Kapitel 7

Relationen

Ω bezeichne die Menge aller Aussagen.

7.1 Grundbegriffe

7.1.1 Definition. Sei $n\colon \mathbb{N}$, und $X_1, \ldots, X_n$ Datentypen. Dann heißt jede Konstruktion P vom Typ

$$P\colon X_1 \to \cdots \to X_n \to \Omega$$

ein *n-stelliges Prädikat*.

7.1.2 Beispiel. Die Eigenschaft, eine Primzahl zu sein, ist ein 1-stelliges Prädikat vom Typ $\mathbb{N} \to \Omega$. Dabei wird z.B. der Zahl 3 die Aussage „3 ist eine Primzahl" zugeordnet, während der Zahl 4 die Eigenschaft „4 ist eine Primzahl" zugeordnet wird. Eng assoziert mit diesem Prädikat ist die Teilmenge aller natürlichen Zahlen, welche diese Eigenschaft haben (d.h. welchen eine wahre Aussage zugeordnet wird), also die Menge aller Primzahlen.

7.1.3 Bemerkung. Einstellige Prädikate (also Konstruktionen vom Typ $X \to \Omega$) nennt man auch Eigenschaften. Die Menge aller Eigenschaften auf der Menge X entspricht gerade der Potenzmenge von X. Dabei wird jede Eigenschaft P mit der Menge $\{\, x\colon X \mid P(x) \,\}$ identifiziert, und umgekehrt jede Teilmenge $A\colon \mathscr{P}X$ mit der Eigenschaft, Element von A zu sein.

7.1.4 Beispiel. Sei L die Menge aller Lehrveranstaltungen (einer Universität), M die Menge aller Studenten, und S die Menge aller Semester. Das 3-stellige Prädikat vom Typ $L \to M \to S \to \Omega$, welches jedem $l\colon L$, $m\colon M$, $s\colon S$ die Aussage „Der Student m nimmt im Semester s and der Lehrveranstaltung l teil" zuordnet, wird typischerweise durch ein Tabelle definiert, welche alle Fälle explizit auflistet, welchen eine wahre Aussage entspricht. Sie könnte etwa folgendermaßen beginnen.

Lehrveranstaltung	Student	Semester
Mathematik und Logik	Eike Eifrig	2008W
Mathematik und Logik	Fausto Faul	2008W
Formale Grundlagen	Eike Eifrig	2009S
Mathematik und Logik	Fausto Faul	2009W
Mathematik und Logik	Fabian Famos	2009W
Formale Grundlagen	Fabian Famos	2009S
Datenmodellierung	Eike Eifrig	2008S

In der Mathematik sind die 2-stelligen Prädikate besonders beliebt. Diese beschreiben Beziehungen zwischen den Elementen zweier Mengen, oder zwischen den Elementen einer Menge.

7.1.5 Definition. Eine *Relation von X nach Y* ist ein 2-stelliges Prädikat vom Typ $X \to Y \to \Omega$. Eine *Relation in X* ist ein 2-stelliges Prädikat vom Typ $X \to X \to \Omega$.

7.1.6 Bemerkung. Die Bezeichnungsweise ist hier nicht ganz einheitlich. So werden z.B. gerade im Zusammenhang mit relationalen Datenbanken auch Prädikate beliebiger Stelligkeit als Relationen bezeichnet. Die 2-stelligen Relationen heißen dann zur Unterscheidung *binäre Relationen*.

7.1.7 Beispiel. Die bekannteste Relation ist die Gleichheitsrelation. Eine solche gibt es für jede Menge. Je zwei Elementen $x, y\colon X$ einer Menge X wird die Aussage zugeordnet, daß diese beiden gleich sind, d.h. die Aussage $x = y$.

7.1.8 Beispiel. Für jede der Mengen $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ wird eine natürliche Ordnung definiert durch die Relation, welche je zwei Elementen x, y die Aussage $x \leq y$ zuordnet.

7.1.9 Beispiel. Für $\mathbb{Z}$ kennen wir auch die Teilbarkeitsrelation, welche zwei Elementen x, y die Aussage $\bigvee_{z\colon \mathbb{Z}} y = x \cdot z$ (abgekürzt: $x \mid y$) zuordnet.

7.1.10 Beispiel. Eine weitere bekannte Relation in $\mathbb{Z}$ ordnet zwei Elementen $x, y\colon \mathbb{Z}$ die Aussage „x und y sind teilerfremd“ zu.

7.1.11 Beispiel. Auch die Kongruenz modulo m ($\equiv_m$) ist für jedes $m\colon \mathbb{N}$ eine wichtige Relation in $\mathbb{Z}$.

7.1.12 Beispiel. Eine wohlbekannte Relation in der Potenzmenge einer jeden Menge ist die Teilmengen-Relation.

7.1.13 Beispiel. Die Enthaltensein-Relation ordnet einem Element $x\colon X$ einer Menge X und einer Teilmenge $A\colon \mathscr{P}X$ die Aussage „x ist Element von A“ (kurz $x \in A$) zu. Sie hat also den Typ $X \to \mathscr{P}X \to \Omega$.

7.1.14 Bemerkung. Ist R eine (binäre) Relation, so schreibt man statt Rxy meist $x\,R\,y$ oder $x \xrightarrow{R} y$; oder einfach $x \to y$, falls R aus dem Zusammenhang ersichtlich ist. Man sagt dann auch, x und y sind in Relation. Oft werden auch diverse spezielle Symbole verwendet (z.B.: $<$, $\leq$, $\subseteq$, $\in$, $\equiv$, $\ldots$).

Da mittels Currying $X \to Y \to \Omega$ praktisch dasselbe ist wie $X \times Y \to \Omega$, was wiederum nichts anderes ist als $\mathscr{P}(X \times Y)$, kann man eine Relation auch genausogut als Teilmenge von $X \times Y$ betrachten. Man schreibt daher auch gerne $(x, y) \in R$. Mit dieser Betrachtungsweise kann man zwei Relationen auch so wie Teilmengen vergleichen; es gilt dann

$$R \subseteq S \iff \bigwedge_{x,y\colon X} \bigl(x \xrightarrow{R} y \implies x \xrightarrow{S} y\bigr).$$

Auch die Vereinigung und der Durchschnitt von Relationen sind damit sinnvoll definiert. Die leere Menge heißt in diesem Zusammenhang dann auch die *leere Relation*, und $X \times Y$ die *Allrelation*.

Analoges gilt für mehrstellige Relationen.

In der Informatik (insbesondere im Zusammenhang mit relationen Datenbanken) werden Relationen auch gerne durch expliziten Aufzählen (in Form einer Tabelle) aller in Beziehung stehender Elemente definiert.

7.1.15 Beispiel. Sei Z (Zuteilung) die Relation, welche einem Mitarbeiter x und einer Abteilung a die Aussage „x arbeitet für die Abteilung a“ zuordnet. Für eine bestimmte Firma könnte diese Relation etwa folgendermaßen aussehen:

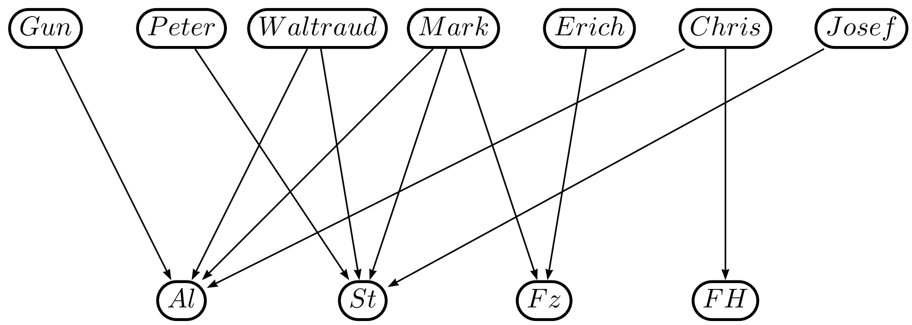
Mitarbeiter	Abteilung
Gun	Al
Peter	St
Waltraud	St
Waltraud	Al
Mark	Al
Mark	St
Erich	Fz
Mark	Fz
Chris	FH
Chris	Al
Josef	St

Diese Tabelle bedeutet, daß z.B. Peter für die Abteilung St arbeitet, d.h. den Elementen Peter (aus der Menge aller Mitarbeiter) und Al (aus der Menge aller Abteilungen) wird eine wahre Aussage zugeordnet. Allen nicht aufgezählten Kombinationen dagegen wird eine falsche Aussage zugeordnet. Dies bedeutet z.B., daß Waltraud nicht für die Abteilung Fz arbeitet, wohl aber für Al, und auch für St.

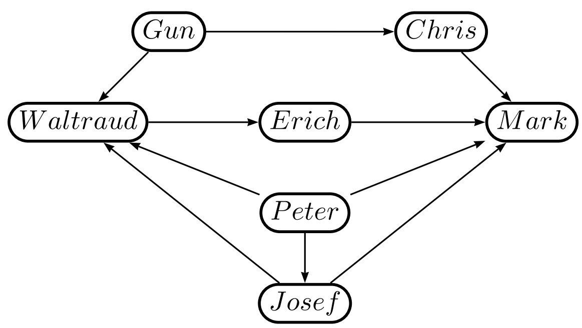
Man beachte, daß die Reihenfolge der Zeilen in der Tabelle für die betrachtete Relation keine Rolle spielt. Dasselbe gilt für Vielfachheiten.

7.1.16 Bemerkung. Da Datenbanken typischerweise nur endlich große Tabellen speichern, entsprechen durch solche Tabellen definierte Relationen gerade den endlichen Teilmengen des entsprechenden kartesischen Produkts.

7.1.17 Beispiel. Anstelle durch eine Tabelle kann man derartige Relationen auch durch eine Graphen darstellen:



Da es sich hier um eine Relation *von* der Menge der Mitarbeiter *in* die Menge der Abteilungen handelt, zeigen in diesem Graphen alle Pfeile von der einen in die andere Menge. Dies sieht etwas anderes aus, wenn wir eine Relation *in* einer Menge betrachten. Sei z.B. D (Delegiert) die Relation „x delegiert Arbeit an y“, welche etwa so aussehen könnte:



Als Tabelle geschrieben könnte dies etwas so aussehen:

Chef	Arbeiter
Gun	Chris
Gun	Waltraud
Chris	Mark
Waltraud	Erich
Erich	Mark
Peter	Waltraud
Peter	Mark
Peter	Josef
Josef	Waltraud
Josef	Mark

7.1.18 Beispiel. Ein weiteres Beispiel für eine wichtige Relation aus der Informatik ist die Input-Output-Relation eines Computerprogramms. Sei etwa P ein

Programm, x ein Zustand des Systems unmittelbar vor Ablauf des Programms, und y ein Zustand, in dem sich das System unmittelbar nach Ablauf des Programms befinden kann, wenn es zuvor im Zustand x war; symbolisch: $x \overset{P}{\to} y$. Man beachte, daß y nicht unbedingt durch x und P eindeutig bestimmt sein muß (Programme können indeterministisch ablaufen).

7.1.19 Beispiel. Relationen in $\mathbb{R}$ stellt man sich am besten geometrisch vor. So entspricht z.B. der Relation, welche jedem $x\colon \mathbb{R}$ und $y\colon \mathbb{R}$ die Aussage $x^2+y^2 = 9$ zuordnet, die Menge $\{\, (x, y)\colon \mathbb{R}^2 \mid x^2 + y^2 = 9 \,\}$, welche man sich am besten als Kreis mit Radius 3 vorstellt.

7.1.20 Beispiel. Auch Relationen in $\mathbb{Z}$ (z.B. Teilbarkeit) kann man sich geometrisch vorstellen, wenn man nur Gitterpunkte (diejenigen mit ganzzahligen Koordinaten) betrachtet.

7.2 Relationenprodukt

Sei R und S Relationen, und $x \overset{R}{\to} y$ sowie $y \overset{S}{\to} z$. Man stelle sich etwa R und S als Programme vor. Wir überlegen uns, was passiert, wenn wir diese hintereinander laufen lassen, also zuerst R, und dann S. Dieses zusammengesetzte Programm bezeichnen wir mit $R; S$. Wenn sich das System nun im Zustand x befindet und wir lassen R laufen, dann kann es sich danach im Zustand y befinden. Wenn wir nun weiters S laufen lassen, dann kann es sich danach somit im Zustand z befinden. Das heißt: $x \overset{R;S}{\to} z$. Man beachte, daß dafür ein einziges derartiges y (quasi ein Vermittler) ausreicht.

7.2.1 Definition. Gegeben seien die Relationen $R\colon X \to Y \to \Omega$ und $S\colon Y \to Z \to \Omega$. Dann wird das Relationenprodukt $(R; S)$ definiert durch

$$x \overset{R;S}{\to} z \iff \bigvee_{y\colon Y} x \overset{R}{\to} y \wedge y \overset{S}{\to} z.$$

7.2.2 Bemerkung. *Vorsicht!* Statt der Notation $R; S$ sind auch $R \circ S$ und $S \circ R$ sowie RS und SR gebräuchlich. Hier ist besondere Vorsicht geboten, zumal das Relationenprodukt nicht kommutativ ist.

7.2.3 Satz. *Das Relationenprodukt ist assoziativ.*

Beweis. Übung. □

7.2.4 Beispiel. Seien etwa Z und D die Relationen aus den Beispielen 7.1.15 und 7.1.17. Dann bedeuted die Aussage $m \overset{D;Z}{\to} a$, daß der Mitarbeiter m an zumindest einen Mitarbeiter der Abteilung a Arbeit delegiert.

7.2.5 Definition. In jeder Menge X definiert die Gleichheit eine als *Diagonale* Δ_X bezeichnete Relation, durch

$$x \,\Delta_X\, y \iff x = y.$$

7.2.6 Satz. *Für jede Relation $R\colon X \to Y$ gilt*

$$R; \Delta_Y = R = \Delta_X; R.$$

7.2.7 Definition. Zu jeder Relation R von X nach Y definiert man die zu R *inverse* Relation durch

$$x \overset{R^{-1}}{\to} y \iff y \overset{R}{\to} x.$$

7.2.8 Beispiel. Die zur Kleiner-Relation inverse Relation ist die Größer-Relation (d.h. $<^{-1} = >$), denn $x < y$ gilt genau dann wenn $y > x$. Ähnliches gilt z.B. für die Teilmengenbeziehung. Hingegen sind die Gleichheitsrelation (bzw. Diagonale) und die Kongruenz modulo m zu sich selbst invers.

Für die inverse Relation einer durch einen Graphen dargestellten Relation sind einfach alle Pfeile umzudrehen.

Für die Tabelle der inversen Relation sind einfach die Spalten zu vertauschen.

Für die geometrische Darstellung der inversen Relation in der x-y-Ebene sind die Koordinaten zu vertauschen, d.h. die geometrische Figur wird an der Hauptdiagonalen gespiegelt.

7.2.9 Satz. *Die Menge der Relationen in einer Menge X, d.h. die Menge $X \to X \to \Omega$, bildet zusammen mit dem Relationenprodukt und der Diagonale Δ_X als neutralem Element ein Monoid.*

7.2.10 Bemerkung. Es ergibt sich jedoch keine Gruppe, weil im allgemeinen $R; R^{-1} = \Delta_X$ nicht gilt.

7.2.11 Definition. So wie für assoziative Operationen üblich, verwenden wir auch für Relationen die Notation mit Potenzen:

$$\begin{aligned} R^1 &= R \\ R^2 &= R; R \\ R^{n+1} &= R^n; R = R; R^n \\ R^0 &= \Delta_X \\ R^{-n} &= (R^{-1})^n \end{aligned}$$

7.2.12 Satz. *Sei R eine Relation von X nach Y. Es gelten für alle $n, m \colon \mathbb{N}$:*

$$\begin{aligned} R^n; R^m = R^{n+m} &= R^m; R^n, \\ (R^n)^m = R^{n \cdot m} &= (R^m)^n. \end{aligned}$$

Und für Relationen R, S in X gilt

$$(R; S)^{-1} = S^{-1}; R^{-1}$$

Die Relationen R, S sind genau dann vertauschbar *(d.h. $R; S = S; R$) wenn für alle $n \colon \mathbb{Z}$ gilt*

$$(R; S)^n = R^n; S^n,$$

7.3 Äquivalenzrelationen

7.3.1 DEFINITION. Sei X ein Datentyp. Dann heißt eine Relation R in X

$$\begin{array}{lll} \textit{reflexiv} & \text{falls} & \bigwedge_{x\colon X} x \xrightarrow{R} x; \\ \textit{symmetrisch} & \text{falls} & \bigwedge_{x,y\colon X} \big(x \xrightarrow{R} y \iff y \xrightarrow{R} x\big); \\ \textit{transitiv} & \text{falls} & \bigwedge_{x,y,z\colon X} \big(x \xrightarrow{R} y \wedge y \xrightarrow{R} z \implies x \xrightarrow{R} z\big). \end{array}$$

7.3.2 BEMERKUNG. Für Äquivalenzrelationen verwendet man gerne Symbole wie $=, \equiv, \cong, \sim, \simeq, \triangleq, \doteq, \iff$.

7.3.3 BEMERKUNG. Auf jeder Menge ist die Diagonale Δ_X eine Äquivalenzrelationen.

7.3.4 SATZ. *Ist $\equiv$ eine Äquivalenzrelation in Y und $f\colon X \to Y$, dann wird durch*

$$x \equiv_f y \iff f(x) \equiv f(y)$$

eine Äquivalenzrelation in X definiert.

7.3.5 SATZ. *Eine Relation R in einer Menge X ist*

$$\begin{array}{ll} \textit{reflexiv} & \iff R \supseteq R^0; \\ \textit{symmetrisch} & \iff R = R^{-1}; \\ \textit{transitiv} & \iff R \supseteq R^2. \end{array}$$

7.3.6 DEFINITION. Unter der *reflexiven*, *symmetrischen* oder *transitiven Hülle* einer Relation R in X versteht man die kleinste R umfassende Relation in X, welche die entsprechende Eigenschaft hat.

7.3.7 SATZ. *Sei R eine Relation in einer Menge X. Dann gelten*

$$\begin{aligned} \textit{reflexive Hülle von } R &= R^0 \cup R; \\ \textit{symmetrische Hülle von } R &= R \cup R^{-1}; \\ \textit{transitive Hülle von } R &= \bigcup_{k=1}^{\infty} R^k =: R^+ \\ \textit{reflexiv-transitive Hülle von } R &= \bigcup_{k=0}^{\infty} R^k = R^0 \cup R^+ =: R^* \end{aligned}$$

7.3.8 SATZ. *Ist R reflexiv bzw. symmetrisch, so gilt dies auch für deren transitive Hülle R^+, und natürlich auch für R^*. Insbesondere ist daher $(R \cup R^{-1})^*$ stets eine Äquivalenzrelation. Sie heißt die von R* erzeugte Äquivalenzrelation*, weil sie die kleinste ist, die R umfaßt.*

7.3.9 BEMERKUNG. Man beachte jedoch, daß die symmetrische Hülle einer transitiven Relation nicht transitiv sein muß, d.h. es gilt nur $R^+ \cup (R^+)^{-1} \subseteq (R \cup R^{-1})^+$.

7.3.10 Definition. Eine Teilmenge $\mathcal{P}$ von $\mathscr{P}X$ nennt man eine *Partition* von X falls

1. alle Elemente von $\mathcal{P}$ mindestens ein Element enthalten;
2. die Vereinigung aller Elemente von $\mathcal{P}$ die Grundmenge X ergibt;
3. je zwei Elemente von $\mathcal{P}$ leeren Durchschnitt haben.

7.3.11 Definition. Sei R eine Äquivalenzrelation in X und $x\colon X$. Dann heißt

$$[x]_R := \{y \mid x \xrightarrow{R} y\}$$

die Äquivalenzklasse von x bezüglich R.

Die Menge $X/_R = \{[x]_R \mid x\colon X\}$ aller Äquivalenzklassen von R heißt Faktormenge. Dadurch wird jeder Relation auf natürliche Weise eine Partition zugeordnet.

Umgekehrt wird jeder Partition mit

$$x \equiv_{\mathcal{P}} y \iff \bigvee_{A\in\mathcal{P}} x \in A \wedge y \in A$$

auf ebenso natürliche Weise eine Äquivalenzrelation zugeordnet.

7.3.12 Bemerkung. Aus den obigen Konstruktionen erkennt man, daß Äquivalenzrelationen und Partitionen „eigentlich dasselbe" sind.

Die Faktormenge $X/_R$ kann man auch als diejenige Menge auffassen, welche dieselben Elemente hat wie X, aber die Äquivalenzrelation R als Gleichheitsbegriff.

7.4 Ordnungsrelationen

7.4.1 Definition. Eine Relation in X heißt *Quasi-Ordung* falls sie reflexiv und transitiv ist.

7.4.2 Bemerkung. Für (Quasi-)Ordnungen verwendet man gerne Symbole wie $\leq, \subseteq, \sqsubseteq, \preceq, \geq$. Dabei bedeutet dann üblicherweise z.B. $x \leq y$ dasselbe wie $y \geq x$.

Der englische Begriff quasi order steht bei manchen Autoren nicht für Quasiordnung sondern für eine transitive und irreflexive Relation.

7.4.3 Satz. *Sei $\sqsubseteq$ eine Quasiordnung in X. Dann wird durch*

$$x \equiv y \iff x \sqsubseteq y \wedge y \sqsubseteq x$$

eine Äquivalenzrelation definiert.

7.4.4 Definition. Eine Relation R in einer Menge X heißt *antisymmetrisch* falls

$$\bigwedge_{x,y\colon X} \big(x \xrightarrow{R} y \wedge y \xrightarrow{R} x \implies x = y\big).$$

Eine antisymmetrische Quasiordnung heißt *partielle Ordnung* oder einfach *Ordnung*.

7.4.5 Satz. *Jede Quasiordnung $\sqsubseteq$ in X kann als Ordnung in $X/_{\equiv}$ aufgefaßt werden.*

7.4.6 Definition. Ein Ordnung $\sqsubseteq$ in X heißt *linear* falls

$$\bigwedge_{x,y\colon X} x \sqsubseteq y \vee y \sqsubseteq x.$$

Im allgemeinen gibt es zu einer partiellen Ordnung mehrere lineare Ordnungen, welche sie umfassen. Das Problem, zu einer partiellen Ordung $\sqsubseteq$ irgendeine diese umfassende lineare Ordnung $\leq$ mit $(\sqsubseteq) \subseteq (\leq)$ zu finden, heißt *topologisches Sortieren.*

7.5 Funktionale Relationen

Im allgemeinen kann es zu einem x mehrere y geben, sodaß $x \to y$. Bedeutet $\to$ etwa die Input/Output-Relation eines Computer-Programmes, so heißt dies, daß der Output nicht notwendigerweise deterministisch vom Input abhängt.

7.5.1 Definition. Eine Relation p von X nach Y heißt *deterministisch*, *wohldefiniert*, *funktional* oder eine *partielle Funktion* falls gilt

$$\bigwedge_{x\colon X} \bigwedge_{y,z\colon Y} \big(x \xrightarrow{p} y \wedge x \xrightarrow{p} z \implies y = z\big).$$

Sie heißt *terminierend* oder *total*, falls gilt

$$\bigwedge_{x\colon X} \bigvee_{y\colon Y} x \xrightarrow{p} y.$$

Erfüllt sie beide Eigenschaften, so nennt man sie eine *(totale) Funktion.*

7.5.2 Bemerkung. Für eine deterministische Relation p gibt es daher zu jedem x höchstens ein y mit $x \xrightarrow{p} y$, und man schreibt dann auch $y = px$ oder $y = p(x)$. Es ist aber zu beachten, daß es so ein y möglicherweise gar nicht gibt. Bei einer Input/Output-Relation entspricht das dem Fall, daß das Programm nie stehen bleibt (oder abstürzt, was mathematisch gesehen dasselbe ist). Die Termination dagegen garantiert, daß das Programm tatsächlich zu jedem gültigen Input irgendwann auch einen gültigen Output liefert.

7.5.3 Bemerkung. Der hier neu eingeführte Funktionsbegriff stimmt mit dem bekannten überein, wenn man eine Funktion $f\colon X \to Y$ mit der Relation

$$x \xrightarrow{f} y :\iff f(x) = y$$

von X nach Y identifiziert. Die Teminations-Eigenschaft besagt gerade, daß zu jedem x ein y zu konstruieren ist, und die Wohldefiniertheit ergibt sich daraus, weil diese Konstruktion deterministisch sein muß. Auch die Hintereinanderausführung von Funktionen entspricht genau dem Relationenprodukt, wobei allerdings auf die Reichenfolge zu achten ist: $g \circ f = f; g$. Ebenso ist die Umkehrfunktion eine Spezialfall der inversen Relation.

7.5.4 Bemerkung. Jede Relation $R\colon X \to Y \to \Omega$ läßt sich als Funktion von Typ $X \to \mathscr{P}Y$ auffassen. $R(x)$ bezeichnet dann die Menge aller möglichen Outputs, also $R(x) = \{\, y \mid x \xrightarrow{R} y \,\}$.

7.5.5 Definition. Eine Funktion $f\colon X \to Y$ heißt *injektiv* falls

$$\bigwedge_{x,y\colon X} f(x) = f(y) \implies x = y;$$

sie heißt *surjektiv* falls

$$\bigwedge_{y\colon Y} \bigvee_{x\colon X} fx = y;$$

und *bijektiv* falls sie injektiv und surjektiv ist.

7.5.6 Bemerkung. Diese Eigenschaften sind dual zu den definierenden Eigenschaften für Funktionen. Die Injektivität bedeutet gerade, daß auch f^{-1} deterministisch ist, die Surjektivität bedeutet daß f^{-1} terminiert, und die Bijektivität, daß auch f^{-1} wieder eine Funktion ist. Insbesondere gilt für bijektive Funktionen

$$f^{-1} \circ f = \mathrm{id}_X \qquad \text{und} \qquad f \circ f^{-1} = \mathrm{id}_Y\,.$$

7.6 Relationenalgebra

In relationalen Datenbanken wird mit Relationen gerechnet. Da Relationen Elemente einer Potenzmenge sind, bieten sich als grundlegende Rechenoperationen zuerst einmal alle Grundoperationen für Teilmengen an, d.h. Durchschnitt, Vereinigung und Mengen-Differenz. Dazu kommen im wesentlichen die folgenden Grundoperationen: Restriktion, Projektion, Produkt, Division.

Die in Datenbanken verwendeten Relationen weisen ein paar formale Unterschiede zu den bisher besprochenen auf. Sie werden üblicherweise durch Tabellen dargestellt und sind daher immer endlich. Da es kein Problem ist, Tabellen mit mehr oder weniger als 2 Spalten zu verwenden, betrachten man hier Relationen beliebiger Stelligkeit. Ferner werden die Spalten nicht einfach durchnumeriert, sondern erhalten Namen (Spaltenüberschriften). Damit ist nicht nur die Reihenfolge der Zeilen irrelevant sondern auch die Reihenfolge der Spalten. Die Zeilen in den Tabellen nennt man dann Tupel.

7.6.1 Definition. Sei U eine Menge von Spaltenüberschriften. Ein *Relationenschema* σ ordnet jeder Überschrift $u\colon U$ einen Datentyp zu. Ein zu σ passendes *Tupel* ordnet jedem $u\colon U$ ein Element vom Typ $\sigma(u)$ zu. Eine *zu σ passende Relation* (im Sinne der Relationenalgebra) besteht dann aus einer (üblicherweise endlichen) Menge von derartigen Tupeln.

7.6.2 Bemerkung. In praktischen Anwendungen kann ein Relationenschema mehr als nur die Datentypen der verschiedenen Spalten festlegen (z.B. Beziehungen zwischen den Spalten und zu anderen Relationen). Darauf wollen wir hier allerdings nicht näher eingehen.

7.6.3 BEMERKUNG. Da Relationen gemäß Definition eine Menge von Tupeln sind, ist klar, wie Durchschnitt, Vereinigung und Mengendifferenz von Relationen (vom selben Schema!) zu bilden sind.

7.6.4 DEFINITION. Eine *Restriktion* (auch *Selektion*) wählt gewisse Tupel aus einer Relation (d.h. Zeilen einer Tabelle) aufgrund eines Kriteriums aus. Bezeichnungsweise: σ_K, wobei K das Kriterium ist.

7.6.5 DEFINITION. Eine *Projektion* wählt aus jedem Tupel einer Relation gewisse Komponenten aus (d.h. Spalten einer Tabelle). Bezeichnungsweise: π_U, wobei U eine Liste von Spaltenüberschriften ist.

7.6.6 DEFINITION. Beim *kartesischen Produkt* zweier Relationen R, S wird jedes Tupel der einen Relation mit jedem Tupel der anderen Relation kombiniert und aus diesen eine neue Relation $R \times S$ gebildet.

7.6.7 BEISPIEL. Wir bilden das kartesischen Produkt $R \times S$ der folgende Relationen R und S:

$$\begin{array}{c|c} \mathrm{A} & \mathrm{B} \\ \hline \mathrm{a} & 1 \\ \mathrm{a} & 2 \\ \mathrm{b} & 1 \end{array} \quad \times \quad \begin{array}{c|c} \mathrm{C} & \mathrm{D} \\ \hline 2 & \mathrm{x} \\ 1 & \mathrm{y} \end{array} \quad = \quad \begin{array}{c|c|c|c} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{a} & 1 & 2 & \mathrm{x} \\ \mathrm{a} & 2 & 2 & \mathrm{x} \\ \mathrm{b} & 1 & 2 & \mathrm{x} \\ \mathrm{a} & 1 & 1 & \mathrm{y} \\ \mathrm{a} & 2 & 1 & \mathrm{y} \\ \mathrm{b} & 1 & 1 & \mathrm{y} \end{array}$$

7.6.8 BEMERKUNG. Das kartesische Produkt wird meist mit einer Restriktion und einer Projektion kombiniert, wodurch verschiedene Arten von Verbindungen („Join") entstehen.

7.6.9 BEISPIEL. In Forsetzung des vorigen Beispiels wählen wir nun all jene Zeilen aus, bei denen die Spalten B und C übereinstimmen (Restriktion mit B=C). Dies ergibt zuerst:

$$\sigma_{B=C}(R \times S) \;=\; \begin{array}{c|c|c|c} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \hline \mathrm{a} & 2 & 2 & \mathrm{x} \\ \mathrm{a} & 1 & 1 & \mathrm{y} \\ \mathrm{b} & 1 & 1 & \mathrm{y} \end{array}$$

Klarerweise ist es unnötig beide Spalten (B und C), die ohnehin gleich sind, zu verwenden. Wir können etwa die Spalte C weglassen, mit einer Projektion auf A,B,D. Dies ergibt:

$$\pi_{A,B,D}(\sigma_{B=C}(R \times S)) \;=\; \begin{array}{c|c|c} \mathrm{A} & \mathrm{B} & \mathrm{D} \\ \hline \mathrm{a} & 2 & \mathrm{x} \\ \mathrm{a} & 1 & \mathrm{y} \\ \mathrm{b} & 1 & \mathrm{y} \end{array}$$

Oft ist man in solchen Fällen an der vermittelnden Spalte gar nicht interessiert und führt eine Projektion auf A,D durch. Dies führt zum ganz normalen Relationenprodukt $(R;S)$:

$$\pi_{A,D}(\sigma_{B=C}(R \times S)) \;=\; \begin{array}{c|c} \text{A} & \text{D} \\ \hline \text{a} & \text{x} \\ \text{a} & \text{y} \\ \text{b} & \text{y} \end{array} \;=\; R;S$$

7.6.10 Bemerkung. Auf eine Erläuterung der (etwas komplizierten und in der Literatur uneinheitlich beschriebenen) klassischen Grundoperation der Division sowie diverse modernere Erweiterungen sei hier verzichtet.

7.6.11 Bemerkung. Ein SQL-Abfragebefehl ist im einfachsten Fall typischerweise wie folgt aufgebaut:

```
select distinct Spaltennamen
  from Relationen
  where Bedingungen
```

Dabei definieren die Relationen nach `from` die Relationen aus denen zuerst ein kartesisches Produkt gebildet wird, mit den Bedingungen nach `where` wird dann ein Restriktion durchgeführt, und von dem ganzen dann gemäß den Spaltennamen nach `select` eine Projektion. Derartige Abfragen können dann noch mit `union`, `intersect` und `except` verbunden werden, um die mengentheoretischen Operationen (Vereinigung, Durchschnitt, Differenz) durchzuführen. Denselben Effekt erzielt man auch, indem man mehrere Bedingungen nach `where` mit `or`, `and`, `not` verbindet.

7.6.12 Beispiel. Das Relationenprodukt $R;S = \pi_{A,D}(\sigma_{B=C}(R \times S))$ im obigen Beispiel läßt sich folgendermaßen mit SQL realisieren:

```
select distinct A, D
  from R, S
  where B = C
```

7.6.13 Bemerkung. Das Angenehme an der Relationenalgebra ist, daß man mit Relationen ähnlich wie mit Zahlen oder Termen rechnen kann. Und ähnlich wie für diese, gelten diverse Rechenregeln (analog zu $3 \cdot 4 = 12$ oder $(x+y)(x-y) = x^2 - y^2$), z.B. $\pi_U(R \cup S) = \pi_U(R) \cup \pi_U(S)$. Und so wie ein Computer rechnen oder algebraische Terme vereinfachen kann, kann er somit auch Datenbankabfragen vereinfachen, wobei „vereinfachen“ in letzterem Fall bedeutet, daß der Zugriff möglichst effizient durchgeführt werden kann. Somit muß ein Datenbankbenutzer überhaupt keinen Gedanken daran verschwenden, wie eine Abfrage möglichst günstig zu stellen ist. Stattdessen beschreibt er die gewünschte Relation einfach so, wie es ihm logisch am einfachsten erscheint, und das Datenbankverwaltungssystem berechnet dann daraus die effizienteste Möglichkeit, die Daten auch wirklich zu finden, gerade so, wie ein Taschenrechner Rechenaufgaben löst.