak{I}(f)$ ist dann also eine Funktion, welche der leeren Menge ein Symbol aus M zuordnet. Wir nennen diese Funktion auch eine **Konstante**. Die Syntax von f ist wie folgt. Es ist $f\mathtt{()}$ ein Term. Man schreibt auch gerne unter Weglassen der Klammern f anstelle von $f\mathtt{()}$.

Ein weiterer Sonderfall ist $\Xi(\sigma) = 0$. Dann ist $\mathfrak{I}(\sigma) \subseteq M^0$. $\mathfrak{I}(\sigma)$ kann also nur zwei Werte annehmen: $0 = \varnothing$ und $1 = \{\varnothing\}$. Auch hier schreibt man σ anstelle des aufwändigen $\sigma\mathtt{()}$.

Im Folgenden benutzen wir folgende Konvention. Es bezeichnen x, y und z Variable.

Definition 107 *Ein Vorkommen einer Variable $x = \mathtt{x}\vec{\alpha}$ ist ein Vorkommen $\langle \vec{x}, \vec{y} \rangle$, von $\mathtt{x}\vec{\alpha}$, wo (a) $\vec{y}$ nicht mit $\mathtt{0}$ oder $\mathtt{1}$ beginnt und (b) $\vec{x}$ nicht mit $\exists$ oder $\forall$ aufhört.*

Definition 108 *Es sei $\mathfrak{M} = \langle M, \mathfrak{I} \rangle$ eine $\langle \Omega, \Xi \rangle$–Struktur. Eine **Belegung in** $\mathfrak{M}$ ist eine Funktion $\beta : \mathrm{Var} \to M$. Ein Paar $\langle \mathfrak{M}, \beta \rangle$ aus einer Struktur und einer Belegung darin heißt ein **Modell**. Sind β und γ Belegungen und x eine Variable, so schreiben wir $\beta \sim_x \gamma$, falls $\beta(y) = \gamma(y)$ für alle $y \in \mathrm{Var}$ verschieden von x. γ heißt auch eine x–**Variante von** β.*

Definition 109 *Wir definieren den **Wert** $[t]^{\mathfrak{M},\beta}$ eines Terms in dem Modell $\langle \mathfrak{M}, \beta \rangle$ wie folgt.*

1. $[\mathtt{x}\vec{y}]^{\mathfrak{M},\beta} := \beta(\mathtt{x}\vec{y})$.
2. $[f\,(t_0\,, t_1\,, \cdots, t_{\Omega(f)-1})]^{\mathfrak{M},\beta} := \mathfrak{I}(f)([t_0]^{\mathfrak{M},\beta}, [t_1]^{\mathfrak{M},\beta}, \cdots, [t_{\Omega(f)-1}]^{\mathfrak{M},\beta})$.

Ist φ eine Formel, so definieren wir $\langle \mathfrak{M}, \beta \rangle \vDash \varphi$ induktiv wie folgt.

$$
\begin{array}{lll}
\langle \mathfrak{M}, \beta \rangle \vDash \sigma\,(t_0\,, t_1\,, \cdots, t_{\Xi(\sigma)-1}) & :\Leftrightarrow & \langle [t_0]^{\mathfrak{M},\beta}, [t_1]^{\mathfrak{M},\beta}, \cdots, [t_{\Xi(\sigma)-1}]^{\mathfrak{M},\beta} \rangle \in \mathfrak{I}(\sigma) \\
\langle \mathfrak{M}, \beta \rangle \vDash (t_0 \doteq t_1) & :\Leftrightarrow & [t_0]^{\mathfrak{M},\beta} = [t_1]^{\mathfrak{M},\beta} \\
\langle \mathfrak{M}, \beta \rangle \vDash (\neg\varphi) & :\Leftrightarrow & \langle \mathfrak{M}, \beta \rangle \nvDash \varphi \\
\langle \mathfrak{M}, \beta \rangle \vDash (\varphi_0 \wedge \varphi_1) & :\Leftrightarrow & \langle \mathfrak{M}, \beta \rangle \vDash \varphi_0 \text{ und } \langle \mathfrak{M}, \beta \rangle \vDash \varphi_1 \\
\langle \mathfrak{M}, \beta \rangle \vDash (\varphi_0 \vee \varphi_1) & :\Leftrightarrow & \langle \mathfrak{M}, \beta \rangle \vDash \varphi_0 \text{ oder } \langle \mathfrak{M}, \beta \rangle \vDash \varphi_1 \\
\langle \mathfrak{M}, \beta \rangle \vDash (\exists y)\varphi & :\Leftrightarrow & \text{für ein } \gamma \sim_y \beta : \langle \mathfrak{M}, \gamma \rangle \vDash \varphi \\
\langle \mathfrak{M}, \beta \rangle \vDash (\forall y)\varphi & :\Leftrightarrow & \text{für alle } \gamma \sim_y \beta : \langle \mathfrak{M}, \gamma \rangle \vDash \varphi
\end{array}
$$

*Ist $\langle \mathfrak{M}, \beta \rangle \vDash \varphi$, so sagen wir, φ sei in $\langle \mathfrak{M}, \beta \rangle$ **erfüllt**.*

Nun, da wir definiert haben, wann eine Formel in einem Modell erfüllt ist, müssen wir natürlich zeigen, dass diese Definition nicht widersprüchlich ist. Dazu müssten wir wiederum die eindeutige Lesbarkeit der Sprache beweisen. Der Beweis ist jedoch ganz ähnlich wie derjenige, den wir für die Aussagenlogik geführt haben. Ganz wie in der Aussagenlogik bedient man sich auch gewisser Kurzschreibweisen, um zum Beispiel den Gebrauch von Klammern so gering wie möglich zu halten. Ferner verwendet man bei 2–stelligen Relationssymbolen sehr gerne die Infixnotation. Die einzelnen Konventionen sind im normalen Gebrauch wie auch aus den Ausführungen zu der Aussagenlogik hinlänglich bekannt und müssen hier nicht extra aufgeführt werden.

Definition 110 *Es sei φ eine Formel und y eine Variable. Wir sagen, ein Vorkommen von y in φ sei **gebunden in** φ, falls es Teilvorkommen eines Vorkommens einer Formel von der Form $(\exists y)\chi$ oder von der Form $(\forall y)\chi$ ist. Ist dies nicht der Fall, so heißt dieses Vorkommen **frei**.*

$$(\forall \mathtt{x11})(\mathtt{r}(\mathtt{g}(\mathtt{x0})) \wedge (\exists \mathtt{x0})\mathtt{r}(\mathtt{g}(\mathtt{g}(\mathtt{x0}))))$$

Das erste Vorkommen von `x0` ist frei, das zweite gebunden. Die Variable `x11` kommt in dieser Formel nicht vor. (Genauer: die Zeichenkette `x11` kommt vor, aber ihr einziges Vorkommen ist kein Vorkommen der Variable `x11`.) Ist y eine Variable, so heißt $(\exists y)$ bzw. $(\forall y)$ ein **Quantor**. Ist $\langle \vec{x}, \vec{y}\rangle$ ein Vorkommen des Quantors, so heißt die Formel, mit der $\vec{y}$ beginnt, der **Skopus** von diesem Vorkommens des Quantors. In der obigen Formel ist der Skopus des einzigen Vorkommens von $(\forall \mathtt{x11})$ das einzige Vorkommen der Formel $(\mathtt{r}(\mathtt{g}(\mathtt{x0})) \wedge (\exists \mathtt{x0})\mathtt{r}(\mathtt{g}(\mathtt{g}(\mathtt{x0}))))$, der Skopus des (einzigen) Vorkommens von $(\exists \mathtt{x0})$ ist das einzige Vorkommen von $\mathtt{r}(\mathtt{g}(\mathtt{g}(\mathtt{x0})))$.

Definition 111 *Es bezeichnet $\mathit{frei}(\varphi)$ die Menge der in φ frei vorkommenden Variablen, $\mathit{geb}(\varphi)$ die Menge der in φ gebunden vorkommenden Variablen. φ heißt **Satz**, falls $\mathit{frei}(\varphi) = \varnothing$.*

Man bemerke, dass $\mathit{frei}(\varphi)$ und $\mathit{geb}(\varphi)$ Mengen von Variablen sind, nicht Mengen von Vorkommen derselben. Da eine Variable in einer Formel zugleich frei wie gebunden vorkommen kann, gilt nicht notwendigerweise $\mathit{frei}(\varphi) \cap \mathit{geb}(\varphi) = \varnothing$.

Wir schreiben $\mathfrak{M} \vDash \varphi$, falls für alle Belegungen β in $\mathfrak{M}$ gilt $\langle \mathfrak{M}, \beta\rangle \vDash \varphi$.

Proposition 112 *Es sei $\mathfrak{M}$ eine Struktur und β, β' Belegungen mit $\beta \restriction \mathit{frei}(\varphi) = \beta' \restriction \mathit{frei}(\varphi)$. Dann gilt $\langle \mathfrak{M}, \beta\rangle \vDash \varphi$ genau dann, wenn $\langle \mathfrak{M}, \beta'\rangle \vDash \varphi$. Ist insbesondere φ ein Satz, so gilt $\langle \mathfrak{M}, \beta\rangle \vDash \varphi$ für keine Belegung oder für alle Belegungen. Es folgt, dass entweder $\mathfrak{M} \vDash \varphi$ oder $\mathfrak{M} \vDash \neg\varphi$.*

Insbesondere ist also im Falle, dass φ ein Satz ist, $\langle \mathfrak{M}, \beta\rangle \vDash \varphi$ unabhängig von β. Ist φ kein Satz, so bedeutet $\mathfrak{M} \vDash \varphi$ lediglich, dass für alle Belegungen β gilt $\langle \mathfrak{M}, \beta\rangle \models \varphi$. Eine Formel, für die dies gilt, ohne dass sie ein Satz ist, ist $(\mathtt{x0} \doteq \mathtt{x0})$.

Die letzte Behauptung der Proposition 112 verdient Beachtung. Falls φ ein Satz ist, so auch $\neg\varphi$. Ist nun β ein Belegung, so ist entweder $\langle \mathfrak{M}, \beta\rangle \vDash \varphi$, sodass $\mathfrak{M} \vDash \varphi$ gilt, oder aber es ist $\mathfrak{M} \nvDash \varphi$, und dann gilt $\langle \mathfrak{M}, \beta\rangle \vDash \neg\varphi$ und mithin $\mathfrak{M} \vDash \neg\varphi$. Für allgemeine Formeln, welche keine Sätze sind, kann dies allerdings falsch sein!

13. Teil: Prädikatenlogik II: Substitution

Zunächst einmal müssen wir uns ausführlich mit dem Thema Substitution befassen. Dazu gehen wir noch einmal zurück zur Aussagenlogik. Wir haben gesehen, dass aus $\Gamma \vDash \varphi$ für eine beliebige Substitution σ folgt $\Gamma^\sigma \vDash \varphi^\sigma$. Ist $\Gamma = \varnothing$, so erhalten wir sofort, dass jede Substitutionsinstanz einer Tautologie eine Tautologie ist. Es gilt sogar noch Folgendes.

Satz 113 *Die Menge der aussagenlogischen Tautologien in der Sprache mit den Junktoren $\{\top, \neg, \wedge, \vee, \rightarrow\}$ ist die kleinste Menge, welche die Formeln (a0) – (a11) enthält und abgeschlossen ist unter den Regeln Substitution (sub) und Modus Ponens (mp).*

$$\text{(sub)} \quad \frac{\varphi}{\varphi^\sigma} \qquad \text{(mp)} \quad \frac{\varphi \quad \varphi \rightarrow \chi}{\chi}$$

Beweis. φ ist genau dann eine Tautologie, wenn $\varnothing \vDash \varphi$. Zunächst einmal sind die Regeln sicher korrekt. Denn ist φ eine Tautologie, so auch φ^σ; ebenso ist mit φ und $\varphi \rightarrow \chi$ auch χ eine Tautologie. Zweitens müssen wir zeigen, dass die Regeln auch ausreichen. Wir haben aber bereits gesehen, dass (mp) alleine genügt, um aus sämtlichen Substitutionsinstanzen der Formeln (a0) – (a11) alle Konsequenzen aus $\varnothing$ herzuleiten. Q. E. D.

Im Übrigen gilt das Deduktionstheorem, sodass es auch genügt, sämtliche Tautologien herzuleiten. Man beachte also, dass für die Herleitung einer Formel aus einer Menge Γ von Prämissen die Substitution nicht zulässig ist; man darf Substitution lediglich bei Tautologien anwenden. Es genügt dabei aber, Substitution auf die Formeln (a0) – (a11) zu beschränken.

In der Prädikatenlogik ist die Situation grundsätzlich ähnlich. Es gibt aber zwei weitere Komplikationen. Erstens hat man noch eine neue Regel hinzuzunehmen, zweitens können wir nicht wie im Falle der Aussagenlogik eine Substitution lediglich als Zeichenkettenersetzung auffassen. Eine Substitution ist hier zunächst eine Abbildung $s : \mathit{Var} \rightarrow \mathit{Tm}_\Omega$. Es gilt, eine Abbildung $\overline{s} : \mathit{Fml}_{\Omega,\Xi} \rightarrow \mathit{Fml}_{\Omega,\Xi}$ zu definieren, welche den Folgerungsbegriff erhält. Wie vorher notieren wir φ^s für $\overline{s}(\varphi)$. Das heißt, wir wollen haben, dass aus $\Gamma \vDash \varphi$ wiederum $\Gamma^s \vDash \varphi^s$ folgt. Die Problematik ist nun diese: wir betrachten die Formel

$$(\exists \mathtt{x0})(\mathtt{x0} \doteq \mathtt{x})$$

in der Sprache, welche das einstellige Symbol `s` besitzt. Betrachten wir eine beliebige Belegung β. Dann ist diese Formel sicher erfüllt. Denn wir können ja immer für $\texttt{x0}$ dasjenige Element wählen, welches β der Variable $\texttt{x}$ zuweist. Nun betrachten wir die Ersetzung $s : \texttt{x0} \mapsto \texttt{s(x)}$ (für alle anderen Variablen y sei $s(y) := y$). Falls wir jetzt einfach jedes Vorkommen der Variablen $\texttt{x0}$ durch $s(\texttt{x0})$ ersetzen, so erhalten wir folgende Formel:

$$(\exists\texttt{x0})(\texttt{s(x)} \doteq \texttt{x})$$

Diese ist in dem Modell der natürlichen Zahlen, in dem $\texttt{s}$ durch die Nachfolgerfunktion interpretiert wird, stets falsch. Wir haben also eine Tautologie, die durch Zeichenkettenersetzung in eine kontingente Formel übergeht. Dies war nicht beabsichtigt. Die Ursache des Problems ist, dass das entsprechende Vorkommen der Variablen durch einen Quantor gebunden ist. Die in dem Quantor versteckte Variable wird nicht mitsubstituiert. Und zwar nicht einfach deshalb, weil sie dort technisch nicht 'vorkommt', sondern weil wir dorthin keine Terme schreiben dürfen. Mit Ausführung der Ersetzung verliert der Quantor aber seine gebundene Variable. Die Lehre daraus ist, dass wir gebundene Variablen nicht in derselben Weise ersetzen dürfen. Ein Ausweg aus dieser Situation ist die sogenannte *gebundene Umbenennung.*

Definition 114 *Es sei χ eine Formel, y und x Variable. Wir sagen, y sei **frei für** x **in** χ, falls (a) y nicht frei in χ vorkommt und (b) x nicht in einer Formel der Form $(Qy)\varphi$ vorkommt mit $Q = \forall$ or $Q = \exists$.*

Definition 115 *Es sei φ eine Formel und $(Qx)\chi$ ein Vorkommen einer Teilformel, $Q = \forall$ oder $Q = \exists$. Wir sagen, φ' sei aus φ durch **einmalige gebundene Umbenennung** entstanden, wenn φ' aus φ durch die Ersetzung dieses Vorkommens von $(Qx)\chi$ durch $(Qy)\chi'$ hervorgeht, wobei χ' seinerseits aus χ durch Ersetzung der in χ freien (!) Vorkommen von x durch y entsteht und y frei für x in χ ist. φ' ist eine **gebundene Variante von** φ, wenn es eine Folge χ_i, $i < n+1$, gibt derart, dass $\chi_0 = \varphi$, $\chi_n = \varphi'$ und für alle $i < n$ χ_{i+1} aus χ_i durch einmalige gebundene Umbenennung entstanden ist.*

Satz 116 *Es sei φ' eine gebundene Variante von φ und $\langle\mathfrak{M}, \beta\rangle$ ein Modell. Dann gilt $\langle\mathfrak{M}, \beta\rangle \vDash \varphi$ genau dann, wenn $\langle\mathfrak{M}, \beta\rangle \vDash \varphi'$.*

Beweis. Zunächst einmal genügt es, den Fall zu betrachten, dass φ' durch einmalige gebundene Umbenennung aus φ hervorgeht. Weiter können wir uns auf den Fall beschränken, wo $\varphi = (\forall x)\chi$ bzw. $\varphi = (\exists x)\chi$, und $\varphi' = (\forall y)\chi'$ bzw. $\varphi' = (\exists y)\chi'$ ist, wo χ' aus χ durch Ersetzung der freien Vorkommen von x durch y entsteht. Sei $\langle \mathfrak{M}, \beta \rangle \vDash (\exists x)\chi$. Dann gibt es ein $\gamma \sim_x \beta$ mit $\langle \mathfrak{M}, \gamma \rangle \vDash \chi$. Definiere γ' durch $\gamma'(y) := \gamma(x)$, $\gamma'(x) := \beta(x)$, $\gamma'(z) := \gamma(z)$ für alle $z \neq x, y$. Nun gilt, wie man durch Induktion leicht bestätigt, $\langle \mathfrak{M}, \gamma' \rangle \vDash \chi'$. Als Letztes bestätigt man, dass $\beta \sim_y \gamma'$. Also gilt $\langle \mathfrak{M}, \beta \rangle \vDash (\exists y)\chi'$. Die Umkehrung folgt genauso: denn φ entsteht aus φ' durch Ersetzung von y durch x. (Das zu ersetzende x ist nicht mehr frei in χ'.) Analog der andere Fall, nämlich $\varphi = (\forall x)\chi$. Q. E. D.

Wir schreiben wieder $\varphi \equiv \chi$, falls für alle Strukturen $\mathfrak{M}$ und alle Belegungen β gilt: $\langle \mathfrak{M}, \beta \rangle \vDash \varphi$ genau dann, wenn $\langle \mathfrak{M}, \beta \rangle \vDash \chi$. Alternativ gilt $\varphi \equiv \chi$ genau dann, wenn $\langle \mathfrak{M}, \beta \rangle \vDash \varphi \leftrightarrow \chi$ für alle Modelle $\langle \mathfrak{M}, \beta \rangle$. Wir beobachten folgende nützliche Äquivalenzen:

$$\begin{array}{rcl} (\forall x)\chi & \equiv & \neg(\exists x)\neg\chi \\ (\exists x)\chi & \equiv & \neg(\forall x)\neg\chi \end{array}$$

Dies erlaubt uns, im vorigen Satz auf einen Beweis des zweiten Falls zu verzichten.

Betrachten wir nun eine andere Abbildung, $t : \mathtt{x} \mapsto \mathtt{s(x0)}$ (auf allen Variablen y ist $t(y) := y$). Führen wir genauso wie eben die Zeichenkettenersetzung durch, so erhalten wir nunmehr die Formel

$$(\exists \mathtt{x0})(\mathtt{x0} \doteq \mathtt{s(x0)})$$

Auch diese Formel ist in dem Modell der natürlichen Zahlen mit der Nachfolgerfunktion nicht erfüllbar. Wir haben diesmal aber kein gebundenes Vorkommen einer Variable ersetzt sondern ein freies. Das Problem hier ist, dass der ersetzende Term die Variable `x0` enthält, welche an der Stelle, wo sie hineingesetzt wird, gebunden vorkommt. Dies ist also auch nicht statthaft.

Wir definieren nun eine Substitutionsabbildung, die beides vermeidet. Von dieser lässt sich dann zeigen, dass sie die Konsequenzrelation erhält. Zunächst aber ein Stück Notation. Sei u ein Term und φ eine Formel. Wir bezeichnen mit $[t/x]u$ bzw. $[t/x]\varphi$ den Effekt der noch zu definierenden Substitution auf u bzw. φ, welche x auf t abbildet, jede andere Variable auf sich selbst. (Oft schreibt man $u[t/x]$ bwz. $\varphi[t/x]$ anstelle von $[t/x]u$ und $\varphi[t/x]$, aber für uns ist diese Schreibweise nicht zweckmäßig.) Wir nennen solche Substitution auch **Elementarsubstitutionen**. Zunächst einmal definieren wir den Effekt nur für Terme, da dies unproblematisch ist.

$$[t/x]y := \begin{cases} t & \text{falls } y = x, y \text{ Variable} \\ y & \text{sonst.} \end{cases}$$

$$[t/x]f(s_0, s_1, \ldots, s_{\Omega(f)-1}) := f([t/x]s_0, [t/x]s_1, \ldots, [t/x]s_{\Omega(f)-1})$$

Auf Termen ist also die Substitutionsabbildung die reine Zeichenkettenersetzung. Bevor wir weitermachen, erwähnen wir, dass man Ersetzungen natürlich iterieren kann. Es ist $[s/y][t/x]u$ der Effekt der Ersetzung $x \mapsto t$ verkettet mit der Ersetzung $y \mapsto s$. Ferner betrachtet man oft auch die gleichzeitige Ersetzung von Termen für verschiedene Variablen. Es bezeichnet daher

$$[t_0/x_0, t_1/x_1, \ldots, t_{n-1}/x_{n-1}]s$$

das Ergebnis der *gleichzeitigen* Ersetzung in s von x_i durch t_i für alle $i < n$. Dies ist nur für endliches n definiert. Man beachte, dass die gleichzeitige Substitution ein anderes Ergebnis liefert als die serielle:

$$[x_2/x_1, x_1/x_0]x_0 = x_1, \qquad [x_2/x_1][x_1/x_0]x_0 = x_2 \ .$$

Trotzdem gilt:

Satz 117 *Jede simultane Ersetzung ist die Verkettung von einfachen Ersetzungen.*

(Dies ist nicht ganz einfach zu zeigen. Wir verzichten deswegen hier auf den Nachweis.) Nun haben wir als Letztes noch die allgemeinen Substitutionen $s : Tm_\Omega \to Tm_\Omega$. Diese sind nicht auf diese Weise darstellbar. Allerdings existiert für jeden gegebenen Term u und jede Substitution s eine simultane Ersetzung

$$u^s = [x_0^s/x_0, x_1^s/x_1, \ldots, x_{n-1}^s/x_{n-1}]u$$

wobei x_i, $i < n$, gerade die in u auftretenden Variablen sind. Also können wir im Folgenden auf die Diskussion von simultanen Ersetzungen verzichten.

Nun definieren wir $[t/x]\varphi$. Die folgenden Fälle sind unproblematisch.

$$\begin{array}{lcl}
[t/x]\sigma(s_0, s_1, \ldots, s_{\Xi(\sigma)-1}) & := & f([t/x]s_0, [t/x]s_1, \ldots, [t/x]s_{\Xi(\sigma)-1}) \ , \\
[t/x](s_0 \doteq s_1) & := & ([t/x]s_0 \doteq [t/x]s_1) \ , \\
[t/x](\neg\vec{x}) & := & (\neg[t/x]\vec{x}) \ , \\
[t/x](\vec{x} \wedge \vec{y}) & := & ([t/x]\vec{x} \wedge [t/x]\vec{y}) \ , \\
[t/x](\vec{x} \vee \vec{y}) & := & ([t/x]\vec{x} \vee [t/x]\vec{y}) \ .
\end{array}$$

Bei den Quantoren ist aber äußerste Vorsicht geboten.

$$[t/x](\exists y)\vec{x} \quad := \quad \begin{cases} (\exists y)\,\vec{x}, & \text{falls } y = x, \\ (\exists y)\,[t/x]\vec{x}, & \text{falls } y \neq x, y \notin var(t), \\ (\exists z)\,[t/x][z/y]\vec{x}, & \text{falls } y \neq x, y \in var(t), \\ & \text{wobei } z \notin var(t), z \text{ frei für } y \text{ in } \vec{x}, \end{cases}$$

$$[t/x](\forall y)\vec{x} \quad := \quad \begin{cases} (\forall y)\,\vec{x}, & \text{falls } y = x \\ (\forall y)\,[t/x]\vec{x}, & \text{falls } y \neq x, y \notin var(t), \\ (\forall z)\,[t/x][z/y]\vec{x}, & \text{falls } y \neq x, y \in var(t), \\ & \text{wobei } z \notin var(t), z \text{ frei für } y \text{ in } \vec{x}. \end{cases}$$

Wir müssen uns offensichtlich vor allem mit den Quantoren befassen. Zunächst einmal sei gesagt, dass das Ergebnis der Substitution gar nicht eindeutig bestimmt ist. Dies kann man beheben, indem man verlangt, dass die Variable z in der zweiten Klausel die kleinste sei, die den gegebenen Bedingungen genügt. Man beachte ferner, dass man zur Berechnung der Substitution nunmehr zwei einzelne Substitutionen hintereinander ausführen muss. Man muss zunächst $[z/y]\vec{x}$ berechnen und damit die Formel auf die anschließende Termersetzung $x \mapsto t$ vorbereiten. Dies hat lediglich den Zweck zu vermeiden, dass wir ein gebundenes Vorkommen einer Variable erzeugen.

Lemma 118 *Es sei $\langle \mathfrak{M}, \beta \rangle$ ein Modell und u ein Term, $\gamma \sim_x \beta$ derart, dass $\gamma(x) = [t]^{\mathfrak{M},\beta}$. Dann gilt $([t/x]u)^{\mathfrak{M},\beta} = u^{\mathfrak{M},\gamma}$.*

Dies ist ein Spezialfall des

Lemma 119 *Es sei s eine Substitution. $\langle \mathfrak{M}, \beta \rangle$ ein Modell und γ definiert durch $\gamma(x) := [\overline{s}(x)]^\beta$. Dann ist für jeden Term t: $[t]^\gamma = [\overline{s}(t)]^\beta$.*

Satz 120 *Es sei $\langle \mathfrak{M}, \beta \rangle$ ein Modell und $\gamma \sim_x \beta$ mit $\gamma(x) = [t]^{\mathfrak{M},\beta}$. Dann gilt $\langle \mathfrak{M}, \gamma \rangle \vDash \varphi$ genau dann, wenn $\langle \mathfrak{M}, \beta \rangle \vDash [t/x]\varphi$.*

Beweis. Auch dieser Beweis wird induktiv geführt. Die Terme sind schon abgehandelt. Sei φ jetzt von der Form $\sigma(s_0, s_1, \ldots, s_{\Xi(\sigma)-1})$. Aus $\langle \mathfrak{M}, \beta \rangle \vDash [t/x]\sigma(s_0, s_1, \ldots, s_{\Xi(\sigma)-1})$ folgt

$$\langle \mathfrak{M}, \beta \rangle \vDash \sigma([t/x]s_0, [t/x]s_1, \ldots, [t/x]s_{\Xi(\sigma)-1})$$

nach Definition der Substitution. Dies ist genau dann der Fall, wenn

$$\langle ([t/x]s_0)^{\mathfrak{M},\beta}, \ldots, ([t/x]s_{\Xi(\sigma)-1})^{\mathfrak{M},\beta} \rangle \in \mathfrak{I}(\sigma) \; .$$

Aber da $([t/x]s_i)^{\mathfrak{M},\beta} = s_i^{\mathfrak{M},\gamma}$, weil $\gamma = \beta \circ \overline{s}$, so haben wir

$$\langle s_0^{\mathfrak{M},\gamma}, \ldots, s_{n-1}^{\mathfrak{M},\gamma} \rangle \in \mathfrak{I}(\sigma) \ .$$

Daraus erhalten wir

$$\langle \mathfrak{M}, \gamma \rangle \vDash \sigma(s_0, \ldots, s_{n-1}) \ .$$

Diese Schlusskette ist umkehrbar. Der Fall $\varphi = (s_0 \doteq s_1)$ ist ähnlich. Nun zu $\varphi = \neg\chi$. Es gilt

$$\begin{array}{ll} & \langle \mathfrak{M}, \gamma \rangle \vDash \neg\chi \\ \text{gdw.} & \langle \mathfrak{M}, \gamma \rangle \nvDash \chi \\ \text{gdw.} & \langle \mathfrak{M}, \beta \rangle \nvDash [t/x]\chi \\ \text{gdw.} & \langle \mathfrak{M}, \beta \rangle \vDash \neg[t/x]\chi \quad (= [t/x]\varphi) \end{array}$$

Ähnlich geradeaus sind die Fälle, wo $\varphi = \chi_1 \wedge \chi_2$ und $\varphi = \chi_1 \vee \chi_2$. Jetzt sei $\varphi = (\exists y)\chi$. Fall 1. $y = x$. Dann ist $[t/x]\varphi = \varphi$. Es ist dann aber $\langle \mathfrak{M}, \gamma \rangle \vDash \varphi$ genau dann, wenn $\langle \mathfrak{M}, \beta \rangle \vDash \varphi$, weil x gar nicht frei in φ ist. Fall 2. $y \neq x$ aber $y \notin var(t)$. Sei $\langle \mathfrak{M}, \gamma \rangle \vDash \varphi$. Dann existiert ein $\gamma' \sim_y \gamma$ mit $\langle \mathfrak{M}, \gamma' \rangle \vDash \chi$. Jetzt definiere $\beta' \sim_x \gamma'$ durch $\beta'(x) := [t]^{\mathfrak{M},\gamma'} = [t]^{\mathfrak{M},\gamma}$, da ja $y \notin var(t)$. Dann gilt nach Induktionsvoraussetzung $\langle \mathfrak{M}, \beta' \rangle \vDash [t/x]\chi$. Nun ist $\langle \mathfrak{M}, \beta \rangle \vDash (\exists y)[t/x]\chi$. Dieser Schluss ist umkehrbar. Fall 3. $y \neq x$ und $y \in var(t)$. Dieser Fall lässt sich wie folgt auf den vorigen zurückspielen. Es ist $(\exists z)[z/y]\vec{x}$ eine gebundene Variante von $(\exists y)\vec{x}$, wie man leicht sieht. Deswegen gilt

$$\langle \mathfrak{M}, \gamma \rangle \vDash (\exists y)\vec{x} \text{ gdw. } \langle \mathfrak{M}, \gamma \rangle \vDash (\exists z)[z/y]\vec{x} \ .$$

Jetzt sind wir im Fall 2, denn die gebundene Variable tritt in t nicht auf. Es gilt jetzt

$$\langle \mathfrak{M}, \gamma \rangle \vDash (\exists z)[z/y]\vec{x} \text{ gdw. } \langle \mathfrak{M}, \beta \rangle \vDash [t/x](\exists z)[z/y]\vec{x} \ .$$

Nun ist aber $[t/x](\exists z)[z/y]\vec{x} = [t/x](\exists y)\vec{x}$, woraus die Behauptung letztlich folgt. Q. E. D.

Es folgt jetzt aus diesem Satz der ersehnte Schluss:

Korollar 121 *Ist φ eine Tautologie, so auch $[t/x]\varphi$.*

Beweis. Es sei $\langle \mathfrak{M}, \beta \rangle$ ein Modell. Dann gilt $\langle \mathfrak{M}, \beta \rangle \vDash [t/x]\varphi$ genau dann, wenn $\langle \mathfrak{M}, \gamma \rangle \vDash \varphi$, wo $\gamma \sim_x \beta$ definiert ist durch $\gamma(x) := [t]^{\mathfrak{M},\beta}$. Da φ eine Tautologie ist, gilt aber $\langle \mathfrak{M}, \beta \rangle \vDash \varphi$. Q. E. D.

14. Teil: Prädikatenlogik III: Ein Hilbert–Kalkül, Horn–Klauseln und Prolog

Wir stellen nun einen Hilbert–Kalkül für die Prädikatenlogik vor. Dieser ist in einem gewissen Sinne eine Erweiterung des Kalküls für die Aussagenlogik. Die Formeln (a0) – (a11) bleiben wie bisher. Nun treten folgende Formeln hinzu.

$$
\begin{array}{lll}
\text{(a12)} & (\forall x)(\varphi \to \chi) \to (\forall x)\varphi \to (\forall x)\chi & \\
\text{(a13)} & (\forall x)\varphi \to [t/x]\varphi & \\
\text{(a14)} & \varphi \to (\forall x)\varphi & (x \notin \mathit{frei}(\varphi)) \\
\text{(a15)} & (\forall x)\varphi \to \neg(\exists x)\neg\varphi & \\
\text{(a16)} & \neg(\exists x)\neg\varphi \to (\forall x)\varphi &
\end{array}
$$

Wir bekommen aus (a15) und (a16) sofort, dass $(\forall x)\varphi \leftrightarrow \neg(\exists x)\neg\varphi$ sowie $(\exists x)\varphi \leftrightarrow \neg(\forall x)\neg\varphi$ beweisbar sind, weswegen man auf einen der beiden Quantoren in der Sprache ganz verzichten kann.

Theorem 122 *Genau dann ist φ eine Tautologie in der Prädikatenlogik mit den Junktoren $\{\top, \neg, \wedge, \vee, \to, \forall, \exists\}$, wenn es aus den Formeln (a0) – (a16) mittels der Regeln* (sub), (mp) *und* (gen) *herleitbar ist.*

$$
\text{(sub)}\ \frac{\varphi}{\varphi^s} \qquad \text{(mp)}\ \frac{\varphi, \quad \varphi \to \chi}{\chi} \qquad \text{(gen)}\ \frac{\varphi}{(\forall x)\varphi}
$$

Zunächst einmal sind diese Regeln sicher korrekt. Die Schwierigkeit besteht darin zu zeigen, dass der durch sie definiert Kalkül vollständig ist.

Wir skizzieren hier den Beweis. Angenommen, es sei φ nicht herleitbar in diesem Kalkül. Wir müssen zeigen, dass es eine Struktur $\mathfrak{M}$ und eine Belegung β gibt mit $\langle \mathfrak{M}, \beta \rangle \vDash \neg\varphi$. In Gegenwart der Regel (gen) kann man sich allerdings auf den Fall beschränken, dass φ keine freien Variablen enthält. Denn falls $[y/x]\varphi$ für eine beliebige Variable nicht herleitbar ist, so auch nicht $(\forall x)\varphi$. So kann man sich also die Definition von β ersparen.

Dazu benötigt man zwei Schritte.

Definition 123 *Eine Menge von Formeln Γ heißt* ***konsistent****, falls nicht für jede Formel φ gilt $\Gamma \vDash \varphi$. Γ heißt* ***maximal konsistent****, falls Γ konsistent ist, aber jede echte Obermenge inkonsistent.*

Wie vorher ist Γ konsistent, falls $\Gamma \nvDash \bot$.

Lemma 124 *Jede konsistente Menge ist in einer maximal konsistenten Menge von Formeln enthalten.*

Beweis. Es sei Γ konsistent. Es existiert eine Wohlordnung auf unseren Formeln: $Fml_{\Omega,\Xi} = \{\xi_i : i \in \omega\}$. Wir definieren jetzt induktiv folgende Mengen.

$$\begin{array}{lcl} \Gamma_0 & := & \Gamma \\ \Gamma_{i+1} & := & \begin{cases} \Gamma_i \cup \{\xi_i\} & \text{falls } \Gamma_i \nvDash \neg\xi_i \\ \Gamma_i & \text{sonst.} \end{cases} \\ \Gamma_\omega & := & \bigcup_{i<\omega} \Gamma_i \end{array}$$

Dann zeigt man, dass für jedes $i \in \omega$ Γ_i konsistent ist. Auch dies geht durch Induktion. Γ_0 ist nach Annahme über Γ konsistent. Ist Γ_i konsistent, so auch Γ_{i+1}. Denn entweder ist $\Gamma_{i+1} = \Gamma_i$, in welchem Fall die Behauptung trivial ist, oder aber $\Gamma_{i+1} = \Gamma_i \cup \{\xi_i\}$, in welchem Fall $\Gamma_i \nvDash \neg\xi_i$, woraus folgt, dass $\Gamma_i; \xi_i \nvDash \bot$. Ferner ist Γ_ω konsistent. Denn gilt $\Gamma_\omega \vDash \bot$, so existiert ein Beweis Π von $\bot$ aus Γ_ω. Dieser verwendet nur endlich viele Formeln, sodass es ein $n \in \omega$ derart geben muss, dass alle diese Formeln schon in Γ_n sind. Also ist Π ein Beweis von φ aus Γ_n. Das kann nicht gelten. Nun wollen wir noch zeigen, dass Γ_ω maximal konsistent ist. Sei $\Delta \supsetneq \Gamma_\omega$ eine konsistente Menge. Dann existiert ein $i \in \omega$ mit $\xi_i \in \Delta - \Gamma_\omega$. Dann ist $\Gamma_\omega \nvDash \xi_i$, aber $\Gamma_\omega; \xi_i \nvDash \bot$. Insbesondere ist dann auch $\Gamma_i; \xi_i \nvDash \bot$. Daraus folgt wiederum $\xi_i \in \Gamma_{i+1}$, nach Definition. Wir haben einen Widerspruch. Also ist Γ_ω maximal konsistent. Q. E. D.

Definition 125 *Eine Menge Γ von Formeln heißt* ***reich an Zeugen****, falls für jede Formel $(\exists x)\varphi \in \Gamma$ ein konstanter Term t existiert mit $(\exists x)\varphi \to [t/x]\varphi \in \Gamma$.*

Der zweite Schritt ist nun

Lemma 126 *Zu jeder konsistenten Menge Γ existiert eine konsistente Menge $\Delta \supseteq \Gamma$ in einer Spracherweiterung, welche reich an Zeugen ist.*

Man leitet aus diesen beiden Sätzen ab, dass zu jedem Γ eine maximal konsistente Obermenge Δ gibt, welche auch reich an Zeugen ist. Nun definieren wir das ersehnte Modell. Es sei C die Menge der konstanten Terme.

$$[t]_\Delta := \{s : t \doteq s \in \Delta\} .$$

Dann ist

$$M_\Delta := \{[t]_\Delta : t \in C\} .$$

Für eine Funktion f setzen wir

$$\mathfrak{I}_\Delta(f)([t_0]_\Delta, \ldots, [t_{\Omega(f)-1}]_\Delta) := [f(t_0, \ldots, t_{\Omega(f)-1})]_\Delta \, .$$

Für eine Relation setzen wir

$$\langle [t_i]_\Delta : i < \Xi(\sigma) \rangle \in \mathfrak{I}_\Delta(\sigma) \quad \text{gdw.} \quad \sigma(t_0, \ldots, t_{\Xi(\sigma)-1}) \in \Delta \, .$$

Endlich ist $\mathfrak{M}_\Delta := \langle M_\Delta, \mathfrak{I}_\Delta \rangle$ das gesuchte Modell. Nun ist φ ohne freie Variablen, und wir haben für alle variablenfreien Formeln.

$$\mathfrak{M} \vDash \chi \quad \text{gdw.} \quad \chi \in \Delta$$

Dies kann man durch Induktion bestätigen. Diese Induktion ist allerdings nicht ganz so einfach.

Wir beschließen diese Ausführungen über Logik mit ein paar Bemerkungen über die Eigenschaften der Prädikatenlogik. Der Beweis, dass der Hilbert–Kalkül vollständig ist, stammt von Kurt Gödel. Wir haben oben gesehen, dass es eine Möglichkeit gibt, für eine endliche Menge Γ und eine Formel φ zu entscheiden, ob $\Gamma \vDash \varphi$ oder nicht. Dazu genügt es zum Beispiel, alle Belegungen (auf den Variablen, die wirklich auftreten) zu überprüfen. Für die Prädikatenlogik gilt das allerdings nicht.

Theorem 127 *Es gibt keinen Algorithmus zu entscheiden, ob für eine endliche Menge von prädikatenlogischen Formeln Γ und eine prädikatenlogische Formel φ gilt $\Gamma \vDash \varphi$ oder nicht.*

Für einen richtigen Beweis (den wir hier sowieso nicht führen wollen) müssten wir noch exakt spezifizieren, was ein Algorithmus ist. Es sei allerdings erwähnt, dass ein Computerprogramm unter den Begriff eines Algorithmus fällt. Theorem 122 sagt aber, dass man alles, was richtig ist, beweisen kann. Ist also $\Gamma \vDash \varphi$, so kann man auch einen Beweis im Hilbert–Kalkül finden. Nun sieht es so aus, als wäre damit auch entscheidbar, ob φ aus Γ folgt. Das ist nicht so. Denn wenn *kein* Beweis existiert, so kann man dies nicht wissen, denn man weiß ja nicht, wie lange man nach einem Beweis suchen muss. Denn nur wenn man alleine aufgrund von Γ und φ weiß, wie lang ein Beweis höchstens sein kann, dann kann man wissen, ob er existiert oder nicht.

An dieser Stelle brechen wir ab und wenden uns einem Spezialfall zu, nämlich der Logik der sogenannten Horn–Klauseln.

Definition 128 *Eine Formel der Form* $(\forall x_0)(\forall x_1)\ldots(\forall x_{n-1})((\bigwedge_{i<n}\lambda_i)\to\pi)$*, wobei die* λ_i*,* $i<n$*, und die Formel* π *atomare Formeln sind, heißt eine* ***Horn–Formel*** *oder auch* ***Horn–Klausel****.*

Man beachte, dass Horn–Formeln keine Existenzquantoren enthalten und keine Negationen. Horn–Formeln werden in der Sprache `Prolog` verwendet. Dort sind sie die (fast) einzige Möglichkeit, etwas niederzuschreiben. Da zwischen freien Variablen und universell quantifizierten Variablen kein großer Unterschied besteht, wird der Quantor in `Prolog` nicht geschrieben. Außerdem notiert man anstelle von $(\bigwedge_{i<n}\lambda_i)\to\pi$ wie folgt.

$$\pi\texttt{:-}\lambda_0\texttt{,}\lambda_i\texttt{,}\cdots\texttt{,}\lambda_{n-1}\texttt{.}$$

Ein `Prolog`–Programm ist schlicht eine Liste von Horn–Klauseln, notiert in der eben angegebenen Weise. Das folgende kleine Programm definiert eine Eigenschaft `nat` auf Termen, die durch das 0–stellige Symbol und das 1–stellige `suc` definiert sind. Dabei bezeichnet `0` eine natürliche Zahl (die Zahl 0), und wann immer `X` eine natürliche Zahl bezeichnet, so auch `suc(X)`. `Prolog` stellt die Möglichkeit bereit, die Stelligkeit explizit zu erklären wie auch zu wählen, ob man Infix, Präfix oder Suffxischreibweise für das Symbol einführt. Konstanten werden grundsätzlich klein geschrieben, Variablen beginnen mit einem Großbuchstaben. Also schreibt man

```
nat(0) :-.
nat(suc(X)) :- nat(X).
```

(Die erste Zeile wird auch mit `nat(0).` wiedergegeben.) Hat man `Prolog` dieses Programm gegeben, so kann man es anschließend fragen. Zum Beispiel, ob `nat(suc(suc(suc(0))))` gilt. `Prolog` wird dies bejahen. Ist allerdings `a` ebenfalls eine Konstante, so wird `Prolog` die Anfrage `nat(suc(a))` verneinen. Solch eine Klausel kann `Prolog` nicht finden. Man kann auch fragen, für welche Werte von `Y` gilt `nat(suc(suc(Y))`. Als erstes erhält man die Antwort `Y = 0`. Falls man `Prolog` bittet, weiterzusuchen, so bekommt man `Y = suc(0)`, dann `Y = suc(suc(0))`, und so weiter.

Was ist das, was `Prolog` tut? `Prolog` bekommt eine Anfrage, zum Beispiel `nat(suc(suc(suc(0))))`. Um zu sehen, ob sie erfüllt ist, versucht es, die linke Seite einer Klausel mit der Anfrage in Deckung zu bringen. Dies geschieht, indem auftretende Variablen so ersetzt werden, dass die linke Seite dem Ziel entspricht. Dieser Prozess heißt **Unifikation**. Die gefundene Substitution heißt **unifizierende Substitution**. Es kann unter Umständen mehrere unifizierende Substitutionen geben; `Prolog` berechnet immer eine kleinste unifizierende Substitution, aus der alle anderen wiederum durch weitere Substitutionen hervorgehen. `Prolog` hält sich sklavisch an die Reihenfolge. Die Klauseln werden in der gegebenen Reihenfolge abgearbeitet. Es wird immer erst die erste Klausel betrachtet, dann die weiteren. Offensichtlich kann die erste mit dem Ziel nicht in Deckung gebracht werden, da sie mit diesem nicht gleich ist und keine Variablen enthält. In der zweiten Klausel aber verhilft die Substitution von `suc(suc(0))` für `X` zum gewünschten Erfolg. Denn nun bekommen wir

```
nat(suc(suc(suc(0)))) :- nat(suc(suc(0))).
```

Die linke Seite ist erfüllt, falls es die rechte ist. Also setzt sich `Prolog` die rechte Seite zum Ziel und versucht, diese zu erfüllen. Wieder gibt es Erfolg bei der zweiten Klausel. Substitution von `suc(0)` für `X` liefert

```
nat(suc(suc(0))) :- nat(suc(0)).
```

Wiederum macht sich `Prolog` anstelle der linken Seite jetzt die rechte zum Ziel. Mit der zweiten Klausel bekommt es

```
nat(suc(0)) :- nat(0).
```

Die rechte Seite wird zum neuen Ziel, und diesmal wird `Prolog` schon bei der ersten Zeile fündig. Es entsteht diesmal kein neues Ziel, `Prolog` ist fertig.

`Prolog` kann man das Rechnen mit natürlichen Zahlen beibringen. Wir definieren ein dreistelliges Prädikat `add` und geben `Prolog` folgende Klauseln auf den Weg.

```
add(X,0,X).
add(X,suc(Y),suc(Z)) :- add(X,Y,Z).
```

Falls man folgende Aufgabe an `Polog` stellt:

```
? - add(suc(0),suc(suc(0)),suc(suc(0)).
```

so wird `Prolog` dies verneinen (es gibt dann die Antwort `no`). Stellen wir ihm die Aufgabe

```
? - add(suc(0),suc(suc(0)),X).
```

so erhält man die Antwort `X = suc(suc(suc(0)))`. Auf diese Weise kann man jede rekursive Definition über natürlichen Zahlen mittels Horn–Klauseln hinschreiben und `Prolog` zur Berechnung überlassen. Es besitzt im Übrigen keinerlei Funktionen, aber man kann zeigen, dass dies keine Einschränkung ist.

Die Substitutionen bei `Prolog` sind das wesentliche Merkmal für die Effektivität. Sie zu verstehen, ist die Grundlage des Verständnisses von `Prolog` und von Logikprogrammierung im allgemeinen. Man stelle sich vor, die folgende Aufgabe sei an `Prolog` gestellt.

```
? - add(suc(X),suc(suc(X)),Y).
```

Um sie zu lösen, wird `Prolog` eine Substitution der zweiten Klausel berechnen. Hier aber tut sich ein Problem auf: es genügt nicht, nur die linke Seite der Klausel zu substituieren. Denn man muss zum Beispiel `X` $\mapsto$ `suc(X)` sowie `Y` $\mapsto$ `suc(X)` sowie `Z` $\mapsto$ `Z` wählen, und dann bekommt man

```
add(suc(X),suc(suc(X)),suc(Z)) :- add(X,suc(X),Z).
```

Offensichtlich muss `Prolog` jetzt auch an seinem Ziel eine Substitution vornehmen, diesmal aber `Y` $\mapsto$ `suc(Z)`. Geschieht dies, so sind Ziel und linke Seite in der Tat gleich, und `Prolog` verbleibt als Aufgabe, nunmehr

```
add(suc(X),suc(X),Z).
```

zu zeigen. Man beachte also, dass es nicht alleine darauf ankommt, die linken Seiten von Klauseln durch Substitution mit dem Ziel zur Deckung bringen, sondern es wird auch nötig sein, in dem Ziel zu substituieren. Dabei geht `Prolog` allerdings sehr genau vor: die Substitutionen sind so geartet, dass keine mögliche Lösung verlorengeht. In diesem Beispiel ist dies sofort klar. In der Klausel

```
add(suc(X),suc(suc(X)),Y).
```

kann nicht `Y` $=$ `0` gelten, sondern es muss `Y` die Form `suc(Z)` für ein gewisses `Z` haben. Die Substitution war also unschädlich. (Solche 'unschädlichen Substitutionen' heißen auch 'generellste Unifikatoren' (Englisch **most general unifier**).)

`Prolog` wird noch einmal gezwungen sein, diesen Prozess zu durchlaufen, bis es zu `add(suc(X),X,Z).` kommt. Hier endlich findet es eine Lösung:

```
add(suc(0),0,suc(0)).
```

(Erste Klausel: `X` geht nach `0`, `Z` nach `suc(0)`.) Nun kommt die Aufgabe: es muss durch die Substitutionen rückwärts durchgehen, um die ursprünglich gestellte Aufgabe zu lösen. Diese ist

```
add(suc(0),suc(suc(0)),suc(suc(suc(0)))).
```

Die Substitution ist also $\texttt{X} \mapsto \texttt{0}, \texttt{Y} \mapsto \texttt{suc(suc(suc(0))))}$. Falls man `Prolog` zwingt, eine neue Lösung anzubringen, wird es als nächstes `X` durch `suc(X)` ersetzen.

Dies ist ein kleiner Abriss von `Prolog`. In der Sprache der Logik ist ein `Prolog`–Programm eine Menge Γ von Prämissen, und eine Anfrage φ an `Prolog` dient zur Verifikation, ob φ aus Γ folgt. (`Prolog` kann allerdings etwas mehr, nämlich auch Werte für Variable finden derart, dass φ nach Einsetzung für die Variablen aus Γ folgt.) Man kann zeigen, dass die Schlussregeln, welche `Prolog` verwendet, vollständig sind. Allerdings spielt in den abstrakten Kalkülen wie zum Beispiel dem Hilbert–Kalkül die Reihenfolge der Programmklauseln (also der Prämissen) keine Rolle, während sie bei `Prolog` essentiell ist. Man bedenke, dass ein Programm eine Liste ist, keine Menge. Man kann ganz leicht Programme schreiben, die vollständig sind im Sinne der logischen Schlussregeln, auf welchen `Prolog` aber nicht immer terminiert. Die Tatsache, dass `Prolog` nicht deckungsgleich mit der logischen Theorie arbeitet, hat zu vielen Verwirrungen Anlass gegeben, auf die hier aber nicht weiter eingegangen werden soll.

15. Teil: Kombinatorik I. Binomialkoeffizienten

Es sei M eine beliebige Menge. Dann bezeichnet $|M|$ die Anzahl der Elemente von M. Im Folgenden setzen wir stets voraus, dass Mengen endlich sind. Dann ist die Mächtigkeit eine natürliche Zahl. 0 ist eine natürliche Zahl, und es gibt genau eine Menge M mit $0 = |M|$, nämlich die leere Menge. Wir betrachten nun ein paar Konstruktionen aus der Mengenlehre und schauen nach, wie sich die Anzahlen der konstruierten Mengen bestimmen. Zunächst ein Satz, der sich unmittelbar aus den Definitionen ergibt. Zunächst ist $|M| = |N|$ genau dann, wenn eine bijektive Abbildung $f : M \to N$ existiert.

Satz 129 *Es seien M und N endlich Mengen. Dann ist*

$$|M \cup N| = |M| + |N| - |M \cap N| .$$

Ist insbesondere M disjunkt zu N, so ist

$$|M \cup N| = |M| + |N| .$$

Satz 130 *Es seien M_i, $1 \leq i \leq n$, Mengen. Dann ist*

$$|M_1 \times M_2 \times \ldots \times M_n| = \prod_{i=1}^{n} |M_i|$$

Satz 131 *Es seien M und N endliche Mengen, $m := |M|$, $n := |N|$. Es bezeichne N^M die Menge aller Funktionen von M nach N. Dann ist*

$$|N^M| = n^m$$

Beweis. Es sei $M = \{x_1, x_2, \ldots, x_m\}$. Einer Funktion $f : M \to N$ ordnen wir die Folge $\Phi(f) := \langle f(x_1), f(x_2), \ldots, f(x_m)\rangle$ zu. Diese Zuordnung ist bijektiv. $\Phi : N^M \to N \times N \times \ldots \times N$ (m–mal). Also ist nach Satz 130 $|N^M| = \prod_{i=1}^{m} n = n^m$. Q. E. D.

Wir wollen nun als erstes ein nicht etwas schwieriges Zählproblem betrachten, welches in vielen verschiedenen Gewändern auftritt. Wir werden verschiedene Zahlen definieren und für ihre Definition einige äquivalente Formulierungen angeben, bevor wir daran gehen, diese Zahlen durch explizite Formeln zu bestimmen. Es sei M eine endliche Menge und k eine natürliche Zahl. Dann bezeichne $\binom{M}{k}$ die Menge der k–elementigen Teilmengen von M. Wir interessieren uns für die Anzahl der Elemente der Menge $\binom{M}{k}$, also der Anzahl der k–elementigen Teilmengen von M. Falls $|M| = n$, so sei diese Anzahl mit $\binom{n}{k}$ bezeichnet. Es ist klar, dass diese Anzahl nur von $|M|$ abhängt. Bevor wir also zur Bestimmung von $\binom{n}{k}$ übergehen, wollen wir uns Anzahlprobleme ansehen, welche ebenfalls zu den Zahlen $\binom{n}{k}$ führen.

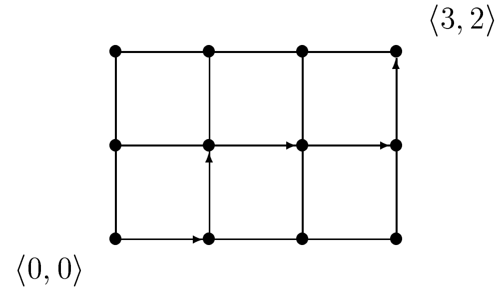
Problem 1. Es sei $m(n, k)$ die Anzahl aller Folgen der Länge n über $\{a, b\}$, welche a genau k mal enthalten. Wir betrachten die Abbildung X, welche der Folge $F = x_1 x_2 \ldots x_n$ die Menge $X(F) := \{i : x_i = a\}$ zuordnet. Es ist $X(F) \subseteq \{1, 2, \ldots, n\}$. X ist bijektiv. Genau dann kommt a in F k–mal vor, wenn $|X(F)| = k$. Also haben wir eine Bijektion zwischen den

Folgen, welche a k–mal enthalten und den Teilmengen von $\{1, 2, \ldots, n\}$ der Mächtigkeit k. Also ist $m(n, k) = \binom{n}{k}$.

Problem 2. Der Term $(x + y)^n$ kann in eine Summe von Termen der Form $a(n, k)x^k y^{n-k}$ zerlegt werden, wobei $0 \leq k \leq n$. Allgemein bekannt ist der Fall $n = 2$: $(x + y)^2 = x^2 + 2xy + y^2$. Also $a(2, 0) = 1$, $a(2, 1) = 2$ und $a(2, 2) = 1$. Wir fragen nach den Zahlen $a(n, k)$. Dazu überlegen wir wie folgt. Offensichtlich kann man $(x + y)^n$ stur ausmultiplizieren; dann entsteht eine Summe von Termen der Form $u_1 u_2 \ldots u_n$, wobei $u_i = x$ oder $u_i = y$. Wir nennen dies einen **Elementarsummanden** von $(x + y)^n$. Jeder Elementarsummand kommt in dieser Summe genau einmal vor. Da beim Multiplizieren die Reihenfolge unerheblich ist, können wir einen Elementarsummanden umschreiben in $x^k y^{n-k}$ für ein $k \leq n$. Dabei ist k gerade die Anzahl aller i für die $u_i = x$. Um also die Zahl $a(n, k)$ zu finden, müssen wir letztlich nur wissen, wie viele Folgen $U = u_1 u_2 \ldots u_n$ es gibt, in denen x genau k mal auftritt. Also ist nach dem vorigen Problem $a(n, k) = \binom{n}{k}$. Man nennt die Zahlen $\binom{n}{k}$ deswegen auch **Binomialkoeffizienten**.

Problem 3. Es sei ein Gitter von Punkten der Ebene gegeben. Es bestehe aus den Punkten (i, j), wobei $0 \leq i \leq m$ und $0 \leq j \leq n$. Ein **Weg der Länge** k in dem Gitter ist eine Folge von Punkten $P_0, P_1, P_2, \ldots, P_k$, wobei P_{i+1} jeweils Nachbar von P_i ist. Der Abstand zwischen zwei Punkten P und Q des Gitters, $d(P, Q)$ist das kleinste k derart, dass ein Weg von P nach Q der Länge k existiert. Ist $P = (p_1, p_2)$ und $Q = (q_1, q_2)$, so ist $d(P, Q) = |p_1 - q_2| + |p_2 - q_2|$. Es interessiert uns die Anzahl der kürzesten Wege zwischen P und Q. Wir nennen sie $w(m, n)$. Das folgende Bild zeigt einen kürzesten Weg von $\langle 0, 0\rangle$ nach $\langle 3, 2\rangle$, nämlich

$$\langle 0, 0\rangle, \langle 1, 0\rangle, \langle 1, 1\rangle, \langle 2, 1\rangle, \langle 3, 1\rangle, \langle 3, 2\rangle$$



(Eine Anwendung: der Stadtplan vieler amerikanischer Städte entspricht einem solchen Gitter. Um von einer beliebigen Kreuzung zu einer anderen zu gelangen, kann man nur entlang des Gitters laufen. Der Abstand wie oben definiert ist gerade die sogenannte **Taximetrik**, sofern das Gitter aus quadratischen Zellen besteht. Denn dieser Abstand bestimmt ziemlich genau die Rechnung, die man fürs Taxi bezahlen muss...) Man kann sich überlegen, dass es reicht, wenn man $P = (0,0)$ wählt, und $Q = (m,n)$. Der Abstand ist gerade $m + n$. Ein kürzester Weg $W = (P, P_1, P_2, \ldots, P_{n+m})$ geht dann immer nach rechts oder oben, niemals nach links oder unten. Das bedeutet, dass, wenn $P_i = (p_i, q_i)$ ist, so ist $P_{i+1} = (p_i + 1, q_i)$ oder $P_{i+1} = (p_i, q_i + 1)$. Einem solchen Weg ordnen wir eine Folge $O(W) = \langle \eta_i : 1 \leq i \leq m + n\rangle$ der Länge $m + n$ zu, wo $\eta_i = o$ falls $P_i = (p_{i-1} + 1, q_{i-1})$ und $\eta_i = r$, falls $P_i = (p_{i-1}, q_{i-1} + 1)$. Die Vorschrift $W \mapsto O(W)$ definiert eine Bijektion zwischen den kürzesten Wegen von $(0,0)$ nach (m,n) und den $n + m$–langen Folgen über $\{o, r\}$, welche genau n mal o enthalten. Also ist $w(m,n) = \binom{m+n}{n}$.

Gehen wir nun zu der Bestimmung von $\binom{n}{k}$ über. Dazu zunächst noch ein neues Zählprinzip. Es sei M eine Menge und $\Pi \subseteq \wp(M) - \{\varnothing\}$ eine Menge von Teilmengen von M. Π heißt **Partition** von M, falls jedes Element von M in genau einer Menge von Π liegt. Ist $\Pi = \{P_1, P_2, \ldots, P_k\}$, so ist natürlich $|M| = \sum_{i=1}^{k} |P_i|$. Der besondere Fall, wo alle P_i die gleiche Mächtigkeit haben, ist besonders hervorhebenswert.

Hilfssatz 132 *Es sei M eine Menge und $\Pi \subseteq \wp(M)$ eine Partition von M, bei der alle Mengen die gleiche Mächtigkeit p haben. Dann ist $|M| = p \cdot |\Pi|$.*

Wir wählen nun als M die Menge aller Folgen der Länge k von Elementen aus N, wobei kein Element wiederholt werden darf. Es habe N die Mächtigkeit n. Dann gibt es genau

$$n^{\underline{k}} := n \cdot (n-1) \cdot \ldots \cdot (n-k+1)$$

Elemente in M. Zu jeder Folge $F = (x_1, x_2, \ldots, x_k)$ sei $\mu(F) := \{x_1, x_2, \ldots, x_k\}$. Da F kein Element wiederholt, ist $|\mu(F)| = k$. Betrachte nun zu jeder k–elementigen Teilmenge X von N die Menge $\Phi(X) := \{F : \mu(F) = X\}$. Dann ist das System

$$\Pi := \{\Phi(X) : X \subseteq N, |X| = k\}$$

eine Partition von M. Ferner hat jedes $\Phi(X)$ die gleiche Anzahl Elemente, nämlich $k^{\underline{k}}$. Für letzteren Ausdruck überlegt man sich, dass er gleich $k!$ ist. Nach dem Hilfssatz 132 ergibt sich nun

Satz 133
$$\binom{n}{k} = \frac{n^{\underline{k}}}{k!} = \frac{n!}{k!(n-k)!}$$
Wir werden damit ein paar Folgerungen für die Binomialkoeffizienten ziehen.

Satz 134
$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}\,.$$

Beweis. Wir nehmen eine Menge X mit n Elementen, und $x \notin X$. Dann hat $X \cup \{x\}$ $n+1$ Elemente. Wir zählen die Anzahl der $k+1$–elementigen Teilmengen von $X \cup \{x\}$. Sei $A \subseteq X \cup \{x\}$. Fall 1. $x \in A$. Dann ist $A - \{x\}$ eine k–elementige Teilmenge von X. Fall 2. $x \notin A$. Dann ist $A - \{x\} = A$ eine $k+1$–elementige Teilmenge von X. Diese Zuordnung ist bijektiv. Ist $A \neq B$, dann ist $A - \{x\} \neq B - \{x\}$. Q. E. D.

Damit kann man die Binomialkoeffizienten relativ leicht berechnen, indem man sie in ein dreieckiges Schema einordnet, wie unten gezeigt. Dieses Schema enthält alle Zahlen $\binom{n}{k}$, wie folgt angeordnet: die $n+1$. Zeile enthält die Zahlen $\binom{n}{k}$ fortlaufend für aufsteigendes k. Die Diagonalen von links unten nach rechts oben enthalten $\binom{n}{k}$, wo k diesmal fest bleibt aber n nach unten gehend ansteigt. Diese Schema ist bekannt als Pascal'sches Dreieck, benannt nach Blaise Pascal (1623 – 1662). Darin ist jede Zahl gerade die Summe der schräg über ihr liegenden Zahlen.

$$\begin{array}{lccccccccccc}
n=0 & & & & & & 1 & & & & & \\
n=1 & & & & & 1 & & 1 & & & & \\
n=2 & & & & 1 & & 2 & & 1 & & & \\
n=3 & & & 1 & & 3 & & 3 & & 1 & & \\
n=4 & & 1 & & 4 & & 6 & & 4 & & 1 & \\
n=5 & 1 & & 5 & & 10 & & 10 & & 5 & & 1
\end{array}$$

Man bemerkt, dass die Summe der Zahlen in der Zeile für n gerade 2^n ist. Zum Beispiel ist $1+4+6+4+1 = 16$. Hingegen ist die alternierende Summe immer 0: $1-4+6-4+1 = 0$.

Satz 135 *Für alle n gilt*

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

2. $\sum_{k=0}^{n}(-1)^n\binom{n}{k} = 0$. *($n > 0$.)*

Beweis. Wir setzen in die Gleichung $(x+y)^n = \sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k}$ 1 für x und y. Dann $2^n = (1+1)^n = \sum_{k=0}^{n}\binom{n}{k}$. Dies ergibt die erste Behauptung. Nun setzen wir -1 für x und 1 für y. Dann erhalten wir für $n > 0$: $0 = ((-)1+1)^n = \sum_{k=0}^{n}(-1)^k\binom{n}{k}$. Q. E. D.

16. Teil: Kombinatorik II. Permutationen.

Es sei im Folgenden stets $N = \{0, 1, \ldots, n-1\}$. Eine bijektive Abbildung von N auf sich heißt eine **Permutation**. Permutationen spielen in der gesamten Mathematik eine wichtige Rolle. Sind π und ρ Permutationen, so auch $\pi \circ \rho$, definiert durch $(\pi \circ \rho)(x) := \pi(\rho(x))$. (Man beachte also, dass erst ρ und dann π angewendet wird.) Ferner ist π^{-1} eine Permutation, wobei $\pi^{-1}(x) = y$, für das eindeutig bestimmte y mit $\pi(y) = x$. Schließlich ist die Identitätsabbildung, bezeichnet mit id_N, eine Permutation auf N. Die Permutationen bilden mit diesen Operationen eine Gruppe, die sogenannte **symmetrische Gruppe**. Dies kann man leicht nachrechnen.

Eine Permutation von N können wir notieren, indem wir für jedes Element einzeln das Bild angeben. Sei etwa π eine Permutation, so schreiben wir

$$\begin{pmatrix} 0 & 1 & \ldots & n-1 \\ \pi(0) & \pi(1) & \ldots & \pi(n-1) \end{pmatrix}$$

Daraus leitet man leicht ab, dass es genau $n!$ Permutationen auf N gibt, wenn N genau n Elemente enthält. Denn jede Permutation ist eindeutig durch die untere Zeile bestimmt. Diese ist eine Folge der Länge n, in der kein Element doppelt auftritt. Davon gibt es genau $n!$ Stück.

Bequemer als die eben gezeigte Darstellung ist die sogenannte Zyklenschreibweise, die wir jetzt entwickeln wollen.

Definition 136 *Es sei $\pi : N \to N$ eine Permutation. $M \subseteq N$ heißt* ***unter π abgeschlossen****, falls für alle $x \in M$ gilt $\pi(x) \in M$. Ein* ***Orbit*** *oder eine* ***Bahn*** *von π ist bezüglich Inklusion minimale Menge $M \subseteq N$, die unter π abgeschlossen ist. Ist $x \in M$, so heißt M* ***der Orbit*** *oder* ***die Bahn von*** *x* ***unter*** *π. $|M|$ heißt die* ***Länge*** *des Orbits M.*

Da N endlich ist, ist jedes Element x von N in einem Orbit von π enthalten. Ferner ist der Schnitt zweier Orbits wieder ein Orbit, sodass der Orbit von x

eindeutig bestimmt ist. Dieser besteht einfach aus allen Elementen der Form $\pi^i(x)$, $0 \leq i < k$.

Satz 137 *Es sei* $\pi : N \to N$ *eine Permutation und* M *ein Orbit von* π *der Länge* k. *Sei schließlich* $x \in M$ *beliebig gewählt. Dann ist stets* $M = \{\pi^i(x) : 0 \leq i < k\}$.

Beweis. Sei $x \in M$. Da M unter π abgeschlossen ist, muss $\pi(x) \in M$ sein. Aus dem gleichen Grund ist $\pi^2(x) \in M$, $\pi^3(x) \in M$ und so weiter. Da die Mächtigkeit von M genau k ist, existiert ein $i < k$ mit $\pi^k(x) = \pi^i(x)$. Ist nun $i > 0$, so gilt $\pi^{k-1}(x) = \pi^{i-1}(x)$, da π bijektiv ist. Es gibt also ein $n \leq k$ mit $\pi^n(x) = x$. Also ist $\{\pi^i(x) : 0 \leq i < n\}$ unter π abgeschlossen. Ist $n < k$, so ist M nicht die kleinste unter π abgeschlossene Menge, also kein Orbit. Daher ist $n = k$ und so $M = \{\pi^i(x) : 0 \leq i < k\}$. Für die zweite Behauptung wähle ein $x \in N$ und betrachte die Menge aller $\pi^m(x)$, $m \in \mathbb{N}$. Dies ist unter π abgeschlossen, und die kleinste unter π abgeschlossene Menge. Q. E. D.

Für eine beliebige Permutation π zerfällt also N in eine Partition von Orbits. Um eine Permutation festzulegen, genügt nicht die Angabe der Orbits. Man muss auch wissen, wie π auf den Orbits operiert. Die Einschränkung von π auf einen Orbit nennen wir einen **Zyklus von** π. Einen Zyklus gibt man an, indem man ein beliebiges Element $x \in M$ herausgreift und eine Liste der Elemente $x, \pi(x), \pi^2(x), \ldots$ angibt. Ein Zyklus wird wie folgt notiert: $(x, \pi(x), \pi^2(x), \ldots, \pi^{k-1}(x))$. Dann ist klar, dass x auf $\pi(x)$, $\pi(x)$ auf $\pi^2(x)$ geworfen wird — und so weiter. Ein Beispiel.

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 6 & 2 & 1 & 7 & 0 & 3 & 5 \end{pmatrix}$$

Diese Permutation besitzt die Orbits $\{0, 4, 5, 7\}$, $\{1, 3, 6\}$ und $\{2\}$. Wir notieren π also durch

$$(0475)(163)(2)$$

Dies bedeutet nach dem eben Gesagten, dass π 1 auf 3, 3 auf 6 und 6 auf 1 wirft; dass es 2 auf 2, 0 auf 4, 4 auf 7, 7 auf 5 und schließlich 5 auf 0 abbildet. Diese Darstellung ist eindeutig. Wenn die Menge N feststeht, kann man auf Zyklen der Länge 1 verzichten. Deshalb schreibt man die obenstehende Permutation auch

$$(0475)(136)$$

Wir sagen nun, eine Folge $F = \langle x_1, x_2, \ldots, x_k\rangle$ heißt ein **Zyklus** der **Länge** k, falls $x_i \neq x_j$ für alle $1 \leq i < j \leq k$. Ebenso nennen wir diejenige Abbildung π_F einen **Zyklus** welche definiert ist durch $\pi_F : x_i \mapsto x_{i+1}$, $1 \leq i < k$,

$\pi_F : x_k \mapsto x_1$, und $\pi_F : y \mapsto y$, falls y in F nicht vorkommt. Schließlich sei $\mathfrak{F} := \{F_1, F_2, \ldots, F_r\}$ eine Menge von Folgen, von denen je zwei kein Element gemeinsam haben. Diese bestimmen eine Permutation $\pi_{\mathfrak{F}} := \pi_{F_1} \circ \pi_{F_2} \circ \ldots \circ \pi_{F_r}$. (In unserem Beispiel ist also $\mathfrak{F} = \{\langle 0, 4, 7, 5\rangle, \langle 1, 6, 3\rangle\}$ oder auch $\{\langle 0, 4, 7, 5\rangle, \langle 1, 3, 6\rangle, \langle 2\rangle\}$. Es gibt allerdings noch mehr Möglichkeiten für $\mathfrak{F}$.) Diese Abbildung kann man auch wie folgt angeben. $\pi_{\mathfrak{F}} : x \mapsto \pi_F(x)$, falls x in F vorkommt, und $\pi_{\mathfrak{F}} : y \mapsto y$ sonst. Das Produkt hängt nicht von der Reihenfolge ab. Dies kann man direkt durch Nachrechnen bestätigen. Dazu nehme man ein beliebiges Element x. Falls x nicht in einem der Zyklen auftritt, so ist $\pi_{\mathfrak{F}}(x) = x$. Ferner ist $\pi_{F_i}(x) = x$ für alle F_i, in denen x nicht auftritt. Sei $x \in F_j$. Dann ist $\pi_{\mathfrak{F}}(x) = \pi_{F_j}(x)$, da auch $\pi_{F_j}(x) \in F_j$ und deswegen $\pi_{F_i}(x) = x$ für alle $i \neq j$. Dieses Argument lässt sich verallgemeinern.

Definition 138 *Es sei $f : N \to N$ eine Abbildung. $x \in N$ heißt **Fixpunkt** von f, falls $f(x) = x$. Wir bezeichnen mit $Fix(\pi)$ die Menge aller Fixpunkte von π.*

Klar ist, dass x genau dann Fixpunkt einer Permutation ist, wenn $\{x\}$ ein Orbit ist. In der verkürzten Zyklendarstellung lässt man also die explizite Angabe der Fixpunkte fallen.

Hilfssatz 139 *Es sei π eine Permutation auf N. Dann ist sowohl $Fix(\pi)$ als auch $N - Fix(\pi)$ unter π abgeschlossen. Ferner: ist ρ Permutation auf N, so ist $Fix(\pi \circ \rho) \supseteq Fix(\pi) \cap Fix(\rho)$.*

Der Beweis ist als Übung überlassen.

Satz 140 *Es seien π und ρ Permutationen auf N mit $N = Fix(\pi) \cup Fix(\rho)$. Dann ist $\pi \circ \rho = \rho \circ \pi$.*

Beweis. Es sei $x \in N$. Ist $x \notin Fix(\pi)$, so ist $x \in Fix(\rho)$ und so $\pi \circ \rho(x) = \pi(\rho(x)) = \pi(x)$ und $\rho \circ \pi(x) = \rho(\pi(x)) = \pi(x)$. Denn mit x ist auch $\pi(x) \notin Fix(\pi)$, also $\pi(x) \in Fix(\rho)$. Ebenso sieht man, dass für $x \in Fix(\pi)$ gilt $\pi \circ \rho(x) = \rho \circ \pi(x)$. Q. E. D.

Wir vereinbaren nun im Folgenden, dass wir das Produkt von Zyklen nur durch die Hintereinanderschreibung signalisieren. Ferner unterscheiden wir nicht zwischen dem Zyklus und der induzierten Permutation. Also ist (124) ein Zyklus und auch eine Permutation. Für die Identitätsabbildung schreiben wir (). Das Produkt von disjunkten Zyklen ist also kommutativ. Falls Zyklen nicht disjunkt sind, so kann man nicht viel mehr sagen, als dass die Zyklen des Produkts in der Vereinigung der Zyklen der Multiplikanden enthalten sind. Zum Beispiel ist $(134)(012) = (03412)$. Es kann auch zu kurzen Zyklen kommen, zum Beispiel in $(013)(031) = (0)(1)(3) = ()$.

Wir betrachten nun eine Permutation π und die Potenzen π^k. Zunächst einmal gilt: ist M unter π abgeschlossen, so ist M auch unter π^k abgeschlossen für jedes k. Insbesondere ist jeder Zyklus von π abgeschlossen unter π^k, also eine disjunkte Vereinigung von Zyklen von π^k. Er muss jedoch nicht immer ein Zyklus von π^k sein, wie wir noch sehen werden. Ist nun $\pi = \zeta_1\zeta_2\ldots\zeta_r$ eine Zerlegung von π in ein Produkt von Zyklen, so gilt $\pi^k = \zeta_1^k\zeta_2^k\ldots\zeta_r^k$, weil die Zyklen paarweise vertauschen. Als Beispiel nehmen wir $\pi = (0475)(163)$. Dann ist

$$\begin{array}{lcllcl}
\pi & = & (0475)(163) & \pi^2 & = & (07)(45)(136) \\
\pi^3 & = & (0547) & \pi^4 & = & (163) \\
\pi^5 & = & (0475)(136) & \pi^6 & = & (07)(45) \\
\pi^7 & = & (0574)(163) & \pi^8 & = & (136) \\
\pi^9 & = & (0475) & \pi^{10} & = & (07)(45)(163) \\
\pi^{11} & = & (0574)(136) & \pi^{12} & = & ()
\end{array}$$

Was ist nun das kleinste k derart, dass $\pi^k = ()$? Dies nennen wir den **Index** von π. Für einen Zyklus ist die Antwort nicht schwer. Ist Z ein Zyklus der Länge k, so ist k gleichzeitig der Index der Permutation. Ferner ist $Z^m = ()$ genau dann, wenn m ein Vielfaches des Index von Z ist. Daraus folgt mit Satz 140 unmittelbar der

Satz 141 *Es sei π eine Permutation. Der Index von π ist das kleinste gemeinsame Vielfache aller Indizes von Zyklen von π.*

Eine **Transposition** ist ein Zyklus der Länge 2.

Satz 142 *Jede Permutation kann dargestellt werden als das Produkt von Transpositionen.*

Beweis. Da wir eine Permutation als Produkt von Zyklen darstellen können, genügt es zu zeigen, dass jeder Zyklus das Produkt von gewissen Transpositionen ist. Ohne Beschränkung der Allgemeinheit ist ein Zyklus von der Form $\zeta = (0\ 1\ 2\ldots k-1)$. Betrachte das Produkt $\pi := (1\ 2)(2\ 3)\ldots(k-2\ k-1)(k-1\ 0)$. Man rechnet nach, dass $\pi(0) = 1$, $\pi(1) = 2$ und so weiter, $\pi(k-1) = 0$. Also $\pi = \zeta$. Q. E. D.

Es sei nun $\theta_1\theta_2\ldots\theta_m = \eta_1\eta_2\ldots\eta_n$ zwei Darstellungen von π als Produkt von Transpositionen. Es gilt nun zwar nicht $m = n$ (wie man leicht sieht, kann man das nicht erwarten), aber wie wir zeigen werden ist $m - n$ stets eine gerade Zahl. Wir nennen deshalb $(-1)^m$ das **Signum** von π. Das Signum eines Zyklusses der Länge k ist also $(-1)^{k-1}$. Da $m - n$ gerade ist, ist nämlich $(-1)^n = (-1)^m$, und so ist diese Zahl unabhängig von der gewählten Darstellung als Produkt. Wir werden dies über einen Umweg beweisen.

Definition 143 *Eine $n \times n$–**Permutationsmatrix** ist eine $n \times n$–Matrix mit Einträgen 0 oder 1 derart, dass in jeder Zeile sowie in jeder Spalte genau einmal 1 auftritt. Insbesondere ist ein Zyklus der Länge k ein Produkt von $k - 1$ Transpositionen.*

Sei e_i derjenige Vektor, welcher genau an der Stelle i eine 1 hat, und sonst 0 ist. Wir schreiben $e_i = (\delta_{ij})_{1\leq j\leq n}$, wobei $\delta_{ij} = 1$ genau dann, wenn $i = j$, und 0 sonst. (δ_{ij} ist das sogenannte *Kroneckersymbol*, benannt nach Leopold Kronecker, 1823 – 1891.) Wir können jeder Permutation π genau eine Permutationsmatrix $A(\pi)$ mit $A(\pi)\cdot e_i = e_{\pi(i)}$ für alle $1 \leq i \leq n$ beschreiben. Denn es ist $A(\pi) = (a_{ij})_{1\leq i,j\leq n}$. Also ist der $k$te Eintrag in $A(\pi)\cdot e_i$ gerade $\sum_{j=1}^{n} a_{kj}\delta_{ji} = a_{ki}$. Dieser muss nun gerade gleich 0 sein, wenn $k \neq \pi(i)$ und 1, falls $k = \pi(i)$. Also haben wir $a_{ij} = 1$ genau dann, wenn $i = \pi(j)$, und 0 sonst. Dies ist eine Permutationsmatrix. Zum Beispiel sei $\pi = (0\ 2\ 3)(1)$. Dann ist

$$A(\pi) = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Denn zum Beispiel ist $\pi(0) = 2$. Daher $a_{20} = 0$. Man überzeuge sich, dass $A(\pi)e_0 = e_2$, $A(\pi)e_1 = e_1$, $A(\pi)e_2 = e_3$ sowie $A(\pi)e_3 = e_0$ ist. Sei umgekehrt B eine Permutationsmatrix. Dann ist $B \cdot e_i = e_j$ für ein gewisses j, und man rechnet nach, dass es eine Permutation π geben muss mit $B \cdot e_i = e_{\pi(i)}$. Also $B = A(\pi)$. Ist dies erst einmal etabliert, so betrachte man zwei Permutationen, π und ρ. Es ist $\rho\circ\pi$ eine Permutation, und zu ihr gehört eine Permutationsmatrix, $A(\rho\circ\pi)$. Nun ist $\rho\circ\pi(i) = \rho(\pi(i))$, und so $A(\rho\circ\pi)e_i = A(\rho)(A(\pi)(e_i))$ für alle $1 \leq i \leq n$. Daraus folgt schließlich der

Satz 144 *Zu jeder Permutation π existiert eine eindeutig bestimmte Permutationsmatrix $A(\pi)$ mit $A(\pi)e_i = e_{\pi(i)}$ für alle i mit $1 \leq i \leq n$. Ferner gilt $A(\rho\circ\pi) = A(\rho)\circ A(\pi)$.*

Nun definieren wir

$$sgn(\pi) := det(A(\pi))$$

Wir werden zeigen, dass $sgn(\pi)$ unser vorher definiertes Signum ist. Alles andere folgt aus dieser Tatsache.

Hilfssatz 145 *Es sei π eine Permutation. Dann gilt*

$$det(A(\pi)) = \pm 1$$

Beweis. Es sei $M = (m_{ij})_{1\leq i,j\leq n}$ eine $n \times n$–Matrix. Dann bezeichne M^{ij} diejenige $(n-1)\times(n-1)$–Matrix, die aus M durch Streichen der $i$ten Zeile und der $j$ten Spalte hervorgeht. Der Determinantenentwicklungssatz besagt

$$det(M) = \sum_{i=1}^{n} (-1)^{i+1} m_{1i} \cdot det M^{1i}$$

Wir beweisen die Behauptung nun durch Induktion über die Anzahl n der permutierten Elemente. Ist $n = 1$, so ist $A(\pi) = (1)$ und $det(A(\pi)) = 1$. Nun sei die Behauptung für n gezeigt, und sei π eine Permutation über $n+1$ Elemente. Ferner sei i so gewählt, dass $\pi(i) = 1$. Dann ist nach dem Entwicklungssatz

$$det(A(\pi)) = \sum_{i=1}^{n} (-1)^{i+1} a_{1i} \cdot det M^{1i} = (-1)^{j+1} det(A(\pi)^{1j})$$

Hierbei ist $j = \pi^{-1}(1)$. Die Behauptung folgt aus der leicht zu prüfenden Tatsache, dass $A(\pi)^{1j}$ wiederum eine Permutationsmatrix ist. Q. E. D.

Als nächstes benötigen wir den Produktsatz für Determinanten. Dieser besagt, dass $det(A \cdot B) = det(A) \cdot det(B)$. Als letztes benötigen wir noch die Tatsache, dass für eine Transposition τ $det(A(\tau)) = -1$. Dies zeigt man so. Bis auf Umbenennen der Elemente ist $\tau = (12)$. Ist aber π bis auf Umbenennung gleich ρ, so gilt $det(A(\pi)) = det(A(\rho))$. (Denn die Matrix der Umbennung ist eine Permutationsmatrix $A(\sigma)$ und so ist $A(\rho) = A(\sigma)^{-1}A(\pi)A(\sigma)$. Daher ist $det(A(\rho)) = det(A(\pi))$. Wer dies umständlich findet: der Beweis benötigt dies auch nicht wirklich.) Wendet man den Entwicklungssatz an, so bekommt man

$$det(A(\tau)) = det\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right) = -1 \ .$$

Jetzt folgt: ist π das Produkt von irgendwelchen Transpositionen τ_j, $1 \leq j \leq r$, so ist $sgn(A(\pi)) = (-1)^r$. Daraus folgt unsere Behauptung, dass $sgn(\pi)$ genau das Signum von π ist. Dies bedeutet auch, dass das Signum unabhängig von der gewählten Zerlegung ist, und somit die Anzahl der Transpositionen, aus denen π zusammengesetzt ist, entweder immer gerade ist oder immer ungerade ist.

17. Teil: Kombinatorik III. Verteilungen.

Im Folgenden wollen wir uns mit Verteilungsproblemen befassen. Einige haben wir schon vorher kennengelernt. Das Abzählproblem wird hier in folgender Fassung erscheinen:

> Gegeben eine Menge N bestehend aus n Bällen sowie eine Menge K von k Fächern. Auf wie viele Weisen lassen sich die Bälle auf die Fächer verteilen? Was passiert mit den Anzahlen, wenn die Bälle und/oder die Fächer nicht unterschieden werden?

Zunächst das einfachste der Probleme: die Verteilung der Bälle auf die Fächer, wobei sowohl die Bälle als auch die Fächer unterschieden werden. Da jeder Ball in genau einem Fach eingeteilt wird, ist eine Verteilung der Bälle auf die Fächer schlicht eine Funktion von N nach K. Die Anzahl der Funktionen haben wir schon errechnet. Sie ist n^k, wo $n = |N|$ und $k = |K|$. Nun betrachten wir die Anzahl der Verteilungen von Bällen auf Fächer, wenn wir einerseits die Bälle nicht unterscheiden oder andererseits die Fächer nicht unterscheiden, oder aber beide nicht. Wir wollen kurz sagen, was das bedeutet. Es bedeutet nicht notwendig, dass die Objekte wirklich ununterscheidbar sind. Es bedeutet, dass Verteilungen nicht als verschieden gelten (und damit nicht extra gezählt werden), wenn man die Objekte untereinander austauscht. Wenn das der Fall ist, bekommt man unter anderem eine ganz andere Wahrscheinlichkeitsrechnung. Hier ist ein Gedankenexperiment. Die Bosonen gelten als wirklich ununterscheidbar. Wenn man 2 Bosonen auf 2 'Fächer' verteilt, so erhält man drei Verteilungen: wenn jedes Fach je ein Boson enthält, und wenn eines der Fächer zwei Bosonen enthält. Die Wahrscheinlichkeit wäre dann je ein Drittel. In der Tat findet man allerdings eine Verteilung 2:1:1. Auch wenn wir also die Teilchen nicht unterscheiden können — sie sind verschieden und verhalten sich entsprechend.

Wir wollen also jetzt so tun, als seien die Bälle bzw. die Fächer ununterscheidbar. Beide Möglichkeiten werden oft benutzt. Der Unterschied zwischen *unterscheidbar* und *ununterscheidbar* kommt zum Beispiel in dem Unterschied zwischen einer Menge und einer nichtwiederholenden Folge heraus. Eine Folge F ordnet jedem Folgenglied einen Platz zu; die zugehörige Menge der Folgenglieder tut dies nicht, sie sieht von der Reihenfolge ab. Wir veranschaulichen den Unterschied durch ein Beispiel. Es sei $N = \{0, 1, 2, 3\}$ und $K = \{a, b, c\}$. Betrachte folgende Verteilungen:

V_1	V_2
a:()	a:()
b:(0)	b:(123)
c:(123)	c:(0)

V_3	V_4
a:()	a:(2)
b:(2)	b:(013)
c:(013)	c:()

Falls wir sowohl die Bälle unterscheiden wie auch die Fächer, so sind alle Verteilungen verschieden. Falls wir die Fächer nicht mehr unterscheiden, so sind V_1 und V_2 sowie V_3 und V_4 nicht mehr zu unterscheiden. Am Besten sieht man das so: wir entfernen von den Fächern die Namen a, b und c. Dann haben wir nur noch die Information, dass bei V_1 ein Fach keinen Ball, ein anderes den Ball 1 und ein drittes die Bälle 1, 2 und 3 enthält. Anders ausgedrückt: falls wir durch Vertauschen der Namen der Fächer eine Verteilung V in die Verteilung V' überführen können, so sind V und V' in dem Falle gleich, wo wir die Fächer nicht mehr unterscheiden. Man sieht nun ebenso, dass wenn wir die Bälle nicht mehr unterscheiden, nunmehr V_1 und V_3 gleich sind. Falls wir schließlich weder Fächer noch Bälle unterscheiden wollen, so sind alle vier Verteilungen gleich. In der folgenden Definition wollen wir ungeordnete Folgen definieren. Die Definition mag etwas umständlich erscheinen, aber sie ist relativ einleuchtend, wenn man sie von der praktischen Seite her sieht. Eine ungeordnete Folge ist eine Folge, in der es nicht auf den Platz ankommt, den ein Element einnimmt. Dies drücken wir so aus, dass wir eine ungeordnete Folge als die Menge aller geordneten Folgen auffassen, welche durch Umordnung ineinander übergehen.

Definition 146 *Es sei X eine Menge, und F und G n–lange Folgen von*

Elementen aus X. Für $x \in X$ sei $j(x, F)$ die Anzahl der Folgenglieder, welche gleich x sind. Dies heiße der ***Index*** *von x in F. Wir setzen $F \approx G$, falls für alle Elemente x aus X gilt $j(x, F) = j(x, G)$. Die Menge $M(F) := \{G : G \approx F\}$ heißt auch eine* ***Multimenge*** *M mit Elementen aus X oder* ***ungeordnete Folge****. Die* ***Mächtigkeit*** *von M ist definiert als die Länge von F.*

Diese Definition ist nicht besonders handlich. Wir notieren Multimengen so:

$$\{a, b, b, a, c, d, d, a, d\}_m$$

Der Index $_m$ deutet an, dass es sich um eine Multimenge handelt. Mengen sind Multimengen, in denen der Index nur 0 oder 1 ist. Wie auch sonst üblich, ist die oben hingeschriebene Folge nur ein Vertreter. Dieselbe Multimenge ist beschrieben durch

$$\{d, d, d, c, b, b, a, a, a\}_m$$

Nehmen wir also wieder unsere Bälle und Fächer. Seien sowohl Bälle als auch Fächer unterschieden. Dann existieren in der Tat k^n viele Verteilungen. Seien nun die Bälle unterschieden, nicht aber die Fächer. Dann entspricht einer Verteilung schlicht eine Partition der Menge N in höchstens k Mengen. (Man beachte: Partitionsmenge sind nicht leer, aber die Fächer müssen in einer Verteilung nicht gefüllt werden. Daher dürfen wir die Anzahl der Verteilung nicht mit der Anzal der Partitionen in *genau* k Mengen gleichsetzen.) Ist $k > n$, so existiert keine Partition. Die Anzahl der Partitionen einer n–elementigen Menge in genau k Mengen bezeichnet man mit $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$. Sie heißen **Stirlingzahlen der zweiten Art**. (Man mag, in Analogie zu den Binomialkoeffizienten, auch für Mengen N und Zahlen k $\left\{ \begin{matrix} N \\ k \end{matrix} \right\}$ als die Menge der k–elementige Partitionen von N bezeichnen. Dann ist $\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = |\left\{ \begin{matrix} N \\ k \end{matrix} \right\}|$, wenn $n = |N|$.) Natürlich muss stets $k \leq n$ sein. Ferner gilt, wie man leicht sieht, $\left\{ \begin{matrix} n \\ 1 \end{matrix} \right\} = 1$ und $\left\{ \begin{matrix} n \\ n \end{matrix} \right\} = 1$. Außerdem ist $\left\{ \begin{matrix} n \\ 0 \end{matrix} \right\} = 0$, falls $n \neq 0$.

n	$\left\{ \begin{matrix} n \\ 0 \end{matrix} \right\}$	$\left\{ \begin{matrix} n \\ 1 \end{matrix} \right\}$	$\left\{ \begin{matrix} n \\ 2 \end{matrix} \right\}$	$\left\{ \begin{matrix} n \\ 3 \end{matrix} \right\}$	$\left\{ \begin{matrix} n \\ 4 \end{matrix} \right\}$	$\left\{ \begin{matrix} n \\ 5 \end{matrix} \right\}$
0	1					
1	0	1				
2	0	1	1			
3	0	1	3	1		
4	0	1	7	6	1	
5	0	1	15	25	10	1

Es fällt auf, dass zum Beispiel $\left\{ \begin{matrix} n \\ 2 \end{matrix} \right\} = 2^{n-1} - 1$ ist. Dies kann man direkt beweisen. Denn wenn es zwei Partitionsmengen gibt, so haben wir einfach eine Einteilung von n (als Menge) in eine Menge H und ihr Komplement. Dabei dürfen weder H noch $n - H$ leer sein. Es existieren $2^n - 2$ solcher Mengen H. Da allerdings das Paar $\langle H, n - H\rangle$ und das Paar $\langle n - H, n\rangle$ dieselbe Partition bilden, so bekommen wir insgesamt $2^{n-1} - 1$ Partitionen. Man kann nun etwas allgemeiner das Folgende zeigen.

Satz 147 *Für die Stirlingzahlen zweiter Art gelten die folgenden Rekursionsvorschriften. (Hier wird $n > 0$ vorausgesetzt.)*

$$\begin{aligned} \left\{ \begin{matrix} n \\ 1 \end{matrix} \right\} &= 1 \\ \left\{ \begin{matrix} n \\ n \end{matrix} \right\} &= 1 \\ \left\{ \begin{matrix} n+1 \\ k \end{matrix} \right\} &= k \left\{ \begin{matrix} n \\ k \end{matrix} \right\} + \left\{ \begin{matrix} n \\ k-1 \end{matrix} \right\} \end{aligned}$$

Beweis. Es sei N eine Menge mit n Elementen und $x \notin N$. Wir fragen nach der Anzahl der Partitionen von $N \cup \{x\}$ in k Mengen. Sei Π eine solche Partition. Fall 1, $\{x\} \in \Pi$. Dann ist $\Pi - \{\{x\}\}$ eine Partition von N in $k - 1$ Mengen. Umgekehrt ist für jede Partition Ξ von N in $k-1$ Mengen $\Xi \cup \{\{x\}\}$ ein Partition von $N \cup \{x\}$ in k Mengen. Fall 2, $\{x\} \notin \Pi$. Dann existiert eine Menge $H \in \Pi$, welche x enthält und noch ein weiteres Element. Dann sei Π' diejenige Partition von N, die aus Π entsteht, wenn wir H durch $H - \{x\}$ ersetzen. Da $H - \{x\} \neq \varnothing$, ist dies eine Partition von N in k Mengen. Nun sei umgekehrt eine Partition Π von N in k Mengen gegeben. Dann sei Π' ein Partition, die entsteht, indem wir zu einer beliebigen Partitionsmenge das neue Element x hinzufügen. Dann ist je nach Wahl dieser Menge eine neue Partition entstanden. Wir haben also für jedes Π insgesamt k viele Partitionen von $N \cup \{x\}$. Q. E. D.

Um keine Missverständnisse zu erzeugen: dies ist nicht eine primitive Rekursion, wie wir sie im Teil 3 betrachtet haben. Trotzdem folgen aus ihr alle Partitionszahlen, wie man sich leicht überlegt. (Dies zeigt man durch Induktion über n, beginnend mit $n = 1$.) Aus dieser Rekursionsvorschrift bekommen wir übrigens für $k = 2$ die Formel

$$\left\{ \begin{matrix} n+1 \\ 2 \end{matrix} \right\} = 2 \left\{ \begin{matrix} n \\ 2 \end{matrix} \right\} + \left\{ \begin{matrix} n \\ 1 \end{matrix} \right\} = 2 \left\{ \begin{matrix} n \\ 2 \end{matrix} \right\} + 1$$

welche genau die Lösung $\left\{ \begin{matrix} n \\ 2 \end{matrix} \right\} = 2^{n-1} + 1$ hat (wie man leicht nachrechnet), wobei der Anfangswert gegeben ist durch $\left\{ \begin{matrix} 2 \\ 2 \end{matrix} \right\} = 1$.

Falls wir nun auch noch die Bälle nicht mehr unterscheiden, so ist die Anzahl der Möglichkeiten gerade die Anzahl der Möglichkeiten, die Zahl n in höchstens k von Null verschiedene Summanden zu zerlegen. Die Zahl $P_{n,k}$ sei die Anzahl aller Zerlegungen von n in genau k von Null verschiedene Summanden. (Oder auch: $P_{n,k}$ ist die Anzahl der monoton aufsteigenden Folgen von Zahlen $\neq 0$ der Länge k deren Summe n ist.) Zum Beispiel ist $P_{7,3} = 4$, denn

$$\begin{aligned} 7 &= 1+1+5 \\ &= 1+2+4 \\ &= 1+3+3 \\ &= 2+2+3 \end{aligned}$$

Die Zahlen $P_{n,k}$ sind sehr schwierig zu berechnen. Es gilt aber folgender Sachverhalt.

Satz 148 *Sei $n > 0$. Es ist $P_{n,1} = 1$, $P_{n,2} = \frac{n}{2}$, falls n gerade und $P_{n,2} = \frac{n-1}{2}$, falls n ungerade.*

Beweis. Es ist klar, dass eine beliebige Zahl sich nur auf eine Weise als Summe einer einzigen Zahl darstellen lässt. Nun sei $n = 2k$ eine gerade Zahl. Dann ist für beliebiges i mit $0 < i < n$, $n = i + (n - i)$. Um Doppelzählungen zu vermeiden, überlegen wir uns, dass, falls $i > k$ ist, $n - i < k$ ist. Wir betrachten deswegen nur solche Summen $i_1 + i_2$, in denen $i_1 \leq i_2$ ist. (Dieses Verfahren wendet man ganz allgemein bei der Bestimmung der $P_{n,k}$ an.) Zu jeder Zahl i mit $0 < i \leq k$ gibt es genau eine Zerlegung von n, nämlich $n = i + (n - i)$, und es ist $i \leq n - i$. Für zwei Darstellungen $i + (n - i) = j + (n - j)$ mit $0 < i, j \leq k$ gilt $i = j$. Wir haben also exakt k viele solcher Darstellungen, das heißt genau $\frac{n}{2}$ viele. Falls nun n ungerade ist, also $n = 2k + 1$, muss man beachten, dass wiederum $0 < i \leq k$ muss. Dies ergibt den Wert $\frac{n-1}{2}$. Q. E. D.

Will man $P_{n,k}$ für $k > 2$ exakt ausrechnen, muss man einige Mühe aufwenden. Betrachten wir kurz den Fall $k = 3$. Zwei Darstellungen $n = i_1 + i_2 + i_3$ und $n = j_1 + j_2 + j_3$ sind gleich, falls $\{i_1, i_2, i_3\} = \{j_1, j_2, j_3\}$. Eine Darstellung $n = i_1 + i_2 + i_3$ kann man immer so wählen, dass $i_1 \leq i_2 \leq i_3$. Ist dann $i_1 + i_2 + i_3 = j_1 + j_2 + j_3$ mit $j_1 \leq j_2 \leq j_3$, so gilt $i_1 = j_1$ und $i_2 = j_2$ und $i_3 = j_3$, oder aber es gilt $\{i_1, i_2, i_3\} \neq \{j_1, j_2, j_3\}$ (das heißt, die Darstellungen sind ungleich). Dies bedeutet, dass wir jede Darstellung tatsächlich nur einmal zählen. Nun geht man so vor. Man wählt i_1, und bestimmt die Anzahl aller Darstellungen von $n - i_1$ in Summen $i_2 + i_3$, wo $i_1 \leq i_2 \leq i_3$. Dies bedeutet, wir wollen nur solche Darstellungen von $n - i_1$ betrachten, in denen alle Summanden $\geq i_1$ sind. Man überlege sich, dass dies gerade die Anzahl aller Darstellungen von $n - 3i_1 + 2$ in Summen $k_1 + k_2$, $k_1, k_2 > 0$, ist. Denn ist $n - 3i_1 + 2 = k_1 + k_2$, so ist

$$n - i_1 = (k_1 + i_1 - 1) + (k_2 + i_1 - 1)$$

Mit $k_1 > 0$ ist $k_1 + i_1 - 1 \geq i_1$, und mit $k_1 \leq k_2$ ist $k_1 + i_1 - 1 \leq k_2 + i_1 - 1$. Dies erledigt die Darstellung von $n - i_1$. Nun muss man im Prinzip nur aufsummieren. Man beachte, dass man stets $i_1 \leq \frac{n}{3}$ hat. Größere i_1 muss man nicht betrachten.

Als letztes betrachten wir den Fall, wo zwar die Fächer unterschieden werden, nicht aber die Bälle. Dann ist eine Verteilung einzig bestimmt durch die Anzahl der Bälle, welche in einem gegebenen Fach liegen. Dies entspricht aber genau einer Multimenge über K, in der die Summe der Indizes gerade n ist. Die Summe aller Indizes $j(x, F)$ ist aber gerade die Mächtigkeit der durch F repräsentierten Multimenge. Die Anzahl der Multimengen der Mächtigkeit k über einer Menge N der Mächtigkeit n lässt sich durch einen Trick berechnen. Ohne Beschränkung der Allgemeinheit ist $N = n = \{0, 1, \ldots, n - 1\}$. Zu jeder Multimenge gibt es genau eine Auflistung der Elemente, die monoton wachsend ist. (Etwa ist $\{0, 0, 1, 1, 2\}_m$ eine solche Auflistung, nicht aber $\{2, 0, 0, 1, 1\}_m$.) Sei $M = \{x_0, x_1, \ldots, x_{k-1}\}_m$ eine Multimenge mit $x_i \leq x_{i+1}$

für alle $i < k-1$. Nun setze $A(M) := \{x_0, x_1+1, x_2+2, x_3+3, \ldots, x_{k-1}+k-1\}$. $A(M)$ ist wohlgemerkt eine Menge, und es ist $x_i + i - 1 < x_{i+1} + i$, nach Wahl der x_i. $A(M)$ ist eine k–elementige Teilmenge von $\{0, 1, \ldots, n+k-2\}$. Sei $Y \subseteq \{0, 1, \ldots, n+k-2\}$ eine k–elementige Menge, etwa $Y = \{y_0, y_1, \ldots, y_{k-1}\}$ mit $y_i < y_{i+1}$. Dann setze $B(Y) := \{y_0, y_1 - 1, y_2 - 2, \ldots, y_{k-1} - (k-1)\}$. $B(Y)$ ist der Repräsentant einer Multimenge der Mächtigkeit k über B. Diese Beziehung ist bijektiv. Also ergibt sich folgender Satz.

Satz 149 *Die Anzahl der Multimengen der Mächtigkeit k über einer Menge der Mächtigkeit n ist*

$$\binom{n+k-1}{k-1} = \frac{(n+k-1)^{\underline{k}}}{k!}$$

Wir werden noch einen zweiten Beweis dieses Sachverhalts geben, der auch instruktiv ist. Wir betrachten einen Automaten, der die Fächer namens 0, 1, bis $k-1$ füllt, beginnend mit 0 und schrittweise nach oben vorrückend. Er wird mit zwei Knöpfen gesteuert. Der Knopf `b` lässt den Automaten einen Ball in das aktuelle Fach füllen, der Knopf `v` lässt den Automaten ein Fach weitergehen. Offensichtlich kann man eine Befüllung der Fächer erreichen, indem man n mal `b` drückt und $k-1$ mal `v`. Dann ist der Apparat bei $k-1$ angekommen, und er hat alle Bälle abgegeben. Jede Folge der Länge $n+k-1$ bestehend aus n Vorkommen von `b` bestimmt eine verschiedene Verteilung. Es gibt genau $\binom{n+k-1}{k-1}$ viele Folgen, wie wir schon berechnet haben.

Die Ergebnisse fassen wir in folgende Tabelle zusammen.

		Fächer (k)	
		unterschieden	nicht unterschieden
Bälle (n)	unterschieden	k^n	$\sum_{i\leq k} \left\{ {n \atop i} \right\}$
	nicht unterschieden	$\binom{n+k-1}{k-1}$	$\sum_{i\leq k} P_{n,i}$

Der Grund, warum wir in der rechten Spalte nicht die einfachen Stirlingzahlen wie auch die einfachen Partitionszahlen haben, ist dieser: die Anzahl der Verteilungen von n Bällen auf k nicht unterschiedene Fächer induziert nicht notwendig eine Partition in k Mengen, weil Fächer ja auch leer sein dürfen, während Partitionsmengen nicht leer sind. Dewegen induziert sie nur eine Partition in *höchstens* k Mengen. Die Anzahl der Partitionen in höchstens k Mengen ist aber gerade die Summe der Anzahlen der Partitionen in i Mengen, wo $i \leq k$. ($i = 0$ ist zwar eigentlich nicht möglich, wenn $n > 0$ ist, aber es ist ohnehin $\left\{ \begin{matrix} n \\ 0 \end{matrix} \right\} = 0$.) Genauso ist es mit den Zahlpartitionen. Was wir suchen, ist nicht die Anzahl der Zerlegungen in *exakt* k Summanden, sondern in *höchstens* k Summanden, was gerade die Summe der Anzahlen der Zerlegungen in i Summanden ist, mit $i \leq k$.

18. Teil: Graphen I. Grundlegende Definitionen.

Das Material für die folgenden vier Teile wurde überwiegend entnommen aus M. Aigner: *Diskrete Mathematik*, Vieweg Verlag, 1993.

Bevor wir beginnen, wollen wir uns mit dem Rechnen in Restklassen beschäftigen, da dies in Zukunft häufiger auftritt. Es sei n eine beliebige natürliche Zahl, $n > 1$. Dann existiert zu jeder ganzen Zahl k eine Darstellung $k = c \cdot n + r$, wobei $0 \leq r < n$. r heißt der **Rest modulo** n von k. Wir schreiben $k \equiv \ell \pmod{n}$ um auszudrücken, dass k und ℓ den gleichen Rest modulo n haben und sagen, *k ist kongruent zu r modulo n*. Die Relation *x ist kongruent modulo n zu y* (in Zeichen $x \equiv y \pmod{n}$) ist eine Äquivalenzrelation, wie man leicht bestätigt. Man überlegt sich ferner, dass $k \equiv \ell \pmod{n}$ genau dann, wenn es $c, d, r \in \mathbb{Z}$ gibt mit $k = c \cdot n + r$ und $\ell = d \cdot n + r$. Dabei muss r nicht zwischen 0 und $n-1$ liegen! Es gilt folgender Sachverhalt.

Satz 150 *Es sei* $k_1 \equiv \ell_1 \pmod{n}$ *und* $k_2 \equiv \ell_2 \pmod{n}$. *Dann ist*

$$k_1 + k_2 \equiv \ell_1 + \ell_2 \pmod{n}, \qquad k_1 \cdot k_2 \equiv \ell_1 \cdot \ell_2 \pmod{n}$$

.

Beweis. Es sei $k_i = c_i \cdot n + r_i$, $\ell_i = d_i \cdot n + r_i$. Dann ist $k_1 + k_2 = (c_1 + c_2) \cdot n + (r_1 + r_2)$ sowie $\ell_1 + \ell_2 = (d_1 + d_2) \cdot n + (r_1 + r_2)$. Also $k_1 + k_2 \equiv r_1 + r_2 \pmod{n}$ und $\ell_1 + \ell_2 \equiv r_1 + r_2 \pmod{n}$. Daher $k_1 + k_2 \equiv \ell_1 + \ell_2 \pmod{n}$. Ebenso sieht man $k_1 \cdot k_2 \equiv \ell_1 \cdot \ell_2 \pmod n$, denn es ist $k_1 \cdot k_2 = (c_1c_2n + c_1r_2 + c_2r_1)n + r_1r_2$ und ebenso $\ell_1 \cdot \ell_2 = (d_1d_2n + d_1r_2 + d_2r_1)n + r_1r_2$. Q. E. D.

Wir können also folgende Addition und Multiplikation auf $\{0, 1, \ldots, n-1\}$ definieren. Es sei $x \oplus_n y$ das eindeutig bestimmte $r \in \{0, 1, \ldots, n-1\}$ mit $x + y \equiv r \pmod n$. Ebenso sei $x \odot_n y$ das eindeutig bestimmte $r \in \{0, 1, \ldots, n-1\}$ mit $xy \equiv r \pmod n$. Wir schreiben in aller Regel $x + y$ anstelle von $x \oplus_n y$ und $x \cdot y$ anstelle von $x \odot_n y$. Es existiert zu x stets ein Element y mit $x \oplus_n y = 0$. Wir notieren es mit $-x$. Die üblichen Gesetze für Addition, Subtraktion und Multiplikation gelten. Ferner: ist n eine Primzahl, so existiert für jedes $x \neq 0$ ein y mit $x \odot y = 1$. Wir behaupten, dass nämlich $y \neq y'$ bedeutet $x \odot_n y \neq x \odot_n y'$. Andernfalls ist $x \odot_n (y - y') = 0$. Nun ist n eine Primzahl; n teilt das Produkt $x(y - y')$. Da n die Zahl x nicht teilt, teilt es $y - y'$. Also $y \equiv y' \pmod n$ und so $y = y'$. Wir merken noch Folgendes an: ist n keine Primzahl, so existieren Zahlen x, x' mit $0 < x, x' < n$ und $x \cdot x' \equiv 0 \pmod n$, also $x \odot_n x' = 0$.

Ein Paar $\mathfrak{G} = \langle E, K \rangle$, wobei E eine beliebige, nicht leere Menge ist und $K \subseteq \binom{E}{2}$, heißt ein **Graph**. Falls nichts anderes gesagt wird, ist E endlich, und damit ist natürlich auch K endlich. E ist die Menge der **Ecken** und K die Menge der **Kanten** des Graphen. Eine Kante ist also nichts anderes als eine Paarmenge $\{u, v\} \subseteq E$. Wir schreiben oft uv (oder vu) für die Kante $\{u, v\}$. Wir sagen, u und v seien **benachbart** oder **adjazent** (in $\mathfrak{G}$), falls $uv \in K$. Ist ferner $u \in k$, $k \in K$, so heißt u mit k **inzident**. Zwei Kanten k und ℓ heißen **inzident**, falls $k \cap \ell \neq \varnothing$, das heißt, wenn sie eine gemeinsame Ecke haben. Es seien $\mathfrak{G} = \langle E, K \rangle$ und $\mathfrak{G}' = \langle E', K' \rangle$ Graphen. Eine Abbildung $h : E \to E'$ heißt **Isomorphismus von $\mathfrak{G}$ nach $\mathfrak{G}'$**, falls (a) h nijektiv ist und (b) für alle $x, y \in E$: $\{x, y\} \in K$ genau dann, wenn $\{h(x), h(y)\} \in K'$. Ein paar Beispiele für Graphen.

Beispiel 1. Sei $K = \binom{E}{2}$. Diese Graphen heißen **vollständig**. Ist $|E| = n$, so bezeichnet man diesen Graphen mit K_n. (Dies ist — bis auf Isomorphie — eindeutig.)

Beispiel 2. Sei $E = S \cup T$, wobei $S \cap T = \varnothing$. Ferner bestehe jede Kante aus je einem Element aus S und einem Element aus T. Dann heißt der Graph **bipartit**. Er heißt vollständig bipartit, falls $K = \{\{u, v\} : u \in S, v \in T\}$. Ist $|S| = m$ und $|T| = n$, so bezeichnen wir den Graphen mit $K_{m,n}$.

Beispiel 3. Sei $E = \{e_0, e_1, \ldots, e_{n-1}\}$, und $K = \{\{e_i, e_{i+1}\} : i < n - 1\}$. Dieser Graph heißt **linearer Graph der Länge** n.

Beispiel 4. Sei $E = \{e_0, e_1, \ldots, e_{n-1}\}$ und $K = \{\{e_i, e_{i+1}\} : i < n - 1\} \cup \{\{e_{n-1}, e_0\}\}$. Dieser Graph heißt schlicht ein **Kreis** der **Länge** n. (Wir können auch schreiben $K = \{\{e_i, e_j\} : j \equiv i + 1 \pmod n\}$.)

Beispiel 5. Es sei E die Menge der Folgen der Länge n über $\{0, 1\}$. Es sei $d(\vec{x}, \vec{y}) := |\{i : x_i \neq y_i\}|$ der sogenannte **Hammingabstand** von $\vec{x}$ und $\vec{y}$. Betrachte $K = \{\{\vec{x}, \vec{y}\} : d(\vec{x}, \vec{y}) = 1\}$. Dieser Graph heißt **(Hyper–)Würfel der Dimension** n. Für $n = 2$ ist dies genau ein Quadrat, für $n = 3$ genau der allbekannte Würfel.

Beispiel 6. Der **Petersen–Graph** sieht wie folgt aus. Es ist $E = E_i \cup E_a$, wobei $E_i = \{i_0, i_1, \ldots, i_4\}$ und $E_a = \{a_0, a_1, \ldots, a_4\}$. $K = K_a \cup K_{ia} \cup K_i$, wobei $K_a = \{\{i_m, i_n\} : m \equiv n + 1 \pmod 5\}$, $K_{ia} = \{\{i_k, a_k\} : 0 \leq i < 5\}$ und $K_i = \{\{i_p, i_q\} : q \equiv p+2, p+3 \pmod 5\}$. Man kann sich den Petersen–Graph so vorstellen: $\langle E_i, K_i \rangle$ ist der sogenannte *Drudenfuß*, dem ein Fünfeck $\langle E_a, K_a \rangle$ umschrieben wird.

Zwei Graphen $G = \langle E, K \rangle$ und $H = \langle F, L \rangle$ heißen **isomorph**, falls es eine bijektive Abbildung $h : E \to F$ gibt, derart, dass für alle $u, v \in E$ gilt $uv \in K$ genau dann, wenn $h(u)h(v) \in L$. (Wir können dies etwas prägnanter ausdrücken. Es bezeichne $h[K] := \{\{h(u), h(v)\} : uv \in K\}$. Dann verlangen wir $h[K] = L$.) Es sind alle vollständigen Graphen mit gleicher Eckenzahl isomorph. Dies haben wir schon in der Schreibweise K_n zum Ausdruck gebracht; diese drückt lediglich eine Abhängigkeit von $n = |E|$ und nicht von E aus. Ebenso ist für ein vollständiger bipartiter Graph bis auf Isomorpie eindeutig durch die Mächtigkeit der Mengen S und T gegeben.

Definition 151 *Es sei $\mathfrak{G} = \langle E, K \rangle$ ein Graph und u eine Ecke von $\mathfrak{G}$. Dann heißt $N(u) := \{v : uv \in K\}$ die Menge der **Nachbarn** von u. Für $S \subseteq E$ sei $N(S) := \{v : uv \in K$ für ein $u \in S\}$. Ferner ist $d(u) := |N(u)|$ der **Grad** von u. Ist $d(u) = 0$, so heißt u **isoliert**. Die n–**Hülle** $\overline{N}^n(S)$ von $S \subseteq E$ sei nun induktiv wie folgt definiert.*

$$\begin{array}{lcl} \overline{N}^0(S) & := & S \\ \overline{N}^{k+1}(S) & := & \overline{N}(\overline{N}^k(S)) \end{array}$$

*$\overline{N}^n(S)$ ist die Menge aller Ecken, die von einer Ecke aus S in höchstens n Schritten erreichbar sind. $Z(S) := \bigcup_i \overline{N}^i(S)$ heißt die **Zusammenhangskomponente** von S. $\mathfrak{G}$ heißt **zusammenhängend**, falls $E = Z(\{x\})$ für ein $x \in E$ gilt.*

Es ist $N^k(S)$ die Menge aller Ecken, die von einer Ecke aus S in höchstens k Schritten erreichbar sind. Für endliche Graphen gilt: es existiert ein k dergestalt, dass $Z(S) = N^k(S)$ ist für alle k.

Definition 152 *Es sei $G = \langle E, K\rangle$ ein Graph und $S \subseteq E$. Dann sei $d(S)$ die kleinste Zahl derart, dass $E = N^d(S)$. Falls diese nicht existiert, so setze $d(S) := \infty$. Der* **Durchmesser** *von G ist*

$$d(G) := \max_{S \subseteq E \varnothing \neq S} d(S)$$

Falls G endlich, so ist $d(G)$ endlich genau dann, wenn G zusammenhängend ist. Dann ist sogar $d(G) < |E|$.

Satz 153 *In einem endlichen Graphen gilt stets*

$$\sum_{u \in E} d(u) = 2 \cdot |K| .$$

Beweis. Die linke Summe zählt alle Paare $\langle u, v\rangle$ derart, dass $uv \in K$. Da nun stets $u \neq v$ gilt, wenn $uv \in K$, so entspricht jeder Kante genau zwei Paaren. Q. E. D.

Satz 154 *In jedem Graphen ist die Anzahl der Ecken ungeraden Grades eine gerade Zahl.*

Beweis. Sei E_u die Menge der Ecken, für die $d(u)$ ungerade ist, und E_g die Menge aller Ecken, für die $d(u)$ gerade ist. Dann ist nach dem vorigen Satz $2 \cdot |K| = \sum_{u \in E} d(u) = \sum_{u \in E_u} d(u) + \sum_{u \in E_g} d(u)$. Es ist $\sum_{u \in E_g} d(u)$ stets gerade, also ist auch $\sum_{u \in E_u} d(u)$ gerade. Dann ist E_u gerade. Das war zu zeigen. Q. E. D.

Es sei nun als erstes ein Satz gezeigt, der gewissermaßen die erste Anwendung der Graphentheorie überhaupt darstellt, nämlich die Lösung des Königsberger Brückenproblems. In abstrakter Form dargestellt, lautet es so: *existiert eine Folge $F = k_1 k_2 \ldots k_r$ von Kanten aus K derart, dass jede Kante aus K genau einmal auftritt, und so dass k_i mit k_{i+1} für $1 \leq i < r$ und k_r mit k_1 jeweils inzident sind*? Eine solche Folge heißt ein **Euler–Zug**. Ferner verabreden wir, eine Folge $u_1 u_2 \ldots u_r$ von Ecken einen **Weg** der Länge $r - 1$ zu nennen, wenn $u_i u_{i+1} \in K$ für alle $1 \leq i < r$. Ein Weg ist **geschlossen**, falls $u_r = u_1$. Ist F ein Euler–Zug, so definiert dieser einen eindeutig bestimmten Weg. Denn seien $k_i k_{i+1}$ zwei aufeinanderfolgende Kanten aus F, dann hat $k_i \cap k_{i+1}$ genau ein Element. Es sei also u_{i+1} das eindeutig bestimmte v mit $v \in k_i \cap k_{i+1}$, und es sei u_1 das eindeutig bestimmte v mit $k_1 = vu_2$, u_{r+1} das eindeutig bestimmte v mit $k_r = u_r v$. Wir nennen einen Weg **nichtwiederholend**, falls $u_i = u_j$ nur dann gilt wenn $i = j$ oder $\{i, j\} = \{1, r\}$. Ein nichtwiederholender, geschlossener Weg heiße **Kreis**.

Satz 155 *In einem Graphen existiert ein Euler–Zug genau dann, wenn der Graph zusammenhängend ist und keine Ecke einen ungeraden Grad hat.*

Beweis. Die Bedingungen sind notwendig. Denn wenn $\mathfrak{G}$ nicht zusammenhängend ist, gibt es keine Ecke, von der aus alle Ecken erreichbar sind. Ferner: ist W ein Euler–Zug der Länge r, so ist u_i, $1 < i \leq r$, stets inzident mit $u_i u_{i+1}$ und $u_{i-1}u_i$ (und diese Kanten sind verschieden). Genauso ist u_{r+1} mit $u_1 u_{r+1}$ und $u_r u_{r+1}$ inzident. Jede Ecke ist also mit einer geraden Anzahl Kanten inzident. Nun also zur Umkehrung. Zunächst einmal zeigen wir: in einem Graphen, in dem jede Ecke geraden Grad hat, existiert ein Kreis durch eine gegebene Ecke. Sei u_1 also gegeben. Wir wählen eine beliebige zu u_1 adjazente Ecke u_2. Sei $\mathfrak{G}_1 := \langle E, K - \{\{u_1, u_2\}\}\rangle$. In $\mathfrak{G}_1$ haben u_1 und u_2 ungeraden Grad. Also existiert eine zu u_2 adjazente Ecke u_3. Es gilt $u_3 \neq u_1$ (sowie auch $u_3 \neq u_2$). Induktiv definieren wir eine Folge u_i, derart dass $u_1 u_2 \ldots u_i$ ein nichtwiederholender Weg ist. Ferner setzen wir

$$\mathfrak{G}_i := \langle E, K - \{\{u_i, u_{i+1}\} : 1 \leq j < i\}\rangle$$

Ist $u_i \neq u_1$, so hat u_i einen ungeraden Grad in $\mathfrak{G}_i$. (Denn wir haben aus $\mathfrak{G}$ genau eine mit u_i inzidente Kante entnommen.) Also existiert in $\mathfrak{G}_i$ eine zu u_i inzidente Ecke u_{i+1}. Ist $u_{i+1} = u_j$ für ein $j \leq i$, so gilt sicher $j < i - 1$. Also haben wir einen Kreis. Ist andererseits u_{i+1} verschieden von allen u_j, so ist $u_1 u_2 \ldots u_{i+1}$ ein nichtwiederholender Weg. Nun haben wir zwar noch nicht gezeigt, dass ein Kreis durch u_1 existiert, aber wir können aus K nun die Kantenmenge des eben konstruierten Kreises entfernen und erhalten so einen Graphen, in dem jede Ecke einen geraden Grad hat, und insbesondere ist der Grad von u_1 nicht Null ist (da wir keine mit u_1 inzidente Kante entnommen haben). Nun machen wir das gleiche Spiel nochmal mit dem neuen Graphen. Es ist klar, dass bei diesem Wegnehmen jede Kante einmal drankommt. Also ist u_1 in einem Kreis, da es in einer Kante ist. Dies zeigt unsere erste Behauptung. Der Satz folgt nun so. Wir wählen einen Kreis K_1 mit Kantenmenge γ_1. $W_1 := K_1$. Dann sei $\mathfrak{G}^1 := \langle E, K - \gamma_1\rangle$. Da wir zu jedem Punkt genau zwei inzidente Kanten entnehmen, hat in $\mathfrak{G}^1$ jede Ecke geraden Grad. Falls die Kantenmenge noch nicht leer ist, existiert eine Kante in $K - \gamma_1$. Es existiert sogar eine Kante $uv \notin \gamma_1$, derart, dass eine Ecke auf dem Kreis liegt, etwa u. Wir wählen nun in $\mathfrak{G}^1$ einen Kreis K_2 durch u. Nun definieren wir einen neuen Weg wie folgt: wir schieben bei u in K_1 den Kreis K_2 ein. Das Ergebnis ist ein geschlossener Weg W, auf dem keine Kante wiederholt wird. Sei γ_2 seine Kantenmenge. Wir setzen $\mathfrak{G}^2 := \langle E, K - \gamma_2\rangle$.

Wir fahren so fort. Es existiert eine Kante uv, die nicht in γ_2 liegt, aber mit einer Ecke aus W_2 inzident ist. Andernfalls ist $\mathfrak{G}$ nicht zusammenhängend. Q. E. D.

19. Teil: Graphen II. Färbungen.

In diesem Abschnitt wollen wir uns Färbungen von Graphen, insbesondere mit Färbungen von planaren Graphen befassen.

Definition 156 *Eine* ***Eckenfärbung*** *eines Graphen* $\langle E, K\rangle$ ***in*** Q *ist eine Funktion* $f : E \to Q$ *in eine Menge* Q *derart, dass für alle* $uv \in K$ *gilt:* $f(u) \neq f(v)$*. Die* ***chromatische Zahl von*** $\mathfrak{G}$ *ist die kleinste Zahl* n *für die ein* Q *mit Mächtigkeit* n *und eine Färbung von* $\mathfrak{G}$ *in* Q *existiert. Die chromatische Zahl von* $\mathfrak{G}$ *wird mit* $\chi(\mathfrak{G})$ *bezeichnet.*

Das klassische Färbungsproblem ist das sogenannte Vierfarbenproblem (welches inzwischen gelöst ist). Gegeben sei eine Landkarte. Länder sollen dabei wegzusammenhängend sein. (Dies bedeutet, dass man je zwei Punkte in einem Land durch einen Weg verbinden kann, der ganz in diesem Land varläuft. Es gibt also keine Exklaven.) Die Länder sind so zu färben, dass je zwei benachbarte Länder nicht dieselbe Farbe erhalten. Dabei gelten zwei Länder als benachbart, falls sie eine gemeinsame Grenze haben, die nicht aus isolierten Punkten besteht. Dies ist zunächst einmal kein graphentheoretisches Problem. Wir können es aber in ein äuivalentes graphentheoretisch Problem umformen. Es sei E die Menge der Länder und $uv \in K$, wenn u und v benachbart sind. Offensichtlich ist eine Landkartenfärbung exakt eine Färbung im Sinne der Definition. Bevor wir nun mit dem Vierfarbenproblem weitermachen, wollen wir uns ein Resultat über chromatische Zahlen beschaffen.

Definition 157 *Es sei* $\mathfrak{G} = \langle E, K\rangle$ *ein Graph und* $F \subseteq E$*. Setze* $L := K \cap \binom{F}{2}$*. Das Paar* $\langle F, L\rangle$ *heißt der von* $\mathfrak{G}$ *auf* F ***induzierte Graph****. Ist lediglich* $L \subseteq K \cap \binom{F}{2}$*, so heißt* $\langle F, L\rangle$ ***Teilgraph*** *von* $\mathfrak{G}$*.*

Ist zum Beispiel $\mathfrak{G}$ vollständig, so ist jeder induzierte Teilgraph auch vollständig.

Definition 158 *Eine* ***Clique*** *in einem Graphen ist eine maximale Menge von Ecken, auf denen der induzierte Graph vollständig ist. Eine* n*–****Clique in*** $\mathfrak{G}$ *ist eine Clique der Mächtigkeit* n*.*

Offensichtlich ist jede Ecke in einer Clique enthalten. Denn im schlimmsten Fall, wenn u isoliert ist, ist $\{u\}$ schon eine Clique. Es ist ferner so, dass Cliquen sich überschneiden können. In einem Kreis bilden je zwei benachbarte Punkte eine Clique, mit Ausnahme des Kreises aus drei Punkten, der selbst schon eine Clique aus drei Punkten bildet.

Lemma 159 *Es enthalte $\mathfrak{G}$ eine $\gamma + 1$–Clique. Dann ist $\mathfrak{G}$ nicht γ–färbbar.*

Satz 160 *Es sei $\mathfrak{G} = \langle E, K\rangle$ ein endlicher Graph und γ das Maximum aller $|d(u)|$, $u \in E$. Dann gilt $\gamma \leq \chi(\mathfrak{G}) \leq \gamma + 1$.*

Beweis. Nach dem vorigen Lemma muss $\gamma \leq \chi(\mathfrak{G})$ sein. Die andere Abschätzung muss noch gezeigt werden. Es sei $E = \{x_i : i < n\}$ für ein gewisses n. Wir wählen nun induktiv eine Farbe für die x_i. x_0 bekommt eine beliebige Farbe. Seien x_i, $i < j$, bereits gefärbt, so betrachten wir x_j. x_j hat höchstens γ viele Nachbarn. Nicht alle von ihnen sind bereits gefärbt. In jedem Fall gibt es aber eine Farbe f, welche kein Nachbar von x_j hat. Wir geben daher x_j die Farbe f. Offensichtlich lässt sich auf diese Weise der gesamte Graph färben. Q. E. D.

Der in dem Beweis verwendete Algorithmus ist ein Beispiel für einen sogenannten *Greedy* Algorithmus. Bei einem solchen Algorithmus wählt man immer die lokal beste Lösung, und dennoch wird am Ende eine global optimale Lösung erreicht. Ein weiteres Beispiel für einen Greedy–Algorithmus ist der Algorithmus zur Bestimmung eines Huffmann–Codes (siehe Teil 22). Nachdem wir die chromatische Zahl derart eingeschränkt haben, wollen wir uns dem Vierfarbenproblem zuwenden. Zunächst holen wir etwas weiter aus.

Sei $\mathfrak{G} = \langle E, K\rangle$. Ohne Beschränkung der Allgemeinheit ist $E = \{0, 1, \ldots, n-1\}$. Dann können wir $\mathfrak{G}$ die folgende **Inzidenzmatrix** $I(\mathfrak{G})$ zuordnen. Es ist $I(\mathfrak{G}) := (a_{ij})_{ij}$, wobei $a_{ij} = 1$ genau dann, wenn $\{i, j\} \in K$ und $a_{ij} = 0$ sonst. Die Inzidenzmatrix ist symmetrisch, das heißt, es ist $a_{ij} = a_{ji}$ für alle i, j. Jede $n \times n$–Matrix $A = (a_{ij})_{ij}$ ist eine Inzidenzmatrix eines Graphen über $\{0, 1, \ldots, n-1\}$, falls für alle i, j mit $i, j < n$ (i) $a_{ij} \in \{0, 1\}$, (ii) $a_{ii} = 0$, und (iii) $a_{ij} = a_{ji}$. Die Inzidenzmatrix eines vollständigen Graphen besteht nur aus Einsen, die Inzidenzmatrix eines Kreises ist eine Permutationsmatrix. Ferner ist die Inzidenzmatrix eines bipartiten Graphen bis auf Umbenennung der Elemente von der Form

$$\begin{pmatrix} \mathbf{0} & A \\ B & \mathbf{0} \end{pmatrix}$$

wobei **0** für die Nullmatrix entsprechender Größe steht, und A und B beliebige Matrizen sind. Im Falle des vollständigen bipartiten Graphen sind $A = \mathbf{1}$ und $B = \mathbf{1}$, wobei **1** wieder für die Matrix aus Einsen der entsprechenden Größe steht.

Mit Hilfe der Inzidenzmatrix kann man folgende interessante Beziehung festhalten.

Satz 161 *Es sei $I^k(\mathfrak{G}) = (c_{ij})_{ij}$ das k–fache Produkt von $I(\mathfrak{G})$ mit sich selbst, $k > 0$. Dann ist c_{ij} genau die Anzahl der Wege der Länge k von i nach j.*

Beweis. Induktion über k. Es ist $I^1(\mathfrak{G}) = I(\mathfrak{G}) = (a_{ij})_{ij}$, und ein Weg der Länge 1 genau eine Kante. Die Behauptung ist in diesem Falle also richtig. Nun sei $I^k(\mathfrak{G}) = (c_{ij})_{ij}$ und $I^{k+1}(\mathfrak{G}) = (d_{ij})_{ij}$. Dann ist $d_{ij} = \sum_{r=1}^{n} c_{ir} \cdot a_{rj}$. Die rechte Summe erstreckt sich über die Anzahl der Wege von i nach r, für die eine Kante von r nach j existiert, $1 \leq r \leq n$. Dies ist aber gerade die Anzahl der Wege der Länge $k + 1$ von i nach j. Q. E. D.

Es seien $\mathfrak{G} = \langle E, K\rangle$ ein Graph, F eine Menge und $h : E \to F$ eine surjektive Abbildung. Ferner sei für jedes $x \in F$ der von $h^{-1}(x)$ in $\mathfrak{G}$ induzierte Teilgraph zusammenhängend. (Dieser ist der Graph $\langle U_x, P_x\rangle$, wo $U_x := \{y \in E : h(y) = x\}$, $P_x = \{\{y, z\} \in \binom{U_x}{2} : \{x, y\} \in K\}$.) Nun setze $L := h[K] \cap \binom{F}{2}$. Das Paar $\langle F, L\rangle$ heißt dann eine **Kontraktion** von $\mathfrak{G}$. Es gilt nun folgender

Satz 162 *Es sei $\mathfrak{G}$ eine Kontraktion von $\mathfrak{H}$. Dann ist $\mathfrak{G}$ genau dann zusammenhängend, wenn $\mathfrak{H}$ zusammenhängend ist.*

Zum Beweis zeigen wir als erstes Folgendes. Eine Kontraktion heiße **elementar**, falls es ein $x \in F$ gibt mit $h^{-1}(x) \in K$ und für alle $y \neq x$ hat $h^{-1}(y)$ genau ein Element. Intuitiv gesprochen kontrahieren wir bei einer elementaren Kontraktion eine Kante. Es ist nicht schwer zu sehen, dass jede Kontraktion eine Verkettung von elementaren Kontraktionen ist. Also muss man obenstehenden Satz nur für elementare Kontraktionen zeigen. Dies ist nicht schwer.

Bevor wir dies tun, werden wir uns noch überlegen, dass man eine elementare Kontraktion auch etwas anders beschreiben kann. Es sei $\mathfrak{G} = \langle E, K\rangle$ gegeben und $uv \in K$. Das Resultat der Kontraktion von uv ist der folgende Graph $\mathfrak{H} = \langle E - \{v\}, L\rangle$, wobei

$$L := K \cap \binom{E - \{v\}}{2} \cup \{ux : x \in E - \{u, v\}, xv \in K\}$$

Das heißt, wir nehmen den auf $E - \{v\}$ induzierten Graphen und fügen alle Kanten ux hinzu, für die $xv \in K$.

Sei nun $\mathfrak{G}$ unzusammenhängend. Dann existieren Mengen A, B mit $A \cup B = E$, A und B disjunkt und $K \subseteq \binom{A}{2} \cup \binom{B}{2}$. Ist $uv \in K$, so ist ohne Beschränkung der Allgemeinheit $uv \in \binom{A}{2}$. Wir behaupten: ist $\mathfrak{H} = \langle E - \{v\}, L\rangle$ das Ergebnis der Kontraktion von uv, so ist jede Kante von $\mathfrak{H}$ entweder in $\binom{A-\{v\}}{2}$ oder in $\binom{B}{2}$. Sei nämlich $xy \in L$ eine Kante. Ist $x \in B$, so ist nach Definition von L $y \in B$. Also ist $L \subseteq \binom{A-\{v\}}{2} \cup \binom{B}{2}$, das heißt $\mathfrak{H}$ ist unzusammenhängend, wie versprochen. Sei umgekehrt $\mathfrak{H}$ unzusammenhängend, etwa $L \subseteq \binom{A}{2} \cup \binom{B}{2}$ für disjunkte Mengen A, B. Es sei $E = A \cup B \cup \{v\}$. Ferner existiert in $\mathfrak{G}$ eine Kante uv. Daher ist $u \in A$ oder $u \in B$. Ohne Beschränkung der Allgemeinheit $u \in A$. Wir setzen $A' := A \cup \{v\}$ und $B' := B$. Wir behaupten, dass $K \subseteq \binom{A'}{2} \cup \binom{B'}{2}$. Dazu sei $xy \in K$. Falls xy nicht mit uv inzident ist, so ist $xy \in \binom{A}{2}$ oder $xy \in \binom{B}{2}$ und wir sind fertig. Andernfalls ist $x = u$ oder $x = v$. (Der Fall $y = u$ oder $y = v$ ist ganz analog.) Sei $x = u$. Ist $y = v$, so ist ja $y \in A'$ und so $xy \in \binom{A'}{2}$. Ist $y \neq v$, so ist $xy \in L$; also $xy \in \binom{A}{2} \subseteq \binom{A'}{2}$. Sei nun $x = v$. Der Fall $y = v$ braucht nicht betrachtet zu werden. Sei also zusätzlich $y \neq v$. Dann ist $uy \in L$, also $y \in A'$. Daraus folgt $uy \in \binom{A}{2}$, und so $xy \in \binom{A'}{2}$. Daher ist $\mathfrak{G}$ unzusammenhängend.

Definition 163 *$\mathfrak{H}$ ist ein **Minor** von $\mathfrak{G}$, falls $\mathfrak{H}$ Kontraktion eines Teilgraphen von $\mathfrak{G}$ ist.*

Die Relation *ist Minor von* ist transitiv, wie aus dem folgenden Satz folgt.

Satz 164 *Ist $\mathfrak{I}$ Teilgraph oder Kontraktion von $\mathfrak{H}$ und ist $\mathfrak{H}$ Minor von $\mathfrak{G}$, so ist $\mathfrak{I}$ Minor von $\mathfrak{G}$.*

Beweis. Es sei $\mathfrak{G} = \langle E, K\rangle$ und $\mathfrak{H} = \langle P, Q\rangle$. Nach Voraussetzung existiert ein $F \subseteq E$ und eine Abbildung $h : F \twoheadrightarrow P$ derart, dass $Q \subseteq h[K] \cap \binom{F}{2}$. Nun sei $\mathfrak{I} = \langle R, T\rangle$ Kontraktum von $\mathfrak{H}$. Dann existiert $g : P \twoheadrightarrow R$ mit $T = g[Q]$. Dann ist $T = g[Q] \subseteq (g \circ h)[K] \cap \binom{F}{2}$, also $g \circ h$ eine Kontraktion eines Teilgraphen von $\mathfrak{G}$ auf $\mathfrak{I}$. Sei nun $\mathfrak{I}$ Teilgraph von $\mathfrak{H}$, also $R \subseteq P$ und $T = Q \cap \binom{R}{2}$. Wähle $F := h^{-1}[T]$ und $L := \{uv \in K : h(u)h(v) \in T\}$. Dann ist $\langle F, L\rangle$ nach Konstruktion ein Teilgraph von $\mathfrak{G}$. Ferner ist $h \restriction F : F \twoheadrightarrow R$ und $h[L] = R \cap \binom{R}{2}$. Q. E. D.

Ein Graph heißt **planar**, wenn er sich in der Ebene zeichnen lässt, ohne dass die Kanten sich überschneiden. Planare Graphen sind aus vielen

Gründen sehr interessant. Das Vier–Farben–Problem bezieht sich auf planare Graphen (Landkarten); aber auch in der Elektronik sind sie von immenser Wichtigkeit. Man denke nur an Schaltkreise. Die Leiterbahnen in Schaltkreisen entsprechen den Kanten eines Graphen, dessen Ecken gerade die Schaltelemente sind. Falls sich Kanten schneiden, so muss man auf der Platine eine sogenannte Brücke einbauen. Hochintegrierte Schaltkreise (VLSI) erlauben zwar auch Brücken, jedoch gilt es, so wenig wie möglich davon zu benützen, da sie schwierig zu realisieren sind — also teuer. Planare Graphen sind der Idealfall: man benötigt überhaupt keine Brücken. Wir teilen ohne Beweis den folgenden Sachverhalt mit.

Satz 165 (Kuratowski) *Genau dann ist $\mathfrak{G}$ planar, wenn weder $K_{3,3}$ noch der vollständige Graph K_5 ein Minor von $\mathfrak{G}$ sind.*

Wir können jedoch ein paar einfache Überlegungen machen. Eine **Realisierung** von $\mathfrak{G} = \langle E, K\rangle$ ist ein Paar $\langle p, \ell\rangle$, wo $p : E \to \mathbb{R}^2$ eine Abbildung der Ecken in die reelle Ebene, und für jedes $k = uv \in K$ $\ell(k) : [0, 1] \to \mathbb{R}^2$ eine stetige Abbildung mit $\ell(k)(0) = u$ und $\ell(k)(1) = v$ ist. Diese Realisierung ist **schnittfrei**, falls es keine Ecken k und k' mit $k \neq k'$ und keine r, r' mit $0 < r, r' < 1$ derart, dass $\ell(k)(r) = \ell(k')(r')$. $\mathfrak{G}$ ist nach Definition genau dann planar, wenn es eine schnittfreie Realisierung gibt. Es gilt nun dies.

Satz 166 *Es sei $\mathfrak{H}$ ein Minor von $\mathfrak{G}$. Ist dann $\mathfrak{G}$ planar, so ist es auch $\mathfrak{H}$.*

Beweis. Sei $\mathfrak{H} = \langle F, L\rangle$ ein Teilgraph von $\mathfrak{G} = \langle E, K\rangle$. Sei $\langle p, \ell\rangle$ eine Realisierung von $\mathfrak{G}$. Dann ist $\langle p \restriction F, \ell \restriction L\rangle$ eine Realisierung von $\mathfrak{H}$ und schnittfrei, falls $\langle p, \ell\rangle$ schnittfrei ist. Für Kontraktionen muss man sich etwas mehr Mühe geben. Da der Beweis relativ trickreich ist, werden wir ihn nicht ausführen. Q. E. D.

Im Übrigen ist jeder Graph schnittfrei im Raum realisierbar. Der Vierfarbensatz — früher als das *Vierfarbenproblem* bekannt, bevor es gelöst wurde — lautet nun wie folgt:

Satz 167 (Appelt & Haken) *Für jeden planaren Graphen $\mathfrak{G}$ ist $\chi(\mathfrak{G}) \leq 4$.*

Dazu man muss zeigen, dass jedem planaren Graphen eine Landkarte zugeordnet werden kann und jeder Landkarte ein planarer Graph. Das ist nicht schwer, wollen wir jedoch nicht tun. Aus dem Satz von Kuratowski folgt übrigens, dass in einem planaren Graphen keine Clique mehr als vier Elemente hat; daraus können wir immerhin zeigen, dass $\chi(\mathfrak{G}) \leq 5$. Da andererseits der vollständige K_4 planar ist, kann man die Schranke 4 nicht weiter verbessern. So blieb lediglich die Frage, ob 4 Farben nun ausreichen oder nicht, und dies war, im Gegensatz zu der Schranke 5, ein äußerst schweres Problem. Bewiesen wurde Satz 167 erst in den siebziger Jahren mit Hilfe eines Computers! Der Beweis ist (bis jetzt) sehr aufwändig.

20. Teil: Graphen III. Matchings.

Wir vereinbaren folgende Schreibweise. Es bezeichnet $\langle S + T, K\rangle$ einen bipartiten Graphen, wobei S und T die ausgezeichneten Mengen sind. ($S + T$ ist eine andere Schreibweise für $S \cup T$ in den Fall, wo S und T disjunkt sind.) Es sei im Folgenden der Einfachheit halber vereinbart, dass stets $|S| \leq |T|$.

Definition 168 *Es sei $\mathfrak{G} = \langle S+T, K\rangle$ ein bipartiter Graph. Ein* ***Matching*** *ist eine Menge $L \subseteq K$ derart, dass jede Ecke mit höchstens einer Kante aus L inzident ist.*

Alternativ dazu ist ein Matching ein Teilgraph $\langle S + T, L\rangle$, bei dem jede Ecke höchstens den Grad 1 hat. Die **Matching–Zahl** $m(\mathfrak{G})$ ist die maximale Anzahl aller Kanten eines möglichen Matchings von $\mathfrak{G}$. Ein Matching heißt ein **Maximum–Matching**, falls die Anzahl der Kanten genau $m(\mathfrak{G})$ ist. Wir fragen uns nun, wann die Matching–Zahl gleich $|S|$ ist. In diesem Fall nennen wir das Matching **ideal**. Ist dann $|S| = |T|$, so ist dann auch jedes Element von T mit jedem Element aus S gematcht. Es muss ideale Matchings nicht geben (zum Beispiel wenn $|S| \neq |T|$). Wann es sie gibt, dazu gibt es ein schönes Kriterium, bekannt auch unter dem Namen *Heiratssatz*. Wir betrachten dazu eine Kante als ein mögliches Heiratspaar. Bei der gegenwärtigen Rechtslage entsteht dann aus einer beliebigen Menge von Menschen ein bipartiter Graph. Ein Matching entspricht dann einer Auswahl aus den möglichen Heiratspaaren dergestalt, dass jeder einen Partner bekommt. In diesem Gewand ist vielleicht eher klar, warum es ideale Matchings nicht geben muss.

Sei $A \subseteq S$. Dann ist $N(A) \subseteq T$. $N(A)$ war definiert als $\{v : uv \in K \text{ für ein } u \in A\}$.

Satz 169 *Sei $\langle S+T, K\rangle$ ein bipartiter Graph. Genau dann ist $m(\mathfrak{G}) = |S|$, wenn $|A| \leq |N(A)|$ ist für alle $A \subseteq S$.*

Beweis. Zunächst einmal überlegen wir, dass die Bedingung notwendig ist. Dazu sei L ein Matching mit $|L| = |S|$. Dann existiert zu jedem $u \in S$ genau ein $v \in T$ mit $uv \in L$. Bezeichne v mit $f_L(u)$. Dann ist $f_L : S \to T$ eine injektive Abbildung nach Definition eines Matchings. Also ist, da ja $N(A) \supseteq f_L[A]$ und $|f_L[A]| = |A|$, $|N(A)| \geq |A|$. Nun zu der Behauptung, dass die Bedingung auch hinreicht. Sei ein Matching L gegeben mit $|L| < |S|$. Wir zeigen: es gibt ein Matching M mit $|M| > |L|$. Es existiert ein $u(0) \in S$ dergestalt, dass $u(0)$ zu keiner Kante aus L inzident ist. Immerhin existiert aber wegen $|N(\{u(0)\})| \geq 1$ ein $v(1) \in T$ mit $u(0)v(1) \in K$. Angenommen, $v(1)$ ist nicht gematcht in L. Dann ist $M := L \cup \{u(0)v(1)\}$ ein Matching mit $|M| > |L|$. Sei also $v(1)$ gematcht, etwa sei $u(1)v(1) \in L$. Wiederum ist wegen $|N(\{u(0), u(1)\})| \geq 2$ gesichert, dass ein $v(2) \neq v(1)$ existiert mit $u(1)v(2) \in K$ oder $u(0)v(2) \in K$. Ist $v(2)$ nicht gematcht, so hören wir auf, ansonsten besorgen wir uns ein $u(2)$ mit $u(2)v(2) \in L$. Wir können so fortfahren, bis wir eine nichtgematchte Ecke $v(r) \in T$ finden. Zu jedem $v(i)$ mit $1 \leq i \leq r$ existiert dann nach Konstruktion ein $u(j)$ mit $j < i$ und $u(j)v(i) \in K$. Wir haben also einen Weg $W = v(r), u(j_1), v(j_1), u(j_2), v(j_2), \ldots, u(j_p), v(j_p), u(0)$ mit $j_{p+1} := 0 < j_p < \ldots < j_2 < j_1 < r =: j_0$, wobei jeweils $u(j_a)v(j_a) \in L$ und $u(j_{a+1})v(j_a) \in K - L$. Es sei $Q := \{u(j_{a+1})v(j_a) : 0 \leq a \leq p\}$, $P := \{u(j_a)v(j_a) : 0 < a \leq p\}$. Dann ist $|P| = p < p + 1 = |Q|$. Setze nun $M := (L - P) \cup Q$. Dies ist ein Matching und $|M| > |L|$. Q. E. D.

Eine Variante des Matchingproblems ist das Auffinden einer sogenannten *Transversale.*

Definition 170 *Es sei $S := \{A_0, A_1, \ldots, A_{k-1}\}$ ein Mengensystem. Eine* ***Transversale*** *von S ist einen Menge $M = \{m_0, m_1, \ldots, m_{k-1}\}$ der Mächtigkeit k derart, dass für jedes $j < k$ gilt $m_j \in A_j$.*

Man beachte, dass keine Transversale existiert, wenn eine der Mengen leer ist. Dazu eine Interpretation. Wir nennen eine Funktion $f : \{0, 1, \ldots, k-1\} \to \bigcup S := \bigcup_{j<k} A_j$ eine **Auswahlfunktion**, falls $f(j) \in A_j$ für alle $j < k$. Solche Auswahlfunktionen gibt es immer. Wir wollen aber eine Auswahlfunktion haben, bei der $f(i) \neq f(j)$ ist für $i \neq j$. Damit haben wir jedem A_i eindeutig einen Repräsentanten $f(i)$ zugeordnet. Es ist damit allerdings nicht ausgeschlossen, dass $f(i) \in A_j$ für $i \neq j$ ist.

Satz 171 *Es sei $S = \{A_0, A_1, \ldots, A_{k-1}\}$ ein Mengensystem. Genau dann existiert eine Transversale von S, wenn für jedes $P \subseteq \{0, 1, \ldots, k-1\}$ gilt $|(\bigcup_{i \in P} A_i)| \geq |P|$.*

Beweis. Definiere einen bipartiten Graphen wie folgt. S sei wie gegeben, $T := \bigcup S$ und $K := \{\{A_i, x\} : x \in A_i\}$. Dann ist $\mathfrak{S} := \langle S{+}T, K\rangle$ bipartiter Graph. Eine Transversale S ist dann genau ein Matching von $\mathfrak{S}$ mit Mächtigkeit $|S|$. Dies existiert genau dann, wenn zu jedem $P \subseteq S$ gilt $|N(P)| \geq |P|$. Aber $|N(P)| = \{x : x \in \bigcup_{i\in P} A_i\}$. Q. E. D.

Ist zum Beispiel $A_0 = \varnothing$, so ist $|A_0| = 0 < 1$, also existiert laut dem Satz keine Transversale. Wir sind allerdings die Antwort schuldig geblieben, wie groß im allgemeinen Fall $m(\mathfrak{G})$ ist. Dazu definieren wir den Matching–Defekt eines Graphen.

Definition 172 *Es sei $\mathfrak{G} = \langle S{+}T, K\rangle$ ein bipartiter Graph. Der **Matching–Defekt** von $\mathfrak{G}$, $\delta(\mathfrak{G})$, ist das Maximum der Anzahlen $|A| - |N(A)|$ für solche A, wo $|A| \geq |N(A)|$ ist.*

Da $|\varnothing| \geq |N(\varnothing)| = 0$, ist der Matching–Defekt immer definiert und Null genau dann, wenn stets $|A| \leq |N(A)|$. Daher existiert ein Matching mit Mächtigkeit $|S|$ genau dann, wenn der Matching–Defekt Null ist.

Satz 173 *Es sei $\mathfrak{G} = \langle S + T, K\rangle$ ein bipartiter Graph. Dann ist $m(\mathfrak{G}) = |S| - \delta(\mathfrak{G})$.*

Beweis. Gewiss kann in einem Matching L nicht $|L| > |S| - \delta(\mathfrak{G})$ sein. Denn sei $A \subseteq S$ so gewählt, dass $|A| - |N(A)| = \delta(\mathfrak{G})$. Dann können höchstens $|A| - \delta(\mathfrak{G})$ Ecken aus A gematcht werden. Selbst wenn im günstigsten Fall alle Ecken aus $S - A$ gematcht sind, existieren $\delta(\mathfrak{G})$ ungematchte Ecken. Also $|L| \leq |A| - \delta(\mathfrak{G}) \leq |S| - \delta(\mathfrak{G})$. Wir zeigen nun, dass das Maximum erreicht wird. Dazu sei $T^* := T \cup D$, wobei D eine zu T disjunkte Menge der Mächtigkeit $\delta(\mathfrak{G})$ ist. Ferner sei $K^* := K \cup \{ud : u \in S, d \in D\}$, $\mathfrak{G}^* := \langle S \cup T^*, K^*\rangle$. $\mathfrak{G}^*$ ist ein bipartiter Graph. Es gilt $N^*(A) = N(A) \cup D$. Also ist der Matching–Defekt von $\mathfrak{G}^*$ gleich Null. Daher existiert ein Matching L^* mit $|L^*| = |S|$. Nun enthält L^* genau $\delta(\mathfrak{G})$ viele Kanten mit einer Ecke aus D. Sei also $L := L^* \cap \binom{S\cup T}{2}$. Dann ist L ein Matching mit $|L| = |L^*| - \delta(\mathfrak{G}) = |S| - \delta(\mathfrak{G})$, wie versprochen. Q. E. D.

Als Letztes noch eine andere Charakterisierung der Matching–Zahl. Wir nennen eine Menge $D \subseteq E$ einen **Träger** des Graphen, falls jede Kante aus $\mathfrak{G}$ mit einer Ecke aus D inzident ist. Wir können in einem bipartiten Graphen zum Beispiel $D = S$ wählen oder $D = T$. Daher können Träger durchaus kleiner sein als die Eckenmenge.

Satz 174 *In einem bipartiten Graphen $\mathfrak{G}$ gilt:*

$$min\{|D| : D \text{ Träger}\} = max\{|L| : L \text{ Matching}\}$$

Beweis. Setze $\tau(\mathfrak{G}) := min\{|D| : D \text{ Träger}\}$. Es ist $max\{|L| : L \text{ Matching}\} = |S| - \delta(\mathfrak{G}) = |S| - |A| + |N(A)|$ für ein gewisses $A \subseteq S$. Setze $D := (S - A) \cup N(A)$. D ist ein Träger von $\mathfrak{G}$. Denn sei $k \in K$, etwa $k = uv$ mit $u \in S$ und $v \in T$. Ist $u \notin A$, dann ist $u \in D$. Ist andererseits $u \in A$, dann ist $v \in N(A)$, also $v \in D$. Es gilt $|D| = |S| - |A| + |N(A)|$. (Beachte, dass A und $N(A)$ disjunkt sind.) Also ist $\tau(\mathfrak{G}) \leq m(\mathfrak{G})$. Aber $\tau(\mathfrak{G}) < m(\mathfrak{G})$ kann nicht gelten. Denn jeder Träger von $\mathfrak{G}$ ist auch Träger jeden Matchings von $\mathfrak{G}$. Ein Träger eines Matchings L hat aber mindestens $|L|$ Elemente. Q. E. D.

In dieser Formulierung hat dieser Satz die Form eines Minimum–Maximum–Prinzips. Solche Prinzipien sind häufig anzutreffen (zum Beispiel auch der Minimum Schnitt–Maximum Fluss–Satz in dem nächsten Kapitel).

Wir beschließen diesen Teil mit einem anschaulichen Beispiel eines bipartiten Graphen.

Definition 175 *Eine **affine Ebene** ist ein Paar $\langle P, G\rangle$, falls P eine Menge ist, die Menge von **Punkten**, und $G \subseteq \wp(P)$ die Menge der **Geraden**, und ferner gilt: (1) je zwei (verschiedene) Punkte liegen auf genau einer Geraden; (2) zu jeder Gerade g und jedem Punkt $Q \notin g$ existiert genau eine Gerade h mit $Q \in h$ und $g \cap h = \varnothing$; (3) je zwei (verschiedene) Geraden schneiden sich in höchstens einem Punkt; (4) eine Gerade enthält mindestens 3 Punkte; (5) es gibt mindestens drei Geraden. g ist **parallel** zu h, falls $g = h$ oder $g \cap h = \varnothing$.*

Wir können eine affine Ebene als bipartiten Graphen schreiben mit $S := P$, $T := G$, und $K := \{Qg : Q \in P, g \in T, Q \in g\}$. Für $Q \in S$ ist $N(Q)$ gerade das sogenannte **Geradenbündel** durch Q.

Satz 176 *Es sei $\langle P, G\rangle$ eine affine Ebene. Dann gilt für alle Punkte Q und alle Geraden g: $|N(Q)| = |g| + 1$. Ferner gilt für alle Punkte Q, Q', dass $|N(Q)| = |N(Q')|$, und es gilt für je zwei Geraden g, g', dass $|g| = |g'|$.*

Beweis. Die erste Behauptung zeigt die anderen. Denn es ist für beliebige Punkte Q, Q' und beliebige Geraden g und g', dass $|N(Q)| = |g| + 1 = |N(Q')| = |g'| + 1$. Also $|N(Q)| = |N(Q')|$ sowie $|g| = |g'|$. Es bezeichne $\overline{QR}$ die Gerade, welche sowohl Q als auch R enthält. Sei $Q \notin g$. Es existiert nun genau eine Gerade $h \in N(Q)$ und $h \cap g = \varnothing$. Dann ist $R \mapsto \overline{QR}$

eine Bijektion zwischen g und $N(Q) - \{h\}$. Nun benötigen wir noch eine Bijektion zwischen g und $N(Q) - \{h\}$ in dem Fall, wo $Q \in g$. In diesem Fall wählen wir $h = g$. Nun wählen einen Punkt $T \notin g$ und nennen f die durch T gehende, zu g parallele Gerade. Ferner sei U ein Punkt nicht auf g oder h und k eine Gerade durch U. Man verschalte nun die Bijektionen $N(Q) - \{g\} \to h \to N(U) - \{k\} \to g$. Q. E. D.

21. Teil: Graphen IV. Netzwerke und Flüsse.

Definition 177 *Ein Paar $\langle E, K\rangle$ heißt **gerichteter Graph**, falls E eine nichtleere Menge (die Menge der Ecken) und $K \subseteq E \times E$ eine Menge von sogenannten **gerichteten Kanten** ist. Ist $k \in K$ und $k = \langle u, v\rangle$, so heißt $k^- := u$ die **Anfangsecke** und $k^+ := v$ die **Endecke** von k.*

Zu einer Ecke $u \in E$ setze $N^-(u) := \{v : \langle u, v\rangle \in K\}$ und $N^+(u) := \{v : \langle v, u\rangle \in K\}$. Ebenso ist $N^-(S)$, $N^+(S)$ für $S \subseteq E$ definiert. Wir definieren den **(Vorwärts–)Transit** von S als die kleinste Menge T mit $S \subseteq T$ und $T \subseteq N^-(T)$. Analog ist der Rückwärtstransit definiert. Der **Ingrad** von u ist $d^+(u) := |N^+(u)|$ und der **Ausgrad** $d^-(u) := |N^-(u)|$. Ist der Ingrad von u Null, so heißt u eine **Quelle**; ist der Ausgrad von u gleich Null, so heißt u eine **Senke**.

Satz 178 *In einem gerichteten Graphen gilt*

$$\sum_{u\in E} d^+(u) = \sum_{u\in E} d^-(u) = |K| \ .$$

Der Beweis ist einfach. Ohne Beschränkung der Allgemeinheit ist $E = n$. Falls nicht zugleich $\langle u, v\rangle \in K$ und $\langle v, u\rangle \in K$ für verschiedene u, v sowie $\langle u, u\rangle \notin K$ für alle $u \in K$, so können wir für $\mathfrak{G}$ eine $n \times n$–Inzidenzmatrix $J(\mathfrak{G}) = (b_{ij})_{ij}$ wie folgt definieren:

$$b_{ij} := \begin{cases} 1 & \text{falls } \langle i, j\rangle \in K, \\ -1 & \text{falls } \langle j, i\rangle \in K, \\ 0 & \text{falls } \langle i, j\rangle, \langle j, i\rangle \notin K. \end{cases}$$

Es ist nicht schwer zu sehen, dass $\sum_{i<n} b_{ij} = 0$ sowie $\sum_{i<n} b_{ji} = 0$ für alle i gilt sowie $J(\mathfrak{G})^T = -J(\mathfrak{G})$. Hierbei ist $J(\mathfrak{G})^T := (b_{ji})_{ij}$. Wir machen jedoch im Folgenden keine Einschränkungen an $\mathfrak{G}$.

Definition 179 *Ein* ***Netzwerk über*** $\mathbb{R}$ *ist ein Quadrupel* $\langle\langle E, K\rangle, p, q, \gamma\rangle$, *wo* $\langle E, K\rangle$ *ein gerichteter Graph ist,* $p, q \in E$ *sowie* $\gamma : K \to \mathbb{R}$ *eine Funktion mit* $\gamma(k) \geq 0$ *für alle* $k \in K$. p *heißt die* ***Quelle*** *und* q *die* ***Senke*** *des Netzwerks und* γ *die* ***Kapazität***.

Analog wird ein *Netzwerk über* $\mathbb{N}_0$ definiert. Die Idee hinter dieser Definition ist diese. In einem System von Kanälen (Leitungen, Straßen) hat jeder Weg seine charakteristische Kapazität (Leitfähigkeit). Wir wollen uns damit beschäftigen, wie man Flüsse in einem solchen System beschreiben kann und werden seine maximale Auslastung bestimmen. Dies ist bei Transportproblemen aber auch beim Auslegen von Schaltkreisen eine sehr wichtige Angelegenheit. Ein **Fluss** ist eine Funktion $f : K \to \mathbb{R}$ mit $f(k) \geq 0$ für alle $k \in K$. Gegeben f, definiere ∂f durch

$$(\partial f)(u) := \sum_{k^+=u} f(k) - \sum_{k^-=u} f(k) .$$

Die Funktion ∂f misst in jedem Punkt die Differenz zwischen dem Einfluss und Ausfluss. Dies ist der sogenannte **Nettofluss**.

Definition 180 *Ein Fluss in einem Netzwerk* $\langle E, K\rangle$ *mit Quelle* p, *Senke* q *und Kapazität* γ *heißt* ***zulässig***, *falls*

1. $0 \leq f(k) \leq \gamma(k)$ *für alle* $k \in K$ *und*

2. $(\partial f)(u) = 0$ *für alle* $u \in E - \{p, q\}$.

Gegeben ein zulässiger Fluss können wir uns fragen, wie hoch der Gesamtdurchfluss ist. Dies ist gleich dem gesamten Fluss aus p, der Quelle, oder genau gleich dem gesamten Fluss in die Senke q. Denn es ist

$$0 = \sum_{x\in E} (\partial f)(x) = (\partial f)(p) + (\partial f)(q)$$

Daher ist also $(\partial f)(q) = -(\partial f)(p)$. Wir nennen $w(f) := -(\partial f)(p)$ den **Wert** des Flusses f. Die Frage, die sich stellt, ist nun, wie groß der Wert eines Flusses sein kann.

Definition 181 *Es sei ein Netzwerk gegeben aus* $\langle E, K\rangle$, *der Quelle* p, *der Senke* q *und der Kapazität* γ. *Es sei* $E = X \cup Y$ *mit* $X \cap Y = \varnothing$, $p \in X$

und $q \in Y$. Das Paar $\langle X, Y\rangle$ heißt dann ein ***Schnitt****. Die* ***Kapazität*** *des Schnittes, $c(X,Y)$, wird errechnet durch*

$$c(X,Y) := \sum_{k^- \in X, k^+ \in Y} \gamma(k)$$

Satz 182 *Es sei ein Netzwerk N gegeben. Ferner sei f ein zulässiger Fluss für N und $\langle X, Y\rangle$ ein Schnitt von N. Dann gilt*

$$w(f) \leq c(X,Y) \; .$$

Beweis. Für eine Partition $\langle A, B\rangle$ sei $S(A,B) := \{k \in K : k^- \in A, k^+ \in B\}$. Es gilt nun

$$w(f) = (\partial f)(q) = \sum_{u \in Y} (\partial f)(u) \; .$$

Es ist ja $(\partial f)(u)$ die Differenz zwischen $\sum_{k^+ \in u} f(k)$ und $\sum_{k^- \in u} f(k)$. Wir können also in $(\partial f)(u)$ zwei Sorten Kanten entdecken: solche, deren beide Ecken in Y liegen, und solche, deren eine Ecke in X und deren andere Ecke in Y liegt. Der Beitrag einer Kante der ersten Sorte taucht in der obenstehenden Summe einmal positiv und einmal negativ auf, und so kann man die Beiträge dieser Kanten weglassen. Also ist

$$w(f) = \sum_{k \in S(X,Y)} f(k) - \sum_{k \in S(Y,X)} f(k) \leq \sum_{k \in S(X,Y)} f(k) \leq \sum_{k \in S(X,Y)} \gamma(k) = c(X,Y)$$

Dies zeigt die Behauptung. Q. E. D.

Ohne Beweis stellen wir folgenden Satz vor.

Satz 183 *Es sei N ein Netzwerk. Dann gilt*

$$min\{c(X,Y) : \langle X, Y\rangle \text{ Schnitt}\} = max\{w(f) : f \text{ zulässiger Fluss}\}$$

Dieser Satz gilt im Übrigen auch für Flüsse über $\mathbb{N}_0$. In diesem Fall existiert ein maximaler Fluss mit $f(k) \in \mathbb{N}_0$. Wir leiten daraus den Satz 174 her. Sei ein bipartiter Graph $\mathfrak{G} := \langle S + T, K\rangle$ gegeben. Seien p und q neue Elemente. Dann setzen wir $\mathfrak{G}^* := \langle S + T + \{p,q\}, K^*\rangle$, wobei $K^* := K \cup \{\langle p, u\rangle : u \in S\} \cup \{\langle v, q\rangle : v \in T\}$. Schließlich sei $\gamma(k) := 1$ für jede Kante. Dann ist $\mathfrak{G}^*$ mit p als Quelle und q als Senke und Kapazität γ ein Netzwerk über $\mathbb{N}_0$. Ein zulässiger Fluss in diesem Netzwerk ist dann immer eine Funktion $f : K^* \to \{0,1\}$. Da für eine Ecke $u \in S$ stets $(\partial f)(u) = 0$ ist, so gibt es höchstens ein $v \in T$ mit $\langle u, v\rangle \in K^*$ (und somit $\langle u, v\rangle \in K$), sodass $f(\langle u, v\rangle) = 1$. Wir bekommen also ein Matching $M(f)$ für diesen Fluss. Sei umgekehrt ein Matching L gegeben. Setze $f_L(\langle u, v\rangle) := 1$ genau dann, wenn $uv \in L$ (es sei also $f_L(\langle u, v\rangle) := 0$ falls $uv \notin L$); $f_L(\langle p, u\rangle) := 1$ genau dann, wenn u gematcht ist und $f_L(\langle v, q\rangle) := 1$ genau dann, wenn v gematcht ist. Dies definiert einen zulässigen Fluss, wie man leicht sieht. Sein Wert ist genau $|L|$. Diese Zuordnung zwischen zulässigen Flüssen und Matchings ist bijektiv, und wir haben $|L| \leq |L'|$ genau dann, wenn $f_L(k) \leq f_{L'}(k)$ für alle $k \in K^*$ (genau dann, wenn $f_L(k) \leq f_{L'}(k)$ für alle $k \in K$). Es folgt, dass ein Matching genau dann maximal ist, wenn der zugeordnete Fluss ein maximaler zulässiger Fluss ist. Wir haben jetzt also

$$(\dagger) \qquad \mathit{max}\{w(f) : f \textit{ Fluss in } \mathfrak{G}^*\} = \mathit{max}\{|L| : L \textit{ Matching in } \mathfrak{G}\}\ .$$

Um nun abzuleiten, dass die minimale Kapazität von Schnitten in $\mathfrak{G}^*$ gleich der minimalen Größe von Trägern von $\mathfrak{G}$ ist, müssen wir etwas genauer hinsehen. Sei dazu $A \subseteq S$. Wir setzen dann $\sigma(A) := \langle X(A), Y(A)\rangle$, wobei $X(A) := \{p\} \cup A \cup N(A)$ ist und $Y(A) := \{q\} \cup (S - A) \cup (T - N(A))$. Dies ist sicher ein Schnitt. Berechnen wir seine Kapazität. Die Kapazität des Schnitts ist die Anzahl der Kanten von $X(A)$ nach $Y(A)$. Man sieht sehr leicht, dass es genau $|S| - |A| + |N(A)|$ Kanten gibt. Nun definieren wir auch einen Träger von $\mathfrak{G}$, nämlich $\tau(A) := (S - A) \cup N(A)$. Dass dies ein Träger ist, sieht man so. Eine Kante in $\mathfrak{G}$ ist entweder (a) in $A \times N(A)$ oder (b) $(S - A) \times T$. In beiden Fällen ist sie mit einer Ecke aus $\tau(A)$ inzident. Dieser Träger hat genauso viele Elemente, wie die Kapazität von $\sigma(A)$ angibt.

Lemma 184 *Sei $A \subseteq S$. Dann gilt $c(\sigma(A)) = |\tau(A)| = |S| - |A| + |N(A)|$.*

Es ist nun so, dass nicht alle Schnitte von der Form $\sigma(A)$ sind und nicht alle Träger von der Form $\tau(A)$. Wir müssen also zeigen, dass wenigstens ein minimaler Schnitt bzw. Träger diese Form hat.

Lemma 185 *Sei U ein minimaler Träger von $\mathfrak{G}$. Dann hat U die Form $\tau(A)$ für ein $A \subseteq S$.*

Beweis. Sei U ein minimaler Träger. Setze $A := S - U$. Wir zeigen $U = \tau(A)$. Dann ist $S \cap U = S - A$. Zu zeigen bleibt, dass $U \cap T = N(A)$. Da U Träger ist, muss sicher $U \cap T \supseteq N(A)$ sein. Denn jede Kante, die mit einem Punkt aus A inzident ist, muss mit einem Punkt aus U inzident sein, und dieser kann nur aus $N(A)$ stammen. Sicher gibt es für jedes $u \in N(A)$ eine solche

Kante, und daraus folgt $U \cap T \supseteq N(A)$. Daher gilt jetzt: $U \supseteq \tau(A)$. Falls also U minimal ist, gilt sogar $U = \tau(A)$. Q. E. D.

Lemma 186 *Es sei $\langle X^*, Y^* \rangle$ ein Schnitt von $\mathfrak{G}^*$ mit maximaler Kapazität. Dann ist $\langle X^*, Y^* \rangle = \sigma(A)$ für ein $A \subseteq S$.*

Beweis. Sei $\langle X^*, Y^* \rangle$ ein Schnitt von $\mathfrak{G}^*$ mit minimaler Kapazität. Dann setze $A := X^* \cap S$. $\sigma(A) = \langle X^*, Y^* \rangle$. Nun ist $X^* = S \cap A$. Ist $\langle X^*, Y^* \rangle \neq \sigma(A)$, so muss also entweder (a) oder (b) gelten. (a) Es existiert ein Punkt $y \in Y^* \cap T$, der auch in $N(A)$ ist. (b) Es existiert ein Punkt $x \in X^* \cap T$, der nicht in $N(A)$ ist. Fall (a) tritt nicht ein. Denn setze andernfalls $X^{**} := X^* \cup \{y\}$, $Y^{**} := Y^* - \{y\}$. Dann hat $\langle X^{**}, Y^{**} \rangle$ geringere Kapazität als $\langle X^*, Y^* \rangle$ im Widerspruch zur Annahme. Fall (b) tritt auch nicht ein. Denn wenn $x \in (X^* \cap T) - N(A)$, so sei $\langle X^{**}, Y^{**} \rangle$ definiert durch $X^{**} := X^* - \{x\}$ und $Y^{**} := Y^* \cup \{x\}$ ein Schnitt mit geringerer Kapazität. Dies beendet den Beweis. Q. E. D.

Aus Lemma 184 folgt jetzt mit Hilfe von Lemma 185 und 186

$$(\ddagger) \qquad \mathit{min}\{c(X,Y) : \langle X, Y \rangle \textit{ Schnitt in } \mathfrak{G}^*\} = \mathit{min}\{|D| : D \textit{ Träger von } \mathfrak{G}\} .$$

Wegen Satz 183 gilt

$$\mathit{max}\{w(f) : f \textit{ Fluss in } \mathfrak{G}^*\} = \mathit{min}\{c(X,Y) : \langle X, Y \rangle \textit{ Schnitt von } \mathfrak{G}^*\}$$

und deswegen, zusammen mit (†) und (‡):

$$\mathit{max}\{|L| : L \textit{ Matching in } \mathfrak{G}\} = \mathit{min}\{|T| : T \textit{ Träger von } \mathfrak{G}\} .$$

Und das ist die Aussage von Satz 174. Ferner erhalten wir aus Lemma 184 die Aussage, dass

$$\mathit{min}\{|T| : T \textit{ Träger von } \mathfrak{G}\} = |S| - \delta(\mathfrak{G}) .$$

22. Teil: Codierung I. Datenkompression.

Das Material zu den folgenden drei Teilen stammt aus R.–H. Schulz: *Codierungstheorie*, Vieweg Verlag, 1991. Vordergründig geht es bei der Codierung darum, eine Nachricht (**Quelltext**) in einer gewissen Form umzuschreiben (also nicht, ihn in eine andere Sprache zu übersetzen); dabei erhält man den **Zieltext**. Dabei kann dies aus mehreren Gründen geschehen; erstens, weil der Quelltext in seiner gegebenen Form nicht aufgenommen werden kann (zum Beispiel, weil wie im Computer das gewöhnliche Alphabet nicht vorhanden ist), zweitens, weil der Quelltext gekürzt werden kann (man sagt, er sei *redundant*), drittens, weil bei der Übertragung Fehler auftreten können, und schließlich, weil ein Fremder den Text lesen könnte und man das nicht immer will. Je nach Umstand und je nach dem, was man vermeiden möchte, sind ganz verschiedene Techniken der Codierung gefragt.

Definition 187 *Ein* ***Code*** *mit* ***Quellalphabet*** A *und* ***Zielalphabet*** B *ist ein Paar* $\langle\varphi, \psi\rangle$ *von berechenbaren Funktionen mit* $\varphi : A^* \to B^*$ *und* $\psi : B^* \to A^* \cup \{\sharp\}$ *derart, dass (1)* $\psi \circ \varphi(\vec{x}) = \vec{x}$ *für alle* $\vec{x} \in A^*$*, (2)* $\psi(\vec{y}) = \sharp$ *für alle* $\vec{y}$*, welche nicht von der Form* $\varphi(\vec{x})$ *sind.* φ *heißt die* ***Verschlüsselungsvorschrift*** *und* ψ *die* ***Entschlüsselungsvorschrift****.* $\vec{x} \in B^*$ *heißt ein* ***Codewort****, falls ein* $\vec{y} \in A^*$ *existiert mit* $\varphi(\vec{y}) = \vec{x}$*.*

Haben wir einen Code $\langle\varphi, \psi\rangle$, so sagen wir, dass wir Zeichenreihen über A durch Zeichenreihen über B codieren. Es ist klar, dass φ injektiv und ψ surjektiv (auf A^*) sein muss. Die Definition verlangt nicht, dass das Codieren oder Decodieren einfach sein muss. In der Tat sind gerade die Methoden zur Datenkompression oder auch die fehlerkorrigierenden Codes nicht ganz einfach zu handhaben (wenn auch algorithmisch nicht besonders schwierig). Da es jedoch nicht einfach ist zu sagen, was eine leichte und was eine schwere Vorschrift ist, haben wir dies aus dem Spiel gelassen. Die einfachste Rechenvorschrift ist zweifellos das zeichenweise Übersetzen. Wie schon in Teil 5 angedeutet, kann man Alphabete ineinander übersetzen, wenn sie nur mindestens zwei Zeichen enthalten. Seien nämlich A, B endliche Alphabete und $v : A \to B^*$. Dann existiert genau ein Homomorphismus $\overline{v} : A^* \to B^*$. Wann kann man nun $\overline{v}$ umkehren?

Definition 188 *Wir sagen, eine Abbildung* $v : A \to B^*$ *habe die* ***Präfixeigenschaft****, falls (a) für alle* $a \in A$*:* $v(a) \neq \varepsilon$ *und (b) für alle* $a, a' \in A$ *mit* $a \neq a'$ *gilt:* $v(a)$ *ist nicht Präfix von* $v(a')$*. ((a) benötigt man im Falle, dass* $|A| = 1$*.)*

Satz 189 *Es sei* $v : A \to B^*$ *mit Präfixeigenschaft. Dann existiert eine berechenbare Umkehrabbildung* $\psi : B^* \to A^* \cup \{\sharp\}$ *von* $\overline{v}$*.*

Beweis. $\overline{v}$ ist injektiv. Denn sei $\vec{x} = x_0 x_1 \ldots x_{m-1} \in A^*$ und $\vec{y} = y_0 y_1 \ldots y_{n-1} \in A^*$ sowie $\overline{v}(\vec{x}) = \overline{v}(\vec{y})$. Wir zeigen induktiv, dass $m = n$ und $x_i = y_i$ für alle $i < n$. ($m = 0$.) Zunächst ist ε der Code nur von ε, da $v(a) \neq \varepsilon$ ist für alle $a \in A$. Sei die Behauptung für m gezeigt. Wir zeigen sie für $m + 1$. Es ist $v(x_0)$ ein Präfix von $\overline{v}(\vec{y})$ und so entweder $v(x_0)$ Präfix von $v(y_0)$ oder $v(y_0)$ Präfix von $v(x_0)$. Daher $x_0 = y_0$ und so $\overline{v}(x_1 x_2 \ldots x_m) = \overline{v}(y_1 y_2 \ldots y_n)$. Ferner haben die beiden Worte kleinere Länge als die Ausgangsworte. Nach Induktionsannahme ist $m = n$ und $x_i = y_i$ für alle $0 < i < m+1$. Also $\vec{x} = \vec{y}$.

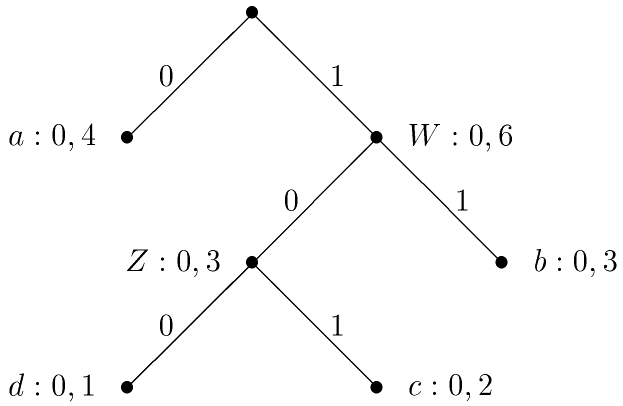
Wie sieht nun die Rückübersetzung aus? Das Verfahren ist denkbar einfach. Wir betrachten $\vec{y} \in B^*$. Sei $\vec{y} = y_0 y_1 \ldots y_{n-1}$. Wir betrachten dasjenige i mit $y_0 y_1 \ldots y_{i-1} = v(a)$ für ein $a \in A$ — sofern dieses existiert. Existiert es nicht, sei $\psi(\vec{y}) := \sharp$. Existiert es, so sei $x_0 := a$ und $\vec{y}^1 := y_i y_{i+1} \ldots y_{n-1}$. Fahre mit $\vec{y}^1$ fort und berechne so x_1, $\vec{y}^2$ und so weiter. Das Verfahren endet entweder damit, dass alle Zeichen gelesen sind und eine Zeichenkette $x_0 x_1 \ldots x_{k-1}$ konstruiert ist, oder dass $\sharp$ ausgegeben wird. Im ersten Fall sieht man sofort, dass $\overline{v}(\vec{x}) = \vec{y}$. Im zweiten Falle existiert keine Zeichenkette in A^*, deren Bild $\vec{y}$ ist. Q. E. D.

Die billigste Variante, die Präfixeigenschaft sicherzustellen, ist die Codierung durch gleichlange Blöcke im Zielalphabet. Ein Beispiel ist der ASCII–Code. Das Ausgangsalphabet besteht aus 128 Zeichen, das Zielalphabet aus nur zwei, 0 und 1. Jedem Zeichen aus A ordnen wir eine Folge über $\{0, 1\}$ der Länge 7 zu. Diese Zuordnung ist bijektiv. (In Wirklichkeit besteht der Code aus 8 Zeichen, aber nur 7 werden gebraucht; das achte Zeichen ist das sogenannte *Prüfbit*. Damit werden wir uns noch befassen.)

Diese Verfahren ist relativ einfach zu handhaben, hat aber einen Nachteil: die übersetzten Texte sind oft länger als nötig. Dies ist immer dann relevant, wenn man Texte abspeichern möchte und Platz knapp ist. Um das genauer darzustellen, überlegen wir uns, dass die Zeichen unseres Alphabets nicht mit der gleichen Häufigkeit auftreten. Zum Beispiel ist der Buchstabe 'e' der bei weitem häufigste Buchstabe. Es ist also plausibel, dass ein Code, der 'e' ein relative kurzes Wort über B zuweist, im Durchschnitt kürzere Texte erzeugt als ein anderer Code, der dies nicht tut. Wir nehmen nun an, wir hätten für jedes Zeichen aus A die Wahrscheinlichkeit $p(a)$ seiner Auftretens in einem beliebigen Textes ermittelt. Die beste Codierung im Falle, dass $B = \{0, 1\}$, ist die Codierung von *Huffman*. Man geht so vor. Seien a_0 und a_1 Buchstaben mit der kleinsten Wahrscheinlichkeit. (Diese müssen nicht eindeutig sein. Wir verlangen nur, dass für einen beliebigen anderen Buchstaben die Häufigkeit des Auftretens nicht kleiner ist als die von a_0 oder a_1.) Ist $A = \{a_0, a_1\}$, so setze $v(a_0) := 0$ und $v(a_1) := 1$. Ansonsten sei $v(a_0) := \vec{x}0$ und $v(a_1) := \vec{x}1$ für ein noch zu bestimmendes $\vec{x}$. Ersetze A durch $A^1 := (A - \{a_0, a_1\}) \cup \{Z\}$, wobei Z ein neues Zeichen ist, das heißt, $Z \notin A$. Ferner sei $p^1(b) := p(b)$, falls $b \in A - \{a_0, a_1\}$ und $p^1(Z) := p(a_0) + p(a_1)$. Verfahre nun mit A^1 und p^1 wie mit A und p. Dies ergibt eine Vorschrift $v^1 : A^1 \to \{0,1\}^*$. Setze $\vec{x} := v^1(Z)$.

Ein Beispiel. Es sei $A = \{a, b, c, d\}$ und $p(a) = 0,4$, $p(b) = 0,3$, $p(c) = 0,2$ und $p(d) = 0,1$.

a	b	c	d
$0,4$	$0,3$	$0,2$	$0,1$



Im ersten Schritt finden wir $v(d) = \vec{x}0$ und $v(c) = \vec{x}1$. $A^1 := \{a, b, Z\}$ mit $p^1(Z) = 0,3$. Nun sind b und Z die Zeichen mit geringster Auftretenswahrscheinlichkeit. Also ist $v^1(Z) = \vec{y}0$ und $v^1(b) = \vec{y}1$. Nun sei $A^2 := \{a, W\}$, $p^2(W) = 0,6$. Setze nun $v^2(a) := 0$, $v^2(W) := 1$. Es ist dann $v^1(Z) = v^2(W)0 = 10$, $v^1(b) = v^2(W)1 = 11$ und schließlich $v(c) = v^1(Z)1 = 101$, $v^1(Z)0 = 100$.

$$\begin{array}{lcl} a & \mapsto & 0 \\ b & \mapsto & 11 \\ c & \mapsto & 101 \\ d & \mapsto & 100 \end{array}$$

Es sollte klar sein, dass dieses Verfahren nicht eindeutig ist. Oft hat man im Verlaufe der Berechnung eine Wahl, einerseits in der Zuweisung von 0 und 1, andererseits, wenn zwei gleiche Wahrscheinlichkeiten auftreten. Der Algorithmus zur Erstellung eines Huffmann–Codes ist also greedy: die in jedem Schritt günstigste Lösung sichert das optimale Ergebnis. Wir teilen ohne Beweis mit, dass die Huffman–Codierung optimal in dem Sinne ist, dass die Zieltexte im Durchschnitt am kürzesten sind. Dies gilt allerdings nur, wenn die Auftretenswahrscheinlichkeit der Buchstaben im Quelltext wie angegeben ist und die Quelle, wie man sagt, kein Gedächtnis hat. Die Huffman–Codierung muss also nicht die beste Codierung überhaupt sein (falls es so etwas gibt). Eine Möglichkeit der Verbesserung ist die sogenannte *Blockcodierung*. Wir fügen zu A ein Symbol $\diamondsuit$ hinzu. Es sei ferner n eine natürliche Zahl, mindestens 1. Ein Quelltext wird von links nach rechts in Blöcke der Länge n geteilt. Geht dies nicht ganz auf (das heißt, seine Länge ist kein Vielfaches von n), so füllen wir den Quelltext am Ende mit $\diamondsuit$ auf. (Nur dafür müssen wir $\diamondsuit$ einführen. Da in der normalen Sprache ein Symbol existiert, das man ohne Schaden hinzufügen kann — nämlich das Leerzeichen —, entfällt die Notwendigkeit eines gesonderten Zeichens $\diamondsuit$.) Nun übersetzen wir nach diesem Verfahren anstelle von Buchstaben aus A, Blöcke über $A \cup \{\diamondsuit\}$ der Länge n. Hat A k Zeichen, so haben wir insgesamt $(k+1)^n$ Blöcke zu übersetzen. Der Witz ist nun, dass wir dies wieder mit Hilfe der Huffman–Codierung tun können. Warum sparen wir dabei? Es stellt sich heraus, dass die Häufigkeit des Auftretens eines Einzelzeichens auch von dem Kontext abhängt, in dem es auftritt. Zum Beispiel ist die Wahrscheinlichkeit, dass nach 'e' erneut 'e' auftritt, sehr klein, obwohl 'e' ja sehr häufig ist. Wir sprechen in diesem Zusammenhang davon, die Quelle *habe ein Gedächtnis*.

Betrachten wir erneut unser Alphabet $A = \{a, b, c, d\}$. Sei p wie folgt

aa	ab	ac	ad	ba	bb	bc	bd
$0,16$	$0,12$	$0,08$	$0,04$	$0,12$	$0,09$	$0,06$	$0,03$
ca	cb	cc	cd	da	db	dc	dd
$0,08$	$0,06$	$0,04$	$0,02$	$0,04$	$0,03$	$0,02$	$0,01$

Wir ignorieren zunächst das zusätzliche Symbol. Man findet folgende Codierung w

aa	ab	ac	ad	ba	bb	bc	bd
100	111	1010	0111	000	010	0010	11010
ca	cb	cc	cd	da	db	dc	dd
1011	0011	11001	01101	11000	01100	110110	110111

Wieso hat man nun damit etwas erreicht? Wir definieren dazu die sogenannte **mittlere Wortlänge**. Diese sei

$$\overline{\ell}(v) := \sum_{a \in A} p(a) \cdot \ell(v(a))$$

wobei $\ell(v(a))$ die Länge der Folge $v(a)$ ist. Es gilt für den Blockcode

$$\begin{aligned}\overline{\ell}(w) = \; & \tfrac{1}{100} \cdot \{3 \cdot (16 + 12 + 12 + 9) + 4 \cdot (8 + 8 + 6 + 6 + 4) + \\ & \qquad 5 \cdot (4 + 4 + 3 + 3 + 2) + 6 \cdot (2 + 1)\} \\ = \; & 3,73\end{aligned}$$

Da wir aber Blöcke der Länge 2 haben, so ist die mittlere Länge bezogen auf $\{a, b, c, d\}$ genau die Hälfte, also $1,865$. Im Falle der Ausgangscodierung bekommen wir $\frac{1}{100} \cdot \{1 \cdot 40 + 2 \cdot 30 + 3 \cdot 20 + 3 \cdot 10\} = \frac{1}{100} \cdot 190 = 1,9$, also leicht schlechter als bei der Blockcodierung.

Nun wollen wir uns dem vernachlässigten Symbol $\diamondsuit$ zuwenden. Man überlege sich, dass von den neun Zweierblöcken, die man mit $\diamondsuit$ und dem alten Alphabet zusätzlich erhält, nur vier braucht, nämlich $a\diamondsuit$, $b\diamondsuit$, $c\diamondsuit$ und $d\diamondsuit$. Die anderen habe jeweils Wahrscheinlichkeit Null des Auftretens. Wie groß ist die Wahrscheinlichkeit, einen solchen Zweierblock anzutreffen? Da er nur am Ende auftritt, hängt dies von der Länge des Textes ab. Falls die übertragenen Texte nur hinreichend groß sind (zum Beispiel dreihundert Zeichen), so sinkt die Summe der Wahrscheinlichkeit für alle vier Blöcke unter die kleinste Wahrscheinlichkeit, in diesem Falle also $0,01$. Man kann sich dann überlegen, dass der Huffman–Code sich nur für dd ändert. Insgesamt sieht eine mögliche Variante dann so aus.

dd	$a\diamondsuit$	$b\diamondsuit$	$c\diamondsuit$	$d\diamondsuit$
1101111	110111000	110111001	110111010	110111011

Für natürliche Sprache ist die Blockcodierung nicht die beste Form der Codierung. Es stellt sich heraus, dass man die Blöcke in der Länge variabel machen muss. Die 50 häufigsten Worte machen immerhin 50 % eines durchschnittlichen Textes aus, sodass es sich offensichtlich lohnt, für diese jeweils einen Block zu reservieren. Ansonsten tut man gut daran, Silben als Blöcke zu nehmen. Wir wollen das jedoch nicht weiter verfolgen.

Eine ganz andere Form der Codierung muss man wählen, wenn die Quelltexte deutlich anderer Natur sind. Ein Beispiel sind die Pixel–Codierung von Buchstaben. Um Buchstaben zu drucken, werden sie in winzige Punkte zerlegt, die entweder schwarz oder weiß sind. Diese Punkte heißen Pixel. Ein Buchstabe (als materielles Zeichen) ist also eine Bitsequenz, welche die Farbe des jeweiligen Pixels angibt. Es stellt sich heraus, dass die Farbe des folgenden Pixels in der Sequenz mit hoher Wahrscheinlichkeit der des vorangegangenen Pixels gleich ist. Daher kann man Pixelfolgen am zum Beispiel dadurch codieren, dass man sie als Zahlenfolgen $n_0, n_1, \ldots, n_{k-1}$ codiert, wobei n_0 die Anzahl der ersten weißen Pixel, n_1 die Anzahl der folgenden schwarzen Pixel, n_2 die Anzahl der darauffolgenden weißen Pixel und so weiter. Wir verlangen noch, dass $n_i > 0$ ist für $0 < i < k$ (warum muss n_0 null sein dürfen?). Es gibt natürlich viele andere Möglichkeiten. Es ist aber klar, dass man bei der Datenkompression immer berücksichtigen muss, um welchen Typ Daten es sich handelt.

23. Teil: Codierung II. Sicherheit.

Codierung kann auch einem anderen Zweck dienen, nämlich, die korrekte Übertragung von Daten zu gewährleisten. Man überlege sich, dass es ziemlich leicht vorkommen kann, dass die ursprüngliche Nachricht nicht korrekt übermittelt wird. Bei der Übertragung durch eine Leitung oder die Luft kann es zu Störungen kommen; wenn wir einen Text niederschreiben oder in einen Rechner eintippen, machen wir gewisse Fehler. Das kann empfindliche Konsequenzen haben, und man sucht deshalb, dies zu vermeiden. Dies geschieht dadurch, dass der Code nunmehr zwei Teile enthält, einen, der zur Erstellung des Quelltextes dient, einen anderen, der die korrekte Übertragung sichert. Dabei bieten sich grundsätzlich zwei Lösungen an. (1.) Man ist daran interessiert, auftretende Fehler so oft wie möglich selbständig reparieren zu können oder (2.) man ist lediglich an der Identifikation von Fehlern interessiert. Das zweite Verfahren bietet sich immer dann an, wenn man interaktiv arbeitet und sich also im Notfall den Zieltext nochmals besorgen kann. Das erste bietet sich an, wenn man mit dem einmal gesendeten Zieltext auskommen muss. Grundsätzlich ist klar, dass man nicht alle Fehler ausschalten kann. Ein Text kann mit gewisser Wahrscheinlichkeit so verunstaltet sein, dass eine Rekonstruktion ohne Kenntnis des Quelltextes unmöglich ist. Daher reduzieren wir das Problem wie folgt. Wir betrachten Blockcodierungen mit fester Blocklänge n. Ferner habe jedes $\varphi(\vec{x})$ für einen Block $\vec{x}$ die gleiche Länge. Jedem Block $\vec{x}$ wird ein zunächst ein Codewort $\varphi(\vec{x})$ zugeordnet und anschließend ein sogenanntes **Kontrollwort** $\gamma(\vec{x})$. Das vollständige Codewort ist dann $\varphi(\vec{x})\gamma(\vec{x})$. Wir wollen $\gamma(\vec{x})$ so gestalten, dass im Falle (1.) lediglich ein Fehler repariert werden kann, im Falle (2.) lediglich ein Fehler erkannt wird.

Im Folgenden wird es sich als nützlich erweisen, wenn wir den Buchstaben des Quellalphabets A sowie des Zielalphabets B jeweils natürliche Zahlen zuordnen. Dann dürfen wir sogar ohne Beschränkung der Allgemeinheit annehmen, dass $A, B \subseteq \mathbb{N}_0$ ist. Sehr oft macht man es zum Beispiel so, dass A die Zahl 1, B die Zahl 2, C die Zahl 3 erhält — und so fort.

Wir wollen uns zunächst mit dem Problem der Fehleridentifikation befassen. Das simpelste Verfahren ist das Verfahren des **Kontrollbits**. Wir betrachten eine Codierung über $\{0, 1\}$. Es bezeichne

$$Q(\vec{y}) := y_0 + y_1 + \ldots y_{n-1} \pmod 2$$

die sogenannte **Quersumme**. Zum Beispiel sei $\gamma(\vec{x})$ so gewählt, dass

$$Q(\varphi(\vec{x})\gamma(\vec{x})) = Q(\varphi(\vec{x})) + Q(\gamma(\vec{x})) = 0$$

Genau dies wird beim ASCII–Code gemacht. Lautet zum Beispiel der Code des Zeichens L 0011001, so wird nun noch 1 angehängt. Damit erhält man den endgültigen Code 00110011 für L. Falls nun bei der Übertragung ein einzelner Fehler unterläuft (also ein Bit 'umklappt'), so wird dies sofort diagnostiziert, indem wir die Quersumme des empfangenen Blocks nehmen. Sei nämlich $\vec{y}$ ein Block der Länge 8; und sei $\vec{y}$ das Resultat aus einem Codewort $\vec{x}$ von höchstens einem Umklappen eines Bits. Dann ist $d(\vec{y}, \vec{x}) \leq 1$, also gilt $Q(\vec{y}) = Q(\vec{x})$ genau dann, wenn auch $\vec{y} = \vec{x}$ ist. (d ist der in Teil 18 besprochene Hamming–Abstand.) Das Kontrollbit erlaubt also, einen Einzelfehler zu diagnostizieren. Es ist klar, dass wir nicht in der Lage sind, den Fehler auch zu reparieren. Falls wir dies wünschen, muss das Kontrollwort wesentlich länger sein Anschaulich gesprochen muss ja in dem Kontrollwort auch Information darüber enthalten sein, wo der Fehler aufgetaucht ist. Daher muss das Kontrollwort mindestens die Länge 3 haben, wenn das Gesamtwort die Länge 8 hat. Man beachte, dass wir ja auch Fehler im Kontrollwort einkalkulieren müssen!

Die Methode des Kontrollbits lässt sich verallgemeinern. Wir nehmen ein $v : A \to B$ derart, dass $v(a)$ jeweils ein Block der Länge κ ist. Sei $B = \{0, 1, \ldots, m-1\}$. Sei $\vec{x} = x_0 x_1 \ldots x_{r-1}$ eine beliebige Folge über B. Sei

$$Q(\vec{x}) := x_0 + x_1 + \ldots + x_{r-1} \pmod m$$

die Quersumme modulo m. Der Code, welcher durch $w(a) := v(a)u$, wo $u = m - Q(v(a))$, definiert ist, kann genau einen einfachen Tippfehler erkennen. Denn wird $\vec{y}$ gelesen, so bilde man $Q(\vec{y})$. Ist $\vec{y}$ in höchstens einem Symbol von einem Codewort verschieden, so ist $\vec{y}$ genau dann ein Codewort, wenn $Q(\vec{y}) = 0$.

Wenden wir uns der Möglichkeit zu, dass es auch noch andere als die einfachen Fehler gibt. Anstatt nun auf Doppelfehler überzugehen, kann man sich die Art der Fehler etwas genauer ansehen. Dazu eine Statistik.

Fehlertyp	Beschreibung	rel. Häufigkeit (in %)
Verwechslung (Einfachfehler)	$a \mapsto b$	79,0
Nachbartransposition	$ab \mapsto ba$	10,2
Sprungtransposition	$acb \mapsto bca$	0,8
Zwillingsfehler	$aa \mapsto bb$	0,6
übrige		9,4

Nach Möglichkeit wollen wir einen Code konstruieren, der Einzelfehler und Nachbartranspositionen erkennt. Eine Möglichkeit bietet der ISBN–Code (International Standard Book Number). Ein typisches ISBN Codewort sieht wie folgt aus: 3-540-05303-4. Wichtig für uns ist die letzte Ziffer; sie ist die Prüfziffer. Sie ist eine Ziffer oder aber X — wir werden sehen, warum. Wir wollen der Einfachheit halber annehmen, es handelt sich um die identische Codierung, das heißt, wir haben es bei den ersten neun Ziffern gar nicht mit einer echten Codierung zu tun. Gegeben eine neunstellige Zahl $a_1a_2\ldots a_9$, definieren wir nun

$$b_{10} := \sum_{i=1}^{9} i \cdot a_i \pmod{11}$$

Zwei Dinge sind bemerkenswert. Zum einen die Tatsache, dass wir eine gewichtete Summe bilden und zum anderen, dass wir modulo 11 rechnen. Deswegen benötigen wir im Übrigen auch das X. Denn wenn $a_{10} = 10$, so sei $b_{10} := X$ ansonsten aber $a_{10} := b_{10}$. Der Witz ist nun folgender. Dieser Code kann nicht nur einfache Tippfehler erkennen sondern auch sämtliche Transpositions– und Sprungtranspositionsfehler! Um das zu sehen, sei $\vec{x} = x_1x_2\ldots x_{10}$ gegeben. (Der Einfachheit wegen sei X gleich 10 gesetzt.) Wir nehmen an, $\vec{x}$ sei durch einen Einfachfehler aus $\vec{a} := a_1a_2\ldots a_{10}$ entstanden. Sei etwa $a_j \neq x_j$. Dann berechnen wir

$$\xi(\vec{x}) := (\sum_{i=1}^{9} i \cdot x_i) - x_{10} \pmod{11}$$

Da $10 \equiv -111$, so ist

$$\xi(\vec{x}) = \sum_{i=1}^{10} i \cdot x_i$$

Im Übrigen ist auch $\xi(\vec{a}) = \sum_{i=1}^{10} i \cdot a_i$. Wir behaupten, dass $\xi(\vec{x}) \neq 0$. Dazu betrachten wir

$$\xi(\vec{a}) - \xi(\vec{x}) \equiv j \cdot (a_j - x_j) \pmod{11}$$

Hier kommt die Tatsache ins Spiel, dass 11 eine Primzahl ist. j hat ein multiplikatives Inverses, da $j \neq 0$ ist. Dies bedeutet, dass $j \cdot (a_j - x_j) \not\equiv 0 \pmod{11}$. Hätten wir also 10 statt 11 gewählt, wäre dies nicht möglich. Nun sei als zweites $\vec{x}$ durch einen Transpositionsfehler aus $\vec{a}$ entstanden, etwa sei $x_j = a_i$ und $x_i = a_j$ für gewisse $0 < i < j < 11$, ansonsten aber $a_k = x_k$. Dann ist

$$\xi(\vec{a}) - \xi(\vec{x}) = i \cdot a_i + j \cdot a_j - i \cdot a_j - j \cdot a_i = (i - j)(a_i - a_j) \ .$$

Es ist $(i-j)(a_i-a_j) \equiv 0$ genau dann, wenn $i = j$ (was ausgeschlossen ist) oder $a_i = a_j$ (in welchem Falle aber $\vec{x} = \vec{a}$). Wir können also Transpositionsfehler, Sprungtranspositionsfehler und mehr entdecken, insgesamt mehr als 90 % der Fehler.

Nun kommen wir zu der Fehlerkorrektur. Hier gibt es recht raffinierte Verfahren. Wir wollen uns aber mit ein paar Handgriffen zufriedengeben. Zunächst einmal wollen wir sehen, unter welchen Umständen eine Fehlerkorrektur möglich ist. Dazu konzentrieren wir uns der Einfachheit halber auf Bitfolgen. Wir erinnern daran, dass der **Hammingabstand** $d(\vec{x}, \vec{y})$ zweier Bitfolgen gleicher Länge gleich der Anzahl der verschiedenen Bits in $\vec{x}$ und $\vec{y}$ ist, das heißt

$$d(\vec{x}, \vec{y}) := \sum_{i<n} |x_i - y_i| = \sum_{x_i \neq y_i} 1 \ .$$

Es sei $\vec{x} \in \{0, 1\}^n$. Dann ist $K_d(\vec{x}) := \{\vec{y} : d(\vec{y}, \vec{x}) \leq d\}$. Dies heißt die **Kugel** mit Radius d um $\vec{x}$. Ein Code heiße d–**korrigierend**, falls bei der Übertragung eines Codeworts bis zu d Fehler auftreten dürfen und dennoch das Codewort eindeutig identifiziert werden kann.

Satz 190 *Es sei $v : A \to \{0,1\}^n$ ein Code derart, dass zu je zwei Codeworten $v(a)$ und $v(b)$ mit $a \neq b$ gilt $K_d(v(a)) \cap K_d(v(b)) = \varnothing$. Dann ist v d–korrigierend.*

Beweis. Hat man $\vec{y}$ und ist $\vec{y}$ aus einem Codewort $v(a)$ entstanden durch höchsten d–fachem Umklappen eines Bits, so ist $d(\vec{y}, v(a)) \leq d$, mithin $\vec{y} \in K_d(v(a))$. Dann existiert nach Voraussetzung kein $b \neq a$ mit $d(\vec{y}, v(b)) \leq d$, also ist a eindeutig. Q. E. D.

Ein Code ist d–korrigierend, falls $d(v(a), v(b)) > 2d$ für alle $a, b \in A$ mit $a \neq b$. Ein Beispiel. Die Punkte $(0,0,0)$ und $(1,1,1)$ haben den Hamming–Abstand 3. Also können wir ein Informationsbit unterbringen, und können einen Fehler korrigieren. Wähle einfach $v : 0 \mapsto (0,0,0), 1 \mapsto (1,1,1)$. Die Ausbeute ist nicht gerade gut. Etwas besser ist folgender Code, der auf 7 Bits gerade 3 Kontrollbits enthält und einen Fehler korrigieren kann. Wir wollen ihn nicht im Detail studieren, sondern ohne Beweis angeben, wie er funktioniert. Es sei H folgende Matrix.

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Das zu übertragende Wort sei (a, b, c, d). Dann seien x, y, z so gewählt, dass für den Vektor $\vec{v} := (x, y, a, z, b, c, d)$ gilt

$$H\vec{v} = \vec{0}$$

Um zu zeigen, dass dies tatsächlich ein 1–korrigierender Code ist, muss man Einiges an algebraischen Hilfsmitteln bereitstellen. (Es zeigt sich also, dass höhere Mathematik am Ende doch ganz praktische Anwendungen haben kann.)

Dies sieht wesentlich besser aus. Jedoch sollte man bedenken, dass die Wahrscheinlichkeit, dass auf 7 Zeichen ein Fehler eintritt, höher ist, als dass auf 3 Zeichen ein Fehler eintritt. Bei 7 Zeichen geht man also schon eher das Risiko ein, dass man einen Doppelfehler hat (und ihn nicht erkennen kann). In Computern ist das Risiko von Doppelfehlern aber bei derart kurzen Worten vernachlässigbar klein.

Definition 191 *Die **Informationsrate** eines Codes mit Codeworten der Länge n über 2 ist definiert durch*

$$\log_2 |A|/n$$

Der erste Code hat eine Informationsrate von 1/3, der zweite von 4/7. Die Informationsrate ist von Belang, wenn man sich für den Aufwand bei der Datenübertragung interessiert. Der Kehrwert der Informationsrate misst, um welchen Faktor der Zieltext länger wird wie der Quelltext. Falls man also die 3–Bit Codierung wählt, so ist der Zieltext immerhin dreimal so lang wie der Quelltext, bei der 7–Bit Codierung immerhin noch fast zweimal so lang!

Wir wollen zum Abschluss eine heuristische Überlegung anstellen, wie viele Codewörter wir unterbringen können, wenn die Blocklänge β und die Anzahl d der zu korrigierenden Fehler vorgegeben ist. Gewiss kann man immer mindestens ein Wort unterbringen! (In diesem Fall ist der zu übertragende Block stets bekannt, die Korrektur gelingt immer. Die Botschaft besteht also nur in der Länge des übertragenen Wortes. Dann ist die Maschinerie der Fehlerkorrektur natürlich höchst überflüssig.) Gegeben ein Codewort, muss es uns sagen, welches Symbol übertragen wurde und an welcher Stelle Fehler aufgetreten sind. Es gibt

$$\sum_{j=0}^{d} \binom{\beta}{j}$$

viele Möglichkeiten, an einem Codewort bis zu d Fehler anzubringen. (Diese entsprechen gerade den Teilmengen der Menge $\{0, 1, \ldots, \beta - 1\}$ der Mächtigkeit $\leq d$.) Wir haben insgesamt 2^β viele Blöcke zur Verfügung, also können wir höchstens

$$\frac{2^\beta}{\sum_{j=0}^{d} \binom{\beta}{j}}$$

Symbole kodieren. Diese Zahl ist in der Regel zu hoch, gibt aber erst mal eine Abschätzung. Für $d = 1$ ergibt sich die sogenannte *Hamming–Volumenschranke*

$$\frac{2^\beta}{\beta + 1}$$

also für $\beta = 2, 3, 4, 5$ gerade 1, 2, 3, 5.

24. Teil: Codierung III. Kryptographie.

Eine mit der Codierung häufig assoziierte Technik ist die *Kryptographie*. Hier geht es darum, eine Botschaft so zu übermitteln, dass sie von Dritten nicht verstanden werden kann, die sie zufällig (oder weniger zufällig) zu lesen bekommen. Wir unterscheiden zwei grundlegend verschiedene Methoden. Die eine ist diejenige, bei der sowohl Sender als auch Empfänger in Besitz des Codes sind. Wir wollen diese *reversible* oder *symmetrische Geheimcodes* nennen. Die andere ist diejenige, bei der der Sender zwar verschlüsseln kann aber nicht entschlüsseln. Diese nennen wir *irreversible* oder *asymmetrische Geheimcodes*. Ein Code besteht, wie wir gesehen haben, aus einem Verschlüsselungsverfahren und einem Entschlüsselungsverfahren. Normalerweise ist es erwünscht, beide zu kennen. Die sogenannten *Public–Key–Systeme* sind aber so gedacht, dass jeder Teilnehmer den Verschlüsselungscode öffentlich macht, damit ihm jeder andere Teilnehmer eine Nachricht zukommen lassen kann, ohne dass dritte (auch nicht andere Teilnehmer) diese lesen können. (Dies ist zum Beispiel bei dem Programm PGP (Pretty Good Privacy) realisiert.) Die Anforderungen an *Public–Key–Systeme* sind also hoch: nicht nur soll die Botschaft ohne Kenntnis des Codes nicht lesbar sein, sie soll auch *mit* Kenntnis des Verschlüsselungsverfahrens unlesbar sein! Dies bedeutet nichts Anderes, als dass die Berechnung des Entschlüsselungsverfahrens aus dem Verschlüsselungsverfahren unmöglich gemacht wird. Da die Verschlüsselung blockweise geschieht, ist das Wort *unmöglich* allerdings unangebracht. Man kann ja einfach alle Blöcke hintereinander verschlüsseln, und bekommt so eine Tabelle, in der man einfach nachschauen kann. Der Witz an den irreversiblen Codes ist also nicht, dass das Entschlüsseln wirklich unmöglich ist, sondern nur, dass der Aufwand dafür unvertretbar hoch ist (mehrere Jahre oder Jahrzehnte).

Betrachten wir zunächst die reversible Codierung. Die einfachste Methode (der sogenannte *Caesar–Code*) ist das Verschieben modulo 26 um eine feste Zahl. Wir ordnen den Buchstaben in der folgenden Weise Zahlen zu.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Unser Alphabet sei Ω, die Zuordnung $\zeta : \Omega \to \{0, 1, 2, 3, \ldots, 25\}$. Die Verschlüsselungsvorschrift ist dann $\varphi(a) := \zeta^{-1}(\zeta(a) + b)$, wobei wir modulo 26 rechnen. Dieser Code ist allerdings leicht zu knacken; man muss ja nur b erraten. Etwas schwieriger ist, wenn wir einfach auf der Grundlage einer beliebigen Bijektion $\pi : \Omega \to \Omega$ arbeiten. Es gibt so viele Bijektionen (26! Stück), dass das sture Durchprobieren nicht sinvoll ist. Allerdings ist man auch hier nicht chancenlos, wenn man ein längeres Stück Text hat. Dann errechne man einfach die Häufigkeit, mit der die Zeichen des Zieltextes auftreten. Diese müssen ja ungefähr den normalen Häufigkeiten entsprechen, mit denen das unverschlüsselte Gegenstück in Texten auftritt. Damit lässt sich das 'e' des Quelltextes und sein Code identifizieren — und sicher noch ein paar mehr Buchstaben. Ist man so weit gekommen, lassen sich die übrigen Buchstaben mit etwas Geschick erraten. Solche statistischen Überlegungen sind eine scharfe Waffe gegen Verschlüsselungsverfahren. (Wir wollen dies nicht weiter diskutieren; es sei nur angemerkt, dass man bei der Entschüsselung nur dann eine Chance hat, wenn die Natur des Codes einigermaßen bekannt ist. Ohne solches Wissen (oder eine gute Vermutung) ist es aussichtslos.) Wir wollen das oben angegebene Verfahren nun gegen diesen statistischen Angriff wie folgt wappnen. Wir wählen eine nicht zu kurze Folge $\vec{b} = b_0 b_1 \ldots b_{p-1} \in \Omega^p$. Diese kann in der Praxis einem sinnvollen Wort entsprechen, muss es aber nicht. Nun wird der Text wie folgt codiert

$$\varphi(x_0 x_1 \ldots x_{m-1}) := y_0 y_1 \ldots y_{m-1} ,$$

wobei $y_{jp+k} := \zeta^{-1}(\zeta(x_{jp+k}) + \zeta(b_k))$. Dies bedeutet, dass das Maß der Verschiebung von der Stelle und von dem Wort $\vec{b}$ abhängt. Kennt der Empfänger diese Methode, so muss er nur $\vec{b}$ zusätzlich zu dem Zieltext kennen, und er kann den Quelltext berechnen. Dies Verfahren ist das Verfahren von *Vigenère*. Da ein fester Buchstabe je nach einer Stelle im Text auf einen anderen Zielbuchstaben geworfen wird, wird die Wahrscheinlichkeit des Quellbuchstaben maskiert. Wir bemerken, dass es reversible Verschüsselungsverfahren gibt (zum Beispiel das Verfahren von *Vernan*), die absolut sicher sind, sofern der (über eine Zufallsfolge ermittelte) Schlüssel geheim bleibt. Hier liegt allerdings die Schwachstelle dieser Verfahren, da sie mit einem variablen Schlüsselwort arbeiten, das ja als erstes einmal übertragen werden muss. Letzteres kann natürlich nur mit einem Code geschehen, der grundsätzlich nicht von dieser Art ist, also unsicher.

Damit beenden wir den Streifzug durch die reversiblen Verschlüsselungstechniken und wenden uns den irreversiblen zu. Wir stellen ein wenig bekanntes Verschlüsselungsverfahren vor. Es sei $A = \{0, 1, \ldots, n-1\}$. Ferner seien Gewichte δ_j, $j < p$, gegeben mit der Eigenschaft, dass $\delta_j > 0$ und $(n+1) \cdot \delta_j \leq \delta_{j+1}$. Gegeben ein Wort der Länge p, $\vec{a} = a_0 a_1 \ldots a_{p-1}$, so sei $W(\vec{a})$ definiert durch

$$W(\vec{a}) := \sum_{j<p} \delta_j \cdot a_j .$$

Ist $W(\vec{a})$ gegeben, so lässt sich sehr leicht $\vec{a}$ ermitteln. Wir zerlegen $P_p := W(\vec{a})$ in $P_p = b_{p-1}\delta_{p-1} + P_{p-1}$ derart, dass $P_{p-1} < \delta_{p-1}$. Ebenso zerlegen wir P_{p-1} in $P_{p-1} = b_{p-2}\delta_{p-2} + P_{p-2}$ mit $P_{p-2} < \delta_{p-2}$. Und so weiter. Falls im Verlaufe der Zerlegung ein $b_j \geq n$ ist oder $P_0 \neq 0$, so haben wir kein Codewort. Andernfalls haben wir die Entschlüsselung $\vec{b} = b_0 b_1 b_2 \dots b_{p-1}$. Wieso ist nun $\vec{b} = \vec{a}$? Dazu sei q der höchste Index $\leq p$ derart, dass $a_q \neq b_q$. Wir nehmen an, dass $a_q > b_q$. Dann ist

$$0 = W(\vec{a}) - W(\vec{b}) = \sum_{j<q} (a_j - b_j)\delta_j \ .$$

Man überlegt sich, dass $|a_j - b_j| \leq n$, daher ist die rechte Summe größer oder gleich

$$(a_q - b_q)\delta_q - \sum_{j<q-1} n\delta_j \geq \delta_q - \sum_{j<q-1} n\delta_j > \delta_0 > 0 \ .$$

Denn $n\delta_j \leq \delta_{j+1} - \delta_j$, sodass

$$\sum_{j<q-1} n\delta_j \leq \sum_{j<q-1} \delta_{j+1} - \delta_j = \delta_{q-1} - \delta_0 \ .$$

Also haben wir ein Verschüsselungsverfahren gefunden. Der Nachteil ist, dass es extrem leicht zu knacken ist. Um dies zu verhindern, bedienen wir uns eines Tricks. Wir wählen eine Primzahl r und eine Zahl u. Ferner sei $uv \equiv 1 \pmod{r}$. Einzige Bedingung an die Primzahl ist, dass kein Codewort $\geq r$ ist. Nun definieren wir als Verschlüsselungsverfahren $\vec{a} \mapsto V(\vec{a})$, wobei

$$V(\vec{a}) := \sum_{j=1}^{p} \gamma_j a_j$$

und $\gamma_j := u \cdot \delta_j \pmod{r}$. Für das Verschlüsseln müssen lediglich die Gewichte γ_j bekannt sein. Wenn man es gut macht, sind die Gewichte so bösartig verteilt, dass das Entschlüsseln sehr schwer fällt. Zum Entschlüsseln geht man so vor. Man erhält die Zahl x. Man berechnet zunächst $v \cdot x \pmod{r}$ und zerlegt dann

$$v \cdot x = \sum_{j<p} \delta_j a_j \pmod{r} \ .$$

Wie wir oben gesehen haben, ist dieses Problem sehr leicht. Dass wir damit korrekt entschlüsseln, belegt folgende Rechnung.

$$\begin{aligned} v \cdot V(\vec{a}) &= v \textstyle\sum_{j<p} \gamma_j a_j \\ &= \textstyle\sum_{j<p} vu\delta_j a_j \\ &= \textstyle\sum_{j<p} \delta_j a_j \\ &= W(\vec{a}) \pmod{r} \end{aligned}$$

Ein Beispiel. Unser Alphabet sei $A = \{0, 1, 2\}$. Wir wollen Blöcke der Länge 5 codieren. Die Gewichte sind

δ_0	δ_1	δ_2	δ_3	δ_4
1	3	9	27	81

Offensichtlich ist $3\delta_j \leq \delta_{j+1}$, $j < 4$. Die größte Zahl, die damit gebildet werden kann, ist $\sum_{j<5} 2\delta_j = 2(1+3+9+27+81) = 242$. Wir nehmen daher als Primzahl $r = 251$. Es gilt $69 \times 211 \equiv 1 \pmod{251}$. Wenn wir also die Gewichte mit 211 multiplizieren, erhalten wir die öffentlichen Gewichte

γ_0	γ_1	γ_2	γ_3	γ_4
211	131	142	175	23

Es sei zum Beispiel die Zahl 774 übermittelt worden. Dann ist $774 \equiv 21 \pmod{251}$. Wir berechnen $69 \cdot 21 = 1449 \equiv 194 \pmod{251}$. Es ist nun leicht zu ermitteln, dass $194 = 2 \times 81 + 1 \times 27 + 1 \times 3 + 2 \times 1$, also ist die übermittelte Nachricht $(2, 1, 0, 1, 2)$. Zur Probe berechnen wir

$$2 \times 211 + 1 \times 131 + 0 \times 142 + 1 \times 175 + 2 \times 23 = 774$$

Dieses Verfahren wird dadurch sicher gemacht, dass man die Blocklänge groß wählt und die Primzahl ebenso. Wichtig ist, dass die offengelegten Gewichte gehörig durcheinandergewirbelt sind. Natürlich garantiert dies nicht, dass es nicht doch raffinierte Verfahren gibt, die das Problem der unbefugten Entschlüsselung in vertretbarer Zeit lösen. Man sollte jedenfalls auch hier im Auge behalten, dass man solche Codes immer auch mit statistischen Methoden angreifen kann, sodass man auch hier etwas raffinierter vorgehen sollte. Das eben diskutierte Verfahren ist im Übrigen geknackt worden, sodass man sich anderen Methoden zugewendet hat, so zum Beispiel dem RSA Algorithmus.

25. Teil: Modale und Dynamische Logik I.

Dieser und der folgende Teil sind nicht offizieller Teil des Skripts. Ein knappe aber gute Darstellung des Materials findet sich in dem Buch von Robert Goldblatt: *Logics of Time and Computation*, CSLI Lecture Notes No. 7, CSLI, Stanford, 1987.

Wir haben in der Aussagenlogik ausführlich Aussagen und deren Verknüpfungen betrachtet. In diesem Teil wollen wir uns weiter mit Aussagenlogik befassen, allerdings wollen wir nicht wie in der Prädikatenlogik Aussagen weiter analysieren sondern neue Operatoren auf Aussagen einführen. Wir beginnen in diesem Abschnitt mit Modal- und Zeitlogik und werden in dem nächsten auf die sogenannte dynamische Logik zu sprechen kommen. Diese Logiken sind Erweiterungen der Aussagenlogik. Sie gehen aber davon aus, dass Aussagen nicht einfach nur wahr oder falsch sind, sondern ihren Wahrheitswert je nach Situation ändern. Wir beginnen wie üblich mit der Syntax.

Definition 192 *Es sei* $M := \{\mathtt{0}, \mathtt{1}, \mathtt{p}, \mathtt{(}, \mathtt{)}, \top, \neg, \wedge, \vee, \Box, \Diamond\}$. *Die Menge MAus der **wohlgeformten modalen Zeichenketten** oder schlicht der **modalen Aussagenterme** ist die kleinste Teilmenge von* M^*, *für die gilt:*

1. $\mathtt{p}\vec{\alpha} \in MAus$, *wo* $\vec{\alpha}$ *eine Binärfolge ist.*
2. $\top \in MAus$.
3. *Ist* $\vec{x} \in MAus$, *so ist auch* $\mathtt{(}\neg\vec{x}\mathtt{)}, \mathtt{(}\Box\vec{x}\mathtt{)}, \mathtt{(}\Diamond\vec{x}\mathtt{)} \in MAus$.
4. *Sind* $\vec{x}$ *und* $\vec{y}$ *in MAus, so auch* $\mathtt{(}\vec{x} \wedge \vec{y}\mathtt{)}$ *und* $\mathtt{(}\vec{x} \vee \vec{y}\mathtt{)}$.

*Hierbei ist eine **Binärfolge** eine endliche Folge aus* $\mathtt{0}$ *und* $\mathtt{1}$. *Wir nennen die Folge* $\mathtt{p}\vec{\alpha}$ *eine **Variable**. Die Menge der Variablen heißt* Var. *Die Junktoren* $\Box$ *und* $\Diamond$ *heißen **Modaloperatoren**.*

Wir verwenden im Folgenden die Konventionen aus der Aussagenlogik. Wie wir in einer Übung gesehen haben, ist es ungefährlich, bei einstelligen Junktoren die Klammern wegzulassen. Dies werden wir ohne Warnung tun. Ferner gebrauchen wir die Konvention $\mathtt{p}_i$, $i \in \omega$, und wählen φ, χ und so weiter als Variable für Aussagen.

Definition 193 *Eine **Kripke–Struktur** ist ein Paar* $\mathfrak{W} = \langle W, \lhd \rangle$, *wo* W *eine Menge und* $\lhd \subseteq W^2$ *eine zweistellige Relation ist. Man spricht von* W

*als **Menge der möglichen Welten** und von $\lhd$ als der **Zugänglichkeitsrelation**. Eine **Belegung auf** $\mathfrak{W}$ ist eine Funktion $\beta : Var \to \wp(W)$. Ein **Kripke–Modell** ist ein Paar $\langle \mathfrak{W}, \beta \rangle$, wo $\mathfrak{W}$ eine Kripke–Struktur und β eine Belegung auf $\mathfrak{W}$ ist.*

Intuitiv bedeutet dies, dass der Wahrheitswert einer Aussage von Welt zu Welt verschieden ausfallen kann. Der Begriff *Welt* sollte nicht zu eng verstanden werden. Man darf die Welten auch als Zeitpunkte sehen, als Orte, als Zustände einer Maschine, und so weiter. Auf diese Interpretationen werden wir noch eingehen. Zunächst werden wir definieren, was es heißt, eine Formel sei unter einer Belegung erfüllt. Man beachte insbesondere die letzten beiden Klauseln der folgenden Definition.

Definition 194 *Es sei $\langle W, \lhd, \beta \rangle$ ein Kripke–Modell und $w \in W$. Die Relation $\langle W, \lhd, w, \beta \rangle \vDash \varphi$ wird induktiv wie folgt erklärt.*

1. $\langle W, \lhd, w, \beta \rangle \vDash \mathtt{p}_i$ *genau dann, wenn* $w \in \beta(\mathtt{p}_i)$.
2. $\langle W, \lhd, w, \beta \rangle \vDash \neg\varphi$ *genau dann, wenn* $\langle W, \lhd, w, \beta \rangle \nvDash \varphi$.
3. $\langle W, \lhd, w, \beta \rangle \vDash \varphi \wedge \chi$ *genau dann, wenn* $\langle W, \lhd, w, \beta \rangle \vDash \varphi; \chi$.
4. $\langle W, \lhd, w, \beta \rangle \vDash \varphi \vee \chi$ *genau dann, wenn* $\langle W, \lhd, w, \beta \rangle \vDash \varphi$ *oder* $\langle W, \lhd, w, \beta \rangle \vDash \chi$.
5. $\langle W, \lhd, w, \beta \rangle \vDash \Box\varphi$ *genau dann, wenn für alle* $u \in W$ *mit* $w \lhd u$ *gilt* $\langle W, \lhd, u, \beta \rangle \vDash \varphi$.
6. $\langle W, \lhd, w, \beta \rangle \vDash \Diamond\varphi$ *genau dann, wenn ein* $u \in W$ *existiert mit* $w \lhd u$ *und* $\langle W, \lhd, u, \beta \rangle \vDash \varphi$.

*Wir schreiben $\langle \mathfrak{W}, \beta \rangle \vDash \varphi$ falls für alle $w \in W$ gilt $\langle \mathfrak{W}, w, \beta \rangle \vDash \varphi$ und $\mathfrak{W} \vDash \varphi$, falls $\langle \mathfrak{W}, \beta \rangle \vDash \varphi$ für alle Belegungen auf $\mathfrak{W}$. Im letzten Fall sagen wir auch, φ sei in $\mathfrak{W}$ **gültig**.*

Definition 195 *Eine **(normale) Modallogik** ist eine Teilmenge $\Theta \subseteq MAus$ mit folgenden Eigenschaften:*

1. *Für jede Formel $\varphi \in Aus$, welche Tautologie der Aussagenlogik ist, ist $\varphi \in \Theta$. (Hier würde auch genügen, dass die Formeln (a0) – (a11) in Θ sind.)*

2. $\Box(\varphi \to \chi) \to (\Box\varphi \to \Box\chi) \in \Theta$ *und* $\Box\varphi \leftrightarrow \neg\Diamond\neg\varphi \in \Theta$.

3. Mit $\varphi \to \chi \in \Theta$ *und* $\varphi \in \Theta$ *ist auch* $\chi \in \Theta$.

4. Mit $\varphi \in \Theta$ *ist auch* $\varphi^\sigma \in \Theta$, *wo* $\sigma : \mathit{Var} \to \mathit{MAus}$ *eine Substitution.*

5. Mit $\varphi \in \Theta$ *ist auch* $\Box\varphi \in \Theta$.

Die kleinste normale Modallogik wird mit ***K*** *bezeichnet.*

Die letzten drei Klauseln beschreiben Abschlusseigenschaften. Die erste ist die schon bekannte Regel **Modus Ponens**, die zweite die Abgeschlossenheit unter Substitution und die dritte die Abgeschlossenheit unter der Regel **MN** (auch **Necessitationsregel** genannt). Aufgrund der ersten Eigenschaft gilt: ist Θ eine konsistente normale Modallogik (das heißt, ist $\Theta \neq \mathit{MAus}$), so ist $\Theta \cap \mathit{Aus}$ die Menge der Aussagenlogischen Tautologien. Das bedeutet, dass wir es wie man sagt mit einer **konservativen Erweiterung** zu tun haben. Wir verstärken nicht die logischen Prinzipien sondern nur die Ausdruckskraft. Wir teilen hier ohne Beweis folgendes Resultat mit.

Satz 196 *Genau dann ist* $\varphi \in \boldsymbol{K}$, *wenn* φ *in jeder Kripke–Struktur* $\mathfrak{W}$ *gilt. Es ist sogar* $\varphi \in \boldsymbol{K}$ *genau dann, wenn* φ *in allen* ***endlichen*** *Kripke–Strukturen gilt.*

Man kann also die Logik der Kripke–Strukturen genau erfassen. Daraus kann man ein Verfahren stricken, welches für gegebenes φ entscheidet, ob $\varphi \in \mathbf{K}$ oder nicht. Dazu baut man zwei Maschinen. Die erste erzeugt schrittweise alle Formeln aus $\mathbf{K}$. Dies ist möglich, weil wir die Menge $\mathbf{K}$ durch einen schrittweisen Prozess definiert haben. Hat man nun eine Formel φ und die erste Maschine wirft φ irgendwann aus, so ist $\varphi \in \mathbf{K}$. Aber da man nie weiß, wann das sein wird, benötigt man eine zweite Maschine. Diese zählt die Menge $\mathit{MAus} - \mathbf{K}$ wie folgt auf. Diese generiert schrittweise alle endliche Kripke–Modelle und alle Formeln und schaut nach, ob die gegebene Formel und dem Modell gilt oder nicht. Wenn nicht, so ist $\varphi \notin \mathbf{K}$ und die Maschine gibt φ aus. Nach dem obenstehenden Satz existiert, falls $\varphi \notin \mathbf{K}$, ein endliches Gegenmodell. Also wird die zweite Maschine irgendwann φ ausgeben.

Aus Satz 196 folgt, dass es keine Aussage φ gibt, die nur auf den endlichen Strukturen gilt. Deen eine Aussage, die auf allen endlichen Strukturen gilt, gilt dann schon auf allen Strukturen schlechthin.

Satz 197 *Für eine Kripke–Struktur* $\mathfrak{W} = \langle W, \lhd \rangle$ *gilt: genau dann ist* $\lhd$ *reflexiv, wenn* $\mathfrak{W} \vDash \mathsf{p}_0 \to \Diamond \mathsf{p}_0$.

Beweis. Es sei $\lhd$ reflexiv. Sei ferner β eine Belegung auf $\mathfrak{W}$ und $w \in W$ mit $\langle \mathfrak{W}, w, \beta \rangle \vDash \mathsf{p}_0$. Dann gilt $\langle \mathfrak{W}, w, \beta \rangle \vDash \Diamond \mathsf{p}_0$, da $w \lhd w$. Nun sei $\lhd$ nicht reflexiv. Dann existiert ein w derart, dass nicht $w \lhd w$. Setze $\beta(\mathsf{p}_0) := \{w\}$. Dann gilt $\langle \mathfrak{W}, w, \beta \rangle \vDash \mathsf{p}_0; \neg \Diamond \mathsf{p}_0$. Q. E. D.

Ebenso ist $\mathfrak{W}$ genau dann transitiv, wenn $\mathfrak{W} \vDash \Diamond\Diamond \mathsf{p}_0 \to \Diamond \mathsf{p}_0$. Auf diese Weise etablieren gewisse Aussagen gewisse Bedingungen auf Kripke–Strukturen. Formal sieht das so aus. Es bezeichne $\mathsf{K} \oplus \Diamond\Diamond \mathsf{p}_0 \to \Diamond \mathsf{p}_0$ die kleinste Modallogik, welche die Aussage $\Diamond\Diamond \mathsf{p}_0 \to \Diamond \mathsf{p}_0$ enthält (diese heißt auch $\mathsf{K4}$). Dann gilt: $\mathfrak{W} \vDash \varphi$ für jedes $\varphi \in \mathsf{K4}$ genau dann, wenn $\mathfrak{W}$ transitiv ist. Diese Korrespondenz zwischen Klassen von Strukturen und Logiken wollen wir nun abstrakt fassen.

Definition 198 *Sei* Θ *eine normale Modallogik. Dann bezeichnet* $\mathsf{Mod}\,\Theta$ *die Klasse aller Kripke–Strukturen* $\mathfrak{W}$ *mit* $\mathfrak{W} \vDash \varphi$ *für jedes* $\varphi \in \Theta$*, die sogenannte* ***Modellklasse*** *von* Θ*. Ist* $\mathcal{K}$ *eine Klasse von Kripke–Strukturen, so bezeichnet* $\mathsf{Th}\,\mathcal{K} := \{\varphi :$ *für alle* $\mathfrak{W} \in \mathcal{K}\}$ *die sogenannte* ***Theorie von*** $\mathcal{K}$.

Das Folgende ist leicht zu zeigen.

Theorem 199 1. *Ist* $\mathcal{K}$ *eine Klasse von Kripke–Strukturen, so ist* $\mathsf{Th}\,\mathcal{K}$ *eine normale Modallogik.*

2. *Ist* $\mathcal{K} \subseteq \mathcal{L}$*, so gilt* $\mathsf{Th}\,\mathcal{K} \supseteq \mathsf{Th}\,\mathcal{L}$.

3. *Ist* $\Theta \subseteq \Lambda$*, so ist* $\mathsf{Mod}\,\Theta \supseteq \mathsf{Mod}\,\Lambda$.

Unglücklicherweise gibt es Klassen $\mathcal{K}$ und $\mathcal{L}$, die die gleiche Theorie haben und es gibt auch Logiken, die die gleiche Klasse von Kripke–Strukturen besitzen. Die erste Tatsache ist leicht zu sehen (die zweite war einigermaßen überraschend und wurde erst 1975 von Thomason gezeigt). Es gibt genau $\aleph_0$ viele Aussagen, und so gibt es höchstens $2^{\aleph_0}$ viele Logiken (diese sind ja Mengen von Aussagen). Es gibt aber weit mehr Klassen von Kripke–Strukturen. Man überlege sich, dass auf jeder Menge der Mächtigkeit κ 2^{κ} viele Kripke–Stukturen existieren. Den Satz 199 kann man aber durch einige Methoden in eine eins–zu–eins Korrespondenz zwischen gewissen Klassen von (verallgemeinerten) Kripke–Strukturen und normalen Modallogiken umformen. Es sei noch angemerkt, dass es in der Tat $2^{\aleph_0}$ viele normale Modallogiken gibt, obwohl ja beleibe nicht jede Menge von Aussagen eine Logik ist.

26. Teil: Modale und Dynamische Logik II.

Die dynamische Aussagenlogik ist eine Erweiterung der Modallogik, welche es erlaubt, komplexe Modalitäten aus einfachen zu gewinnen. Die Modaloperatoren werden in der dynamischen Logik als Programme aufgefasst. Programme werden wiederum als Relationen zwischen den Zuständen eines Computers interpretiert. Dies erlaubt, Aussagen über das zeitliche Verhalten und ihre Interaktion zu formulieren und zu verifizieren. Die Syntax der dynmischen Aussagenlogik ist wie folgt.

Definition 200 *Es se Q eine endliche Menge. Es sei*

$$D_Q := Q \cup \{\mathtt{0}, \mathtt{1}, \mathtt{p}, \mathtt{(}, \mathtt{)}, \top, \neg, \wedge, \vee, \mathtt{;}, \cup, \mathtt{?}, ^*, \langle, \rangle, [,]\} .$$

*Wir definieren in Tandem die Menge $Prog_Q$ der Q–**Programme** sowie $DAus_Q$, die Menge der **wohlgeformten Q–dynamischen Zeichenketten** oder schlicht der **dynamischen Aussagenterme** ist die kleinste Teilmenge von D_Q^*, für die gilt:*

1. *Ist $q \in Q$, so ist $q \in Prog_Q$.*
2. *Ist $\vec{y} \in Prog_Q$, so ist $\mathtt{(}\vec{y}^*\mathtt{)} \in \mathtt{Prog}_Q$.*
3. *Ist $\vec{y}, \vec{z} \in Prog_Q$, so ist $\mathtt{(}\vec{y};\vec{z}\mathtt{)}, \mathtt{(}\vec{y} \cup \vec{z}\mathtt{)} \in \mathtt{Prog}_Q$.*
4. *Ist $\vec{x} \in DAus_Q$, so ist $\mathtt{(}\vec{x}\mathtt{?)} \in Prog_Q$.*
5. *$\mathtt{p}\vec{\alpha} \in DAus_Q$, wo $\vec{\alpha}$ eine Binärfolge ist.*
6. *$\top \in DAus_Q$.*
7. *Ist $\vec{x} \in DAus_Q$, so ist auch $\mathtt{(}\neg\vec{x}\mathtt{)}, \mathtt{(}[\vec{y}]\vec{x}\mathtt{)}, \mathtt{(}\langle\vec{y}\rangle\vec{x}\mathtt{)} \in DAus_Q$.*
8. *Sind $\vec{x}$ und $\vec{y}$ in $DAus_Q$, so auch $\mathtt{(}\vec{x} \wedge \vec{y}\mathtt{)}$ und $\mathtt{(}\vec{x} \vee \vec{y}\mathtt{)}$.*

*Hierbei ist eine **Binärfolge** eine endliche Folge aus $\mathtt{0}$ und $\mathtt{1}$. Wir nennen die Folge $\mathtt{p}\vec{\alpha}$ eine **Variable**. Die Menge der Variablen heißt Var.*

Definition 201 *Eine **Kripke–Struktur** für die Q–dynamische Logik ist ein Paar $\langle W, R\rangle$, wo W eine Menge ist, die Menge der **Welten**, und $R : Q \to \wp(W \times W)$ eine Funktion, welche jedem Programm eine zweistellige Relation auf W zuordnet. Eine Belegung ist eine Funktion $\beta : \mathrm{Var} \to \wp(W)$.*

Wir verwenden α, β und so weiter als Metavariable über Programme. (Man beachte, dass Programme hier technisch gesehen Konstanten sind.) Wir schreiben $v \xrightarrow{\alpha} w$ falls $\langle v, w\rangle \in R(\alpha)$. Man sagt auch, es **existiere ein α–Übergang von v nach w**. Wir setzen R auf ganz $Prog_Q$ in folgender Weise fort, indem wir gleichzeitig definieren, wann eine Formel in einer Struktur gilt.

1. $\langle W, R, w, \beta\rangle \vDash \mathtt{p}_i$ genau dann, wenn $w \in \beta(\mathtt{p}_i)$.

2. $\langle W, R, w, \beta\rangle \vDash \neg\varphi$ genau dann, wenn $\langle W, R, w, \beta\rangle \nvDash \varphi$.

3. $\langle W, R, w, \beta\rangle \vDash \varphi \wedge \chi$ genau dann, wenn $\langle W, R, w, \beta\rangle \nvDash \varphi; \chi$.

4. $\langle W, R, w, \beta\rangle \vDash \varphi \vee \chi$ genau dann, wenn $\langle W, R, w, \beta\rangle \nvDash \varphi$ oder $\langle W, R, w, \beta\rangle \vDash \chi$.

5. $\langle W, R, w, \beta\rangle \vDash [\alpha]\varphi$ genau dann, wenn für alle $u \in W$ mit $w \xrightarrow{\alpha} u$ gilt $\langle W, R, u, \beta\rangle \vDash \varphi$.

6. $\langle W, R, w, \beta\rangle \vDash \langle\alpha\rangle\varphi$ genau dann, wenn ein $u \in W$ existiert mit $w \xrightarrow{\alpha} u$ und $\langle W, R, u, \beta\rangle \vDash \varphi$.

$$\begin{array}{lcl} R(\alpha \cup \beta) & := & R(\alpha) \cup R(\beta) \\ P(\alpha; \beta) & := & R(\alpha) \circ R(\beta) \\ R(\alpha^*) & := & R(\alpha)^* \\ R(\varphi?) & := & \{\langle v, v\rangle : \langle W, R, v, \beta\rangle \vDash \varphi\} \end{array}$$

Wie man leicht nachrechnet, ist $v \xrightarrow{\alpha\cup\beta} w$ genau dann, wenn $v \xrightarrow{\alpha} w$ oder $v \xrightarrow{\beta} w$. Ferner ist $v \xrightarrow{\alpha;\beta} w$ genau dann, wenn ein $u \in W$ existiert mit $v \xrightarrow{\alpha} u$ und $u \xrightarrow{\beta} w$. $v \xrightarrow{\alpha^*} w$ genau dann, wenn eine Folge existiert $v \xrightarrow{\alpha} v_1 \xrightarrow{\beta} v_2 \ldots v_{n-1} \xrightarrow{\beta} w$. Endlich ist $v \xrightarrow{\varphi?} w$ genau dann, wenn $v = w$ und $\langle W, R, v, \beta\rangle \vDash \varphi$.

Kommen wir nun auf die Programme zurück. Wir nehmen nicht an, dass Programme deterministisch sind. Sie mögen entweder irgendwann aufhören (was sehr erwünscht ist) oder auch verzweigen, das heißt, von einem Punkt aus sind mehrere Fortführungen möglich (was nicht immer unerwünscht ist). Wir lesen nun die Formeln $[\alpha]\varphi$ und $\langle\alpha\rangle\varphi$ wie folgt.

$[\alpha]\varphi$	*nach jeder Ausführung von α gilt φ*
$\langle\alpha\rangle\varphi$	*nach einer Ausführung von α gilt φ*

Es lassen sich ferner komplexe Programmkonstrukte etablieren wie

$$\begin{array}{lcl} \textsf{if } \varphi \textsf{ then } \alpha \textsf{ else } \beta \textsf{ fi} & := & (\varphi?);\alpha \cup (\neg\varphi?);\beta \\ \textsf{while } \varphi \textsf{ do } \alpha \textsf{ od} & := & (\varphi?;\alpha)^*;\neg\varphi? \\ \textsf{until } \varphi \textsf{ do } \alpha \textsf{ od} & := & (\neg\varphi?;\alpha)^*;\varphi? \end{array}$$

Die erste ist die sogenannte bedingte Anweisung. (Man kann auch bedingte Anweisungen mit beliebig vielen Bedingungen formulieren.) Die beiden anderen heißen die while– bzw. until–Schleife. Um die Korrektheit der letzten zwei Definitionen zu sehen, betrachten wir, wann es einen Übergang von v nach w mittels $(\varphi?;\alpha)^*;\neg\varphi?$ gibt. Dies ist der Fall, wenn es eine Sequenz der folgenden Art gibt.

$$v = v_0 \stackrel{\varphi?;\alpha}{\to} v_1 \stackrel{\varphi?;\alpha}{\to} v_2 \stackrel{\varphi?;\alpha}{\to} v_3 \ldots v_{n-1} \stackrel{\varphi?;\alpha}{\to} v_n \stackrel{\neg\varphi?}{\to} w$$

Dies wiederum ist der Fall, wenn

$$v = v_0 \stackrel{\alpha}{\to} v_1 \stackrel{\alpha}{\to} v_2 \stackrel{\alpha}{\to} v_3 \ldots v_{n-1} \stackrel{\alpha}{\to} v_n = w$$

wobei alle Welten v_i, $i < n$, die Formel φ erfüllen, v_n aber nicht. Genauso zeigt man, dass es einen Übergang von v nach w mittels $(\neg\varphi?;\alpha)^*;\varphi?$ gibt, falls es eine endliche Folge v_i, $i < n+1$, gibt mit

$$v = v_0 \stackrel{\alpha}{\to} v_1 \stackrel{\alpha}{\to} v_2 \stackrel{\alpha}{\to} v_3 \ldots v_{n-1} \stackrel{\alpha}{\to} v_n = w\ ,$$

wobei φ bei v_i gilt für alle $i < n$, aber nicht bei v_n.

Definition 202 *Eine (**normale**) **dynamische Logik** ist eine Teilmenge $\Theta \subseteq DAus_Q$ mit folgenden Eigenschaften:*

1. *Die Formeln (a0) – (a11) sind in Θ.*

2. *$[\alpha](\mathtt{p}_0 \to \mathtt{p}_1) \to ([\alpha]\mathtt{p}_0 \to [\alpha]\mathtt{p}_1) \in \Theta$ und $[\alpha]\mathtt{p}_0 \leftrightarrow \neg\langle\alpha\rangle\neg\mathtt{p}_0 \in \Theta$ für alle Programme α.*

3. *$[\varphi?]\mathtt{p}_1 \leftrightarrow (\mathtt{p}_1 \to \mathtt{p}_1) \in \Theta$ für alle Aussagen φ.*

4. *$[\alpha \cup \beta]\mathtt{p}_0 \leftrightarrow [\alpha]\mathtt{p}_0 \wedge [\beta]\mathtt{p}_0 \in \Theta$ für alle Programme α, β.*

5. *$[\alpha;\beta]\mathtt{p}_0 \leftrightarrow [\alpha][\beta]\mathtt{p}_0 \in \Theta$ für alle Programme α, β.*

6. *$[\alpha]\mathtt{p}_0 \to \mathtt{p}_0 \wedge [\alpha;\alpha^*]\mathtt{p}_0 \in \Theta$ für alle Programme α.*

7. $p_0 \wedge [\alpha^*](p_0 \to [\alpha]p_0) \to [\alpha^*]p_0 \in \Theta$ *für alle Programme* α.

8. Mit $\varphi \to \chi \in \Theta$ *und* $\varphi \in \Theta$ *ist auch* $\chi \in \Theta$.

9. Mit $\varphi \in \Theta$ *ist auch* $\varphi^\sigma \in \Theta$, *wo* $\sigma : Var \to DAus_Q$ *eine Substitution ist.*

10. Mit $\varphi \in \Theta$ *ist auch* $[\alpha]\varphi \in \Theta$.

Die kleinste normale dynamische Logik wird mit $\boldsymbol{PDL}_Q$ *bezeichnet.*

Man beachte, dass das Axiomensystem unendlich ist. Um zu sehen, dass die Axiome das gewünschte leisten, muss man der Reihe nach zeigen, dass sie in den geünschten Strukturen richtig sind sowie, dass sie *nur* in den gewünschten Strukturen richtig sind. Dies wollen wir nicht in allen Einzelheiten vormachen.

Lemma 203 *1. Genau dann gilt* $\mathfrak{W} \vDash [\gamma]p_0 \leftrightarrow [\alpha][\beta]p_0$, *wenn* $R(\gamma) = R(\alpha) \circ R(\beta)$.

2. Genau dann gilt $\mathfrak{W} \vDash [\gamma]p_0 \leftrightarrow [\alpha]p_0 \wedge [\beta]p_0$, *wenn* $R(\gamma) = R(\alpha) \cup R(\beta)$.

Beweis. Wir zeigen die zweite Behauptung. Es sei $R(\gamma) \neq R(\alpha) \cup R(\beta)$. Dann existieren $v, w \in W$ derart, dass entweder (Fall 1) $v \xrightarrow{\gamma} w$ aber nicht $v \xrightarrow{\alpha} w$ und auch nicht $v \xrightarrow{\beta} w$, oder (Fall 2) nicht $v \xrightarrow{\gamma} w$ aber $v \xrightarrow{\alpha} w$ oder (Fall 3) $v \xrightarrow{\beta} w$ aber nicht $v \xrightarrow{\gamma} w$. Setze $\beta(p_0) := W - \{w\}$. (Fall 1) Es gilt $\langle\mathfrak{W}, v, \beta\rangle \vDash \neg[\gamma]p_0; [\alpha]p_0; [\beta]p_0$. (Fall 2) Es gilt $\langle\mathfrak{W}, v, \beta\rangle \vDash [\gamma]p_0; \neg[\alpha]p_0; [\beta]p_0$. (Fall 3) Es gilt $\langle\mathfrak{W}, v, \beta\rangle \vDash [\gamma]p_0; [\alpha]p_0; \neg[\beta]p_0$. In allen drei Fällen ist also $\langle\mathfrak{W}, v, \beta\rangle \nvDash [\gamma]p_0 \leftrightarrow ([\alpha]p_0 \wedge [\beta]p_0)$. Nun sei $R(\gamma) = R(\alpha) \cup R(\beta)$. Sei β eine Belegung und $v \in W$. Angenommen, $\langle\mathfrak{W}, v, \beta\rangle \vDash [\gamma]p_0$. Sei $v \xrightarrow{\alpha} w$. Dann ist auch $v \xrightarrow{\gamma} w$, nach Vorasussetzung, also $\langle\mathfrak{W}, w, \beta\rangle \vDash p_0$. Dies zeigt $\langle\mathfrak{W}, v, \beta\rangle \vDash [\alpha]p_0$. Ebenso sieht man $\langle\mathfrak{W}, v, \beta\rangle \vDash [\beta]p_0$. Umgekehrt, sei $\langle\mathfrak{W}, v, \beta\rangle \vDash [\alpha]p_0; [\beta]p_0$ und $v \xrightarrow{\gamma} w$. Dann ist $v \xrightarrow{\alpha} w$ oder $v \xrightarrow{\beta} w$, und in beiden Fällen gilt $\langle\mathfrak{W}, w, \beta\rangle \vDash p_0$. Also haben wir $\langle\mathfrak{W}, v, \beta\rangle \vDash [\gamma]p_0$. Q. E. D.

Setzt man also $\gamma := \alpha; \beta$ im ersten und $\gamma := \alpha \cup \beta$, so erhält man, dass die Axiome genau die Verhältnisse der Strukturen wiederspiegeln. Die meisten Schwierigkeiten macht der *.

Lemma 204 *Genau dann ist* $\mathfrak{W} \vDash [\beta]p_0 \to p_0 \wedge [\alpha; \beta]p_0$, *wenn* $R(\beta)$ *reflexiv ist und* $R(\alpha; \beta) \subseteq R(\beta)$.

Wir sagen auch, $R(\beta)$ sei unter $R(\alpha)$ **abgeschlossen**, falls gilt $R(\beta) \supseteq R(\alpha;\beta)$.

Lemma 205 *Ist $R(\beta)$ reflexiv und unter $R(\alpha)$ abgeschlossen, so ist $R(\alpha)^* \subseteq R(\beta)$.*

Beweis. Zunächst ist $R(\beta)$ reflexiv, sodass $R(\alpha)^0 \subseteq R(\beta)$. Ferner zeigen wir: ist $R(\alpha)^n \subseteq R(\beta)$, so auch $R(\alpha^{n+1} \subseteq R(\beta)$. Sei dazu $v \overset{\alpha^{n+1}}{\to} w$. Dann existiert ein u mit $v \overset{\alpha}{\to} u \overset{\alpha^n}{v}$. Nach Voraussetzung ist dann $u \overset{\beta}{\to} v$ und so $w \overset{\alpha;\beta}{v}$, sodass wir $w \overset{\beta}{\to} v$ haben. Q. E. D.

Lemma 206 *Es seien α und β Programme und $\mathfrak{W}$ eine Kripke–Struktur derart, dass $R(\beta)$ reflexiv und unter $R(\alpha)$ abgeschlossen. Genau dann ist $\langle W, R\rangle \vDash \mathsf{p}_0 \wedge [\beta](\mathsf{p}_0 \to [\alpha]\mathsf{p}_0) \to [\beta]\mathsf{p}_0$, wenn $R(\beta) = R(\alpha)^*$.*

Beweis. Es sei $R(\beta) \neq R(\alpha)^*$. Da $R(\beta) \supseteq R(\alpha)^*$ nach Lemma 205, so existiert ein Paar $\langle v, w\rangle$ derart, dass $v \overset{\beta}{\to} w$ aber für kein $n \in \omega$: $v \overset{\alpha^n}{\to} w$. Setze $\beta(\mathsf{p}_0) := \{u :$ für ein $n \in \omega : v \overset{\alpha^n}{\to} u\} =: P$. Dann gilt $\langle \mathfrak{W}, v, \beta\rangle \vDash \mathsf{p}_0$, da $v \in P$. Ferner: sei $v \overset{\beta}{\to} u$ derart, dass $\langle \mathfrak{W}, u, \beta\rangle \vDash \mathsf{p}_0$. Dann ist $u \in P$, also $v \overset{\alpha^n}{\to} u$ für ein n. Sei nun u' irgendein α–Nachfolger von u. Dann ist $v \overset{\alpha^{n+1}}{\to} u'$, also $u' \in P$. Deswegen gilt $\langle \mathfrak{W}, u, \beta\rangle \vDash [\alpha]\mathsf{p}_0$. Alles zusammen haben wir $\langle \mathfrak{W}, v, \beta\rangle \vDash [\beta](\mathsf{p}_0 \to [\alpha]\mathsf{p}_0)$. Aber es ist $\langle \mathfrak{W}, w, \beta\rangle \nvDash \mathsf{p}_0$, also $\langle \mathfrak{W}, v, \beta\rangle \nvDash [\beta]\mathsf{p}_0$. Dies zeigt die erste Behauptung. Nun nehmen wir an, $R(\beta) = R(\alpha)^*$. Ferner nehmen wir an, $\langle \mathfrak{W}, v, \beta\rangle \vDash \mathsf{p}_0; [\beta](\mathsf{p}_0 \to [\alpha]\mathsf{p}_0)$. Dann gilt für all $n \in \omega$: ist $v \overset{\alpha^n}{\to} w$, so $\langle \mathfrak{W}, v, \beta\rangle \vDash \mathsf{p}_0 \to [\alpha]\mathsf{p}_0$, woraus eine Induktion ihrerseits liefert dass $\langle \mathfrak{W}, v, \beta\rangle \vDash \mathsf{p}_0$. Dies zeigt mit $R(\beta) = R(\alpha)^*$, dass $\langle \mathfrak{W}, v, \beta\rangle \vDash [\beta]\mathsf{p}_0$. Q. E. D.

Man sagt, ein Programm habe die Konfluenzzeigenschaft, falls jede (iterierte) Berechnung des Programms von einem gegebenen Zustand in stets denselben Endzustand läuft.

Theorem 207 *Es sei $\langle W, R\rangle$ eine endliche Kripke–Struktur. $R(\alpha)$ ist genau dann konfluent, wenn*

$$\langle W, R\rangle \models \langle \alpha^*\rangle[\alpha^*]\mathsf{p}_0 \to [\alpha^*]\langle \alpha^*\rangle \mathsf{p}_0$$

Beweis. Es sei $R(\alpha)$ konfluent, β eine Belegung und $w \in W$. Angenommen, $\langle W, R, w, \beta\rangle \vDash \langle \alpha^*\rangle[\alpha^*]\mathsf{p}_0$. Es sei v derjenige Zustand, in den jede Berechungen von α von v aus enden wird. Dann gilt $u\ R(\alpha^*)\ v$ für jeden Punkt u

mit $w\ R(\alpha^*)\ u$. Also haben wir $\langle W, R, v, \beta\rangle \vDash \mathrm{p}_0$, da es einen Punkt u gibt mit $w\ R(\alpha^*)\ u$, welcher $[\alpha^*]\mathrm{p}_0$ erfüllt. Sei nun u' ein beliebiger Punkt mit $w\ R(\alpha^*)\ u'$. Da wegen der Konfluenz nun $u'\ R(\alpha^*)\ v$ ist, gilt $\langle\alpha^*\rangle\mathrm{p}_0$ in u'. Da u' beliebig war, gilt $[\alpha^*]\langle\alpha^*\rangle\mathrm{p}_0$ bei w, wie gewünscht. Sei nun $R(\alpha)$ nicht konfluent. Dann existiert ein w und v_0, v_1 derart, dass eine iterierte Berechung von $R(\alpha)$ von u nach v_0 wie auch v_1 existiert, aber keine Berechnung von v_0 nach v_1 sowie keine von v_1 nach v_0. Setze $\beta(\mathrm{p}_0) := \{v_0\}$. Es gilt $\langle W, R, w, \beta\rangle \vDash \langle\alpha^*\rangle[\alpha^*]\mathrm{p}_0$ (wähle etwa als $R(\alpha^*)$–Nachfolger v_0). Es gilt aber nicht $\langle W, R, w, \beta\rangle \vDash [\alpha^*]\langle\alpha^*\rangle\mathrm{p}_0$ (wähle als Gegenbeispiel v_1 als Nachfolger). Q. E. D.

Axiomatisch kann man neben Eigenschaften von einzelnen Programmen auch Interaktionen definieren. Wir wollen dies nicht weiter vertiefen. Wichtige Varianten von PDL sind die sogenannte **deterministische PDL** (DPDL), welche zusätzlich für alle Basisprogramme α die Axiome $\langle\alpha\rangle\mathrm{p}_0 \to [\alpha]\mathrm{p}_0$ hat. Ferner die **PDL mit Konversen** ($\mathsf{PDL}^{\smile}$), welche eine Operation $^{\smile}$ auf den Programmen besitzt, für die wir verlangen, dass $R(\alpha^{\smile}) = R(\alpha)^{\smile}$. Dies kann man axiomatisch fassen durch die Aussagen

$$\mathrm{p}_0 \to [\alpha]\langle\alpha^{\smile}\rangle\mathrm{p}_0, \qquad \mathrm{p}_0 \to [\alpha^{\smile}]\langle\alpha\rangle\mathrm{p}_0 \,.$$

Schließlich wollen wir noch erwähnen, dass es auch eine Prädikatenlogische Version gibt. Hier tauscht man die Aussagenlogik durch Prädiktenlogik ohne Quantoren aus. Strukturen entstehen aus den Strukturen der Prädikatenlogik wie folgt. Die Menge der Welten ist die Mange der Belegungen, das heißt, der Funktionen von Variablen in eine die Struktur $\mathfrak{M}$. Ferner hat man für jede Variable x ein Programm $x \leftarrow ?$, welches auch **random assignment** heißt. Seine Aktion ist die Zuweisung eines beliebigen Werts aus $\mathfrak{M}$ zu x. Es gilt also $\beta \overset{x\leftarrow?}{\to} \beta'$ genau dann, wenn β und β' sich höchstens in x unterscheiden. Dies ist genau dann der Fall, wenn $\beta \sim_x \beta'$. Die Quantoren kann man dann wie folgt definieren:

$$\begin{array}{lcl} (\forall x)\varphi & := & [x \leftarrow ?]\varphi \\ (\exists x)\varphi & := & \langle x \leftarrow ?\rangle\varphi \end{array}$$

Das Programm $x := 5$ kann man wie folgt zusammensetzen:

$$x \leftarrow ?; (x \doteq 5)?$$

Es ist leicht zu sehen, dass man genau dann von β nach β' mittels $x \leftarrow ?; (x \doteq 5)?$ kommt, wenn β und β' sich höchstens in x unterscheiden, und $\beta'(x) = 5$.