Kapitel 1

Mathematische Objekte

Dieses Kapitel beschreibt grundlegende mathematische Objekte wie Mengen, Tupel, Relationen und Funktionen und führt die entsprechenden Begriffe und Notationen ein.

1.1 Grundbegriffe

Die Sprache der Mathematik geht davon aus, dass es *mathematische Objekte* gibt. Die *natürlichen Zahlen*

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

sind Beispiele für mathematische Objekte. Statt von mathematischen Objekten sprechen wir im Folgenden einfach von *Objekten*.

Objekte können zu *Mengen* zusammengefasst werden. Die Teilobjekte einer Menge werden als *Elemente der Menge* bezeichnet. Wir schreiben $x \in X$, um zu sagen, dass das Objekt x ein Element der Menge X ist. Es gibt genau eine Menge, die keine Elemente enthält. Diese Menge heißt *leere Menge* und wird mit $\emptyset$ bezeichnet. Mengen können unendlich viele Elemente enthalten.

Eine Menge heißt *endlich*, wenn sie nur endlich viele Elemente enthält. Die Anzahl der verschiedenen Elemente einer endlichen Menge heißt *Kardinalität* der Menge. Die Kardinalität einer endlichen Menge X bezeichnen wir mit $|X|$. Eine n*-elementige Menge* ($n \geq 0$) ist eine endliche Menge mit der Kardinalität n. Zu $n \geq 0$ verschiedenen Objekten $x_1, \ldots, x_n$ gibt es genau eine n-elementige Menge, die genau diese Objekte als Elemente enthält. Diese Menge schreibt man als $\{x_1, \ldots, x_n\}$.

Wir verwenden die folgenden Mengen von Zahlen:

$\mathbb{Z}$	$\stackrel{\text{def}}{=}$	$\{0, -1, 1, -2, 2, -3, 3, \ldots\}$	*Ganze Zahlen*
$\mathbb{N}$	$\stackrel{\text{def}}{=}$	$\{x \in \mathbb{Z} \mid x \geq 0\}$	*Natürliche Zahlen*
$\mathbb{N}^+$	$\stackrel{\text{def}}{=}$	$\{x \in \mathbb{N} \mid x > 0\}$	*Positive natürliche Zahlen*
$\mathbb{R}$			*Reelle Zahlen*
$\mathbb{R}^+$	$\stackrel{\text{def}}{=}$	$\{x \in \mathbb{R} \mid x > 0\}$	*Positive reelle Zahlen*

Um mathematische Aussagen übersichtlich formulieren zu können, verwenden wir die folgenden logischen Notationen (A und B sind Aussagen):

$A \wedge B$	A und B	*Konjunktion*
$A \vee B$	A oder B	*Disjunktion*
$\neg A$	nicht A	*Negation*
$A \Rightarrow B$	aus A folgt B	*Implikation*
$A \Leftarrow B$	A folgt aus B	*Implikation*
$A \iff B$	A genau dann, wenn B	*Äquivalenz*
$\forall x \in X : A(x)$	für alle $x \in X$ gilt $A(x)$	*Universelle Quantifizierung*
$\exists x \in X : A(x)$	es existiert ein $x \in X$ sodass $A(x)$	*Existenziele Quantifizierung*

Für die *Gleichheit von zwei Mengen* X und Y gilt:

$$X = Y \iff (\forall x \in X : x \in Y) \wedge (\forall y \in Y : y \in X)$$

Mengen legen also keine Ordnung für ihre Elemente fest. Außerdem können Objekte nicht mehrfach in einer Menge enthalten sein.

Die *Inklusionsbeziehung* zwischen zwei Mengen X und Y ist wie folgt definiert:

$$X \subseteq Y \stackrel{\text{def}}{\iff} \forall x \in X : x \in Y$$

Wenn $X \subseteq Y$ gilt, sagt man, dass X eine *Teilmenge* von Y ist; weiter sagt man, dass Y eine *Obermenge* von X ist. Zwischen Gleichheit und Inklusion von Mengen besteht der folgende Zusammenhang:

$$X = Y \iff X \subseteq Y \wedge Y \subseteq X$$

Die *strikte Inklusionsbeziehung* zwischen zwei Mengen X und Y ist wie folgt definiert:

$$X \subsetneq Y \stackrel{\text{def}}{\iff} X \subseteq Y \wedge X \neq Y$$

Zwei Mengen X und Y heißen *disjunkt* genau dann, wenn sie kein gemeinsames Element haben.

Seien X und Y zwei Mengen. Der *Durchschnitt*, die *Vereinigung* und die *Differenz* von zwei Mengen X und Y sind die wie folgt definierten Mengen:

$$X \cap Y \stackrel{\text{def}}{=} \{ z \mid z \in X \ \wedge \ z \in Y \} \qquad \textit{Durchschnitt}$$

$$X \cup Y \stackrel{\text{def}}{=} \{ z \mid z \in X \ \vee \ z \in Y \} \qquad \textit{Vereinigung}$$

$$X - Y \stackrel{\text{def}}{=} \{ z \in X \mid z \notin Y \} \qquad \textit{Differenz}$$

Dabei steht $z \notin Y$ für $\neg\, z \in Y$.

Wir haben bereits gesagt, dass Mengen Zusammenfassungen von Objekten sind. Weiter gilt, dass Mengen wieder Objekte sind. Wir können also durch Mengenbildung aus Objekten komplexere Objekte konstruieren. Beispielsweise können wir die folgenden verschiedenen Mengen konstruieren:

$$\emptyset, \quad \{\emptyset\}, \quad \{\{\emptyset\}\}, \quad \{\{\{\emptyset\}\}\}, \quad \ldots$$

Die *Potenzmenge*

$$\mathcal{P}(X) = \{ Y \mid Y \subseteq X \}$$

einer Menge X ist die Menge aller Teilmengen von X. Wenn X eine endliche Menge mit n Elementen ist, dann ist $\mathcal{P}(X)$ eine endliche Menge mit 2^n Elementen.

Sei $X \subseteq \mathbb{R}$. Dann heißt $x \in X$

- *Minimum* von X genau dann, wenn $\forall y \in X\colon \ x \leq y$.
- *Maximum* von X genau dann, wenn $\forall y \in X\colon \ y \leq x$.

Das Minimum von X bezeichnen wir mit $\min X$, wenn es existiert. Entsprechend bezeichnen wir das Maximum mit $\max X$, wenn es existiert.

1.2 Tupel

Objekte können zu Tupeln zusammengefasst werden. Ein *n-stelliges Tupel* ist ein zusammengesetztes Objekt der Form

$$\langle x_1, \ldots, x_n \rangle$$

das aus $n \geq 0$ Objekten $x_1, \ldots, x_n$ besteht. Das i-te Teilobjekt x_i heißt die *i-te Komponente* des Tupels. Es gibt genau ein nullstelliges Tupel $\langle\rangle$, das man als *leeres Tupel* bezeichnet. Zweistellige Tupel bezeichnet man als *Paare* und dreistellige Tupel als *Tripel*. Tupel mit keiner oder mit mindestens zwei Komponenten kann man auch mit runden Klammern schreiben:

$$(x_1, \ldots, x_n)$$

Für die Gleichheit von Tupeln gilt

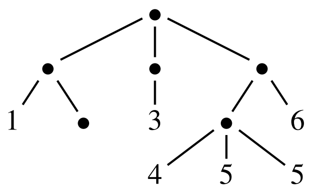
$$\langle x_1, \ldots, x_n \rangle = \langle y_1, \ldots, y_n \rangle \iff x_1 = y_1 \wedge \cdots \wedge x_n = y_n$$

Tupel unterscheiden sich von Mengen in drei Punkten: Für die Komponenten eines Tupels ist eine Ordnung festgelegt; dasselbe Objekt kann mehrfach als Komponente auftreten (zum Beispiel $(2, 2)$); und Tupel haben nur endlich viele Komponenten.

Die Komponenten von Tupeln können wieder Tupel sein. Die hierarchische Struktur solcher geschachtelten Tupel kann man grafisch durch sogenannte Bäume darstellen. Beispielsweise hat das Tupel

$$\langle\langle 1,\ \langle\rangle\rangle,\ \ \langle 3\rangle,\ \ \langle\langle 4,\ 5,\ 5\rangle,\ \ 6\rangle\rangle$$

die *Baumdarstellung*



Wenn man die Grafik um 180 Grad dreht, erinnert sie tatsächlich an einen Baum.

Das *Produkt* von $n \geq 2$ Mengen $X_1, \ldots, X_n$ ist die Menge

$$X_1 \times \cdots \times X_n \stackrel{\text{def}}{=} \{\, \langle x_1, \ldots, x_n \rangle \mid x_1 \in X_1, \ldots, x_n \in X_n \,\}$$

aller n-stelligen Tupel, deren i-te Komponente jeweils aus X_i ist. Für ein Produkt $X \times \cdots \times X$ schreibt man kürzer X^n, wobei $n \geq 2$ die Anzahl der Faktoren angibt.

Die *Summe* von $n \geq 2$ Mengen $X_1, \ldots, X_n$ ist die Menge

$$X_1 \uplus \cdots \uplus X_n \stackrel{\text{def}}{=} \{\, \langle i, x \rangle \mid i \in \{1, \ldots, n\} \ \wedge\ x \in X_i \,\}$$

Die Elemente einer Summe sind also Paare $\langle i, x\rangle$, die aus einer sogenannten Variantennummer i und einem Element x aus dem entsprechenden *Summanden* X_i bestehen. Statt von Summen spricht man auch von *disjunkten Vereinigungen*.

In Standard ML kann man Produkte und Summen mit Typen beschreiben. Beispielsweise können wir die Menge

$$\mathbb{Z} \uplus \mathbb{Z} \uplus (\mathbb{Z} * \mathbb{Z})$$

durch den Typ

```
datatype t = A of int | B of int | C of int * int
```

beschreiben.

1.3 Relationen und Funktionen

Eine *n-stellige Relation* ist eine Menge, deren Elemente n-stellige Tupel sind. Eine *binäre Relation* ist eine zweistellige Relation.

Sei r eine binäre Relation. Der *Definitionsbereich* (englisch „domain“) von r ist die Menge

$$\mathit{Dom}\ r = \{\, x \mid \exists y\colon (x, y) \in r \,\}$$

Der *Wertebereich* (englisch „range“) von r ist die Menge

$$\mathit{Ran}\ r = \{\, y \mid \exists x\colon (x, y) \in r \,\}$$

Hier ist ein Beispiel für eine binäre Relation:

$$\begin{aligned} r &= \{\, (x, y) \in \mathbb{N}^2 \mid x < 2y \,\} \\ \mathit{Dom}\ r &= \mathbb{N} \\ \mathit{Ran}\ r &= \mathbb{N}^+ \end{aligned}$$

Eine binäre Relation r sollte man sich als eine Zuordnung vorstellen, die jedem $x \in \mathit{Dom}\ r$ ein und mehrere $y \in \mathit{Ran}\ r$ zuordnet. Beispielsweise ordnet die obige Relation jeder natürlichen Zahl x alle natürlichen Zahlen y mit $x < 2y$ zu.

Eine *Funktion* f ist eine binäre Relation mit der Eigenschaft, dass zu jedem $x \in \mathit{Dom}\ f$ genau ein $y \in \mathit{Ran}\ f$ existiert mit $(x, y) \in f$. Eine Funktion ist also eine binäre Relation f, die jedem $x \in \mathit{Dom}\ f$ genau ein $y \in \mathit{Ran}\ f$ zuordnet. Wenn $(x, y) \in f$ gilt, sagt man, dass f für das *Argument* x das *Ergebnis* y liefert.

Das Ergebnis von f für x bezeichnet man mit $f(x)$. Statt $(x, y) \in f$ schreibt man meistens $f(x) = y$.

Seien X und Y Mengen. Wir werden die folgenden Notationen benutzen:

$$
\begin{aligned}
X \rightharpoonup Y &= \{ f \mid f \text{ Funktion mit } \mathit{Dom}\ f \subseteq X \text{ und } \mathit{Ran}\ f \subseteq Y \} \\
X \rightarrow Y &= \{ f \in X \rightharpoonup Y \mid \mathit{Dom}\ f = X \} \\
X \overset{\mathit{fin}}{\rightharpoonup} Y &= \{ f \in X \rightharpoonup Y \mid f \text{ endliche Menge} \}
\end{aligned}
$$

Die Elemente von $X \rightharpoonup Y$ bezeichnet wir als *Funktionen* von X nach Y. Die Elemente von $X \rightarrow Y$ bezeichnet wir als *totale Funktionen* von X nach Y. Die Elemente von $X \overset{\mathit{fin}}{\rightharpoonup} Y$ bezeichnet wir als *endliche Funktionen* von X nach Y.

Machen Sie sich klar, dass eine unendliche Funktion keinen Algorithmus zu ihrer Berechnung vorgibt. Wenn Sie eine Prozedur angeben, die eine Funktion berechnet, müssen Sie einen Algorithmus angeben, der die Funktion berechnet.

Eine Funktion $f \in X \rightharpoonup Y$ heißt bezogen auf X und Y

- *total* genau dann, wenn $\mathit{Dom}\ f = X$,
- *partiell* genau dann, wenn $\mathit{Dom}\ f \neq X$,
- *surjektiv* genau dann, wenn $\mathit{Ran}\ f = Y$,
- *injektiv* genau dann, wenn

  $$\forall\, x, x' \in \mathit{Dom}\ f\colon\ f(x) = f(x') \ \Rightarrow\ x = x'$$

- *bijektiv* genau dann, wenn f total, injektiv und surjektiv ist.

Sei $f \in X \rightarrow Y$. Dann ist f bijektiv genau dann, wenn eine Funktion $g \in Y \rightarrow X$ existiert mit

$$\forall x \in X\colon g(f(x)) = x \quad \wedge \quad \forall y \in Y\colon f(g(y)) = y$$

Wenn f bijektiv ist, dann ist die obige Funktion g eindeutig bestimmmt und heißt *Umkehrfunktion* von f. Statt Umkehrfunktion sagt man auch *inverse Funktion*. Bijektive Funktionen bezeichnet man auch als *Bijektionen*.

Darstellung von endlichen Funktionen

Endliche Funktionen

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

schreiben wir oft als

$$\{x_1 \mapsto y_1, \ldots, x_n \mapsto y_n\}$$

Endliche Funktionen kann man sich anschaulich als Tabellen vorstellen:

x_1	y_1
x_2	y_2
...	...
x_n	y_n

Lambda-Notation

Oft ist es bequem, Funktionen mithilfe der sogenannten Lambda-Notation zu beschreiben:

$$\begin{aligned}
&\lambda\, n \in \mathbb{N} \,.\, n && = \{\, (n, n) \mid n \in \mathbb{N} \,\} \\
&\lambda\, x \in \mathbb{Z} \,.\, x^2 && = \{\, (x, y) \mid x \in \mathbb{Z} \wedge y = x^2 \,\} \\
&\lambda\, x \in \mathbb{Z} \,.\, (\text{if } x \geq 0 \text{ then } x \text{ else } -x) && = \{\, (x, y) \mid x \in \mathbb{Z} \wedge y = |x| \,\}
\end{aligned}$$

In jeder Zeile wird links und rechts dieselbe Funktion beschrieben. Dabei verwendet die linke Spalte Lambda-Notation und die rechte Spalte die übliche Mengen-Notation. Das letzte Beispiel verwendet die Lambda-Notation zusammen mit einer Konditional-Notation.[1] Hier ist ein weiteres Beispiel:

$$\lambda\, (x, y) \in \mathbb{Z}^2 \,.\, x + y \quad = \quad \{\, ((x, y), z) \mid x \in \mathbb{Z} \wedge y \in \mathbb{Z} \wedge z = x + y \,\}$$

Adjunktion von Funktionen

Die *Adjunktion* von zwei Funktionen f und g ist wie folgt definiert:

$$f + g \quad = \quad \lambda\, x \in \mathit{Dom}\ f \cup \mathit{Dom}\ g \,.\, \text{ if } x \in \mathit{Dom}\ g \text{ then } g(x) \text{ else } f(x)$$

Statt

$$f + \{x \mapsto y\}$$

schreibt man auch

$$f[y/x]$$

[1] Die Lambda- und die Konditional-Notation findet man nur selten in Mathematik-Lehrbüchern. Dagegen findet man beide Notationen häufig in Programmiersprachen und Informatik-Lehrbüchern.

(lies „f mit y für x“).

Klammersparende Notation

Oft ist es bequem, bei Funktionsapplikationen Klammern einzusparen:

$$\begin{array}{rl} f\ x & \text{statt}\ \ f(x) \\ f\ x\ y & \text{statt}\ \ (f\ x)(y) \end{array}$$

Diese Konvention ist Ihnen bereits aus Standard ML vertraut. Auch bei der Notation $X \to Y$ (alle totale Funktionen von X nach Y) kann man Klammern einsparen:

$$X \to Y \to Z \ \ \text{statt}\ \ X \to (Y \to Z)$$

1.4 Unterobjekte und Wohlfundiertheit

Im folgenden werden wir stets annehmen, dass es genau drei verschiedene Arten von mathematischen Objekten gibt: Zahlen, Tupel und Mengen. Jedes mathematische Objekt gehört zu genau einer dieser drei Arten.

Wir unterscheiden zwischen *atomaren und zusammengesetzten Objekten*. Zahlen, das leere Tupel und die leere Menge sind atomare Objekte. Alle anderen Objekte sind zusammengesetzt.

Ein Objekt x heißt *Unterobjekt* eines Objekts y genau dann, wenn eine der folgenden Bedingungen gilt:

1. y ist eine Menge und x ist ein Element von y.
2. y ist ein Tupel und x ist eine Komponente von y.

Die Unterobjekte der Menge $\mathbb{N}$ sind also genau die natürlichen Zahlen. Die Unterobjekte eines Tupels $\langle x_1, \ldots, x_n \rangle$ sind die Objekte $x_1, \ldots, x_n$. Atomare Objekte haben keine Unterobjekte.

Wir nehmen an, dass es *keine* unendliche Folge

$$x_1,\ x_2,\ x_3,\ \ldots$$

von Objekten gibt, sodass gilt:

$$\forall i \in \mathbb{N}^+ :\ \ x_{i+1} \text{ ist Unterobjekt von } x_i$$

Für jedes Objekt kann also der Abstieg zu einem Unterobjekt nur endlich oft wiederholt werden. Diese Eigenschaft bezeichnet man als die *Wohlfundiertheit* von mathematischen Objekten.

Aus der Wohlfundiertheitsannahme folgt sofort, dass es keine Menge gibt, die sich selbst enthält. Insbesondere folgt, dass es keine Menge gibt, die alle Mengen als Elemente enthält. Die Wohlfundiertheitsannahme sorgt also dafür, dass die Russelsche Antinomie nicht anwendbar ist.

Ein Objekt x heißt *Teilobjekt* eines Objekts y genau dann, wenn es $n \geq 1$ Objekte $x_1, \ldots, x_n$ gibt, sodass die folgenden Bedingungen gelten:

1. $y = x_1$ und $x = x_n$.
2. $\forall i \in \{1, \ldots, n-1\}: \quad x_{i+1}$ ist Unterobjekt von x_i.

Ein Teilobjekt x von y heißt *echtes Teilobjekt* von y, wenn $x \neq y$ gilt.

Ein Objekt heißt *finitär* genau dann, wenn es nur endlich viele Teilobjekte hat.

Übungen

Aufgabe 1.1 (Unter- und Teilobjekte) Sei das Objekt

$$x = \langle 1, \ \{2, 3, \langle 2, 3\rangle\}, \ \langle 2, \langle 3, 7\rangle\rangle\rangle$$

gegeben.

1. Ist x finitär?
2. Geben Sie alle Unterobjekte von x an.
3. Geben Sie alle echten Teilobjekte von x an, die keine Unterobjekte sind.