

Kapitel 2

Mathematische Objekte

Im letzten Kapitel haben wir eine kleine Teilsprache von Standard ML mithilfe von Beispielen beschrieben. Dabei blieben viele Fragen offen. Um eine Programmiersprache präzise und übersichtlich zu beschreiben, benötigt man die Sprache der Mathematik. Die Sprache der Mathematik ist die flexibelste und präziseste Beschreibungssprache über die wir verfügen. Viele der in der Informatik verwendeten Strukturen sind so komplex, dass sie sich nur mit der Sprache der Mathematik klar und genau beschreiben lassen.

Dieses Kapitel beschreibt grundlegende mathematische Objekte wie Mengen, Tupel und Funktionen und führt die entsprechenden Begriffe und Notationen ein. Wir werden zeigen, wie man mithilfe von Rekursion aus einfachen Objekten komplexere Objekte wie die natürlichen Zahlen konstruieren kann. Solche rekursiven Konstruktionen spielen in der Programmierung eine wichtige Rolle.

Wir gehen davon aus, dass Sie schon einige Erfahrung mit der Sprache der Mathematik haben. Sollte dies nicht der Fall sein, empfehlen wir Ihnen das einleitende Kapitel des Lehrbuchs von Michael Sipser.¹

2.1 Grundbegriffe

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

$0, 1, 2, 3, \dots$

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

¹ Michael Sipser, Introduction to the Theory of Computation. PWS Publishing Company, 1997.

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 $x_1, \dots, x_n$ gibt es genau eine *n-elementige Menge*, die genau diese Objekte als Elemente enthält. Diese Menge schreibt man als $\{x_1, \dots, x_n\}$.

Wir verwenden die folgenden Mengen von Zahlen:

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

Die Booleschen Werte *true* und *false* repräsentieren wir durch die natürlichen Zahlen 0 und 1:

$\mathbb{B} \stackrel{\text{def}}{=} \{0, 1\}$	<i>Boolesche Werte</i>
--	------------------------

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

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

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:

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

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

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

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:

$$\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\{\{\emptyset\}\}\}, \dots$$

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

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

2.2 Tupel

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

$$\langle x_1, \dots, x_n \rangle$$

das aus $n \geq 0$ Objekten $x_1, \dots, x_n$ 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:

$$(x_1, \dots, x_n)$$

Für die Gleichheit von Tupeln gilt

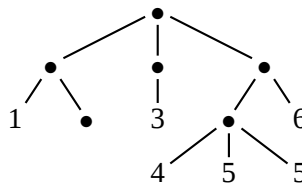
$$\langle x_1, \dots, x_n \rangle = \langle y_1, \dots, y_n \rangle \iff x_1 = y_1 \wedge \dots \wedge x_n = y_n$$

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 $X_1, \dots, X_n$ ist die Menge

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

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 $X_1, \dots, X_n$ ist die Menge

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

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*.

2.3 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

$$\text{Dom } r = \{ x \mid \exists y: (x, y) \in r \}$$

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

$$\text{Ran } r = \{ y \mid \exists x: (x, y) \in r \}$$

Hier ist ein Beispiel für eine binäre Relation:

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

Eine binäre Relation r sollte man sich als eine Zuordnung vorstellen, die jedem $x \in \text{Dom } r$ ein und mehrere $y \in \text{Ran } r$ 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 $x \in \text{Dom } f$ genau ein $y \in \text{Ran } f$ existiert mit $(x, y) \in f$. Eine Funktion ist also

eine binäre Relation f , die jedem $x \in \text{Dom } f$ genau ein $y \in \text{Ran } f$ 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:

$$X \rightarrow Y = \{ f \mid f \text{ Funktion mit } \text{Dom } f \subseteq X \text{ und } \text{Ran } f \subseteq Y \}$$

$$X \rightarrow Y = \{ f \in X \rightarrow Y \mid \text{Dom } f = X \}$$

$$X \xrightarrow{\text{fin}} Y = \{ f \in X \rightarrow Y \mid f \text{ endliche Menge} \}$$

Die Elemente von $X \rightarrow Y$ 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 $X \xrightarrow{\text{fin}} Y$ 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 $f \in X \rightarrow Y$ heißt bezogen auf X und Y

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

$$\forall x, x' \in \text{Dom } f: f(x) = f(x') \Rightarrow x = x'$$

- *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

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

Wenn f bijektiv ist, dann ist die obige Funktion g eindeutig bestimmt 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

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

schreiben wir oft als

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

Endliche Funktionen kann man sich anschaulich als Tabellen vorstellen:

x_1	y_1
x_2	y_2
$\dots$	$\dots$
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.² Hier ist ein weiteres Beispiel:

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

Adjunktion von Funktionen

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

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

Statt

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

schreibt man auch

$$f[y/x]$$

² 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 “).

Standardfunktion für Zahlen

Die Rundungsfunktionen *Floor* $\lfloor _ \rfloor$ und *Ceiling* $\lceil _ \rceil$ sind wie folgt definiert:

$$\begin{aligned} \lfloor _ \rfloor &\in \mathbb{R} \rightarrow \mathbb{Z} & \lceil _ \rceil &\in \mathbb{R} \rightarrow \mathbb{Z} \\ \lfloor x \rfloor &= \max\{z \in \mathbb{Z} \mid z \leq x\} & \lceil x \rceil &= \min\{z \in \mathbb{Z} \mid x \leq z\} \end{aligned}$$

Die *Modulo-Funktion* ist wie folgt definiert:

$$\begin{aligned} \text{mod} &\in \mathbb{Z} \times (\mathbb{Z} - \{0\}) \rightarrow \mathbb{Z} \\ x \text{ mod } y &= x - \lfloor x/y \rfloor y \end{aligned}$$

Für $y > 0$ gilt $x \text{ mod } y < y$.

Die Standard ML Operation $x \text{ div } y$ heißt *ganzzahlige Division* und berechnet $\lfloor x/y \rfloor$ (für $x, y \in \mathbb{Z}$ und $y \neq 0$). Die Operation $x \text{ mod } y$ liefert $x \text{ mod } y$.

2.4 Rekursive Definition von Mengen

Betrachten Sie die folgende Definition einer Menge M :

$$M = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \dots\}$$

Offensichtlich wird die Menge M durch eine Konstruktionsvorschrift für ihre Elemente definiert:

1. Die leere Menge $\emptyset$ ist ein Element von M .
2. Wenn x ein Element von M ist, dann ist die Menge $\{x\}$ ein Element von M .

Diese Konstruktionsvorschrift kann man übersichtlich mit zwei *Inferenzregeln* formulieren:

$$\frac{}{\emptyset \in M} \qquad \frac{x \in M}{\{x\} \in M}$$

Die zweite Regel macht klar, dass M mit einer rekursiven Konstruktionsvorschrift definiert ist. Man kann M auch mit einer rekursiven Gleichung definieren:

$$M = \{\emptyset\} \cup \{\{x\} \mid x \in M\}$$

Alternativ kann man M auch wie folgt definieren: Sei M die kleinste Menge, die $\emptyset$ enthält und die unter Bildung von einelementigen Mengen abgeschlossen ist.

Hier ist ein weiteres Beispiel für eine rekursiv definierte Menge:

$$B = \mathbb{N} \cup (B \times B)$$

Alternativ können wir die Menge B durch zwei Inferenzregeln definieren:

$$\frac{x \in \mathbb{N}}{x \in B} \quad \frac{x \in B \quad y \in B}{(x, y) \in B}$$

Hier sind drei Elemente von B :

$$5, (5, 2), (3, (7, 2)), ((2, (4, 9)), (5, 3))$$

Man kann B auch wie folgt definieren: Sei B die kleinste Menge, die alle Elemente von $\mathbb{N}$ enthält und unter Paarbildung abgeschlossen ist.

Als drittes Beispiel definieren wir eine Menge $P \subseteq \mathbb{N}$ durch die folgenden Inferenzregeln:

$$\frac{}{3 \in P} \quad \frac{}{5 \in P} \quad \frac{x \in P \quad y \in P \quad z = x \cdot y}{z \in P}$$

Diese Regeln definieren P als die kleinste Menge, die 3 und 5 enthält und unter Produktbildung abgeschlossen ist. Alternativ kann man P durch eine rekursive Gleichung definieren:

$$P = \{3, 5\} \cup \{x \cdot y \mid x \in P \wedge y \in P\}$$

Man kann P auch ohne Rekursion definieren:

$$P = \{3^m \cdot 5^n \mid m \in \mathbb{N} \wedge n \in \mathbb{N} \wedge m + n > 0\}$$

Die allgemeine Form einer Inferenzregel ist

$$\frac{A_1 \quad \dots \quad A_n}{A}$$

wobei $n \geq 0$ und $A_1, \dots, A_n$ und A Aussagen sind. Man liest eine Inferenzregel als: Wenn $A_1, \dots, A_n$ gelten, dann gilt A . Man bezeichnet die Aussagen $A_1, \dots, A_n$ als die *Prämissen* und die Aussage A als die *Konklusion* der Inferenzregel. Die obigen Beispiele zeigen, dass Inferenzregeln ohne Prämissen sinnvoll sein können.

Rekursive Definitionen sind konstruktiv in dem Sinne, dass jedes Element der definierten Menge durch *endlich viele Anwendungen* der definierenden Inferenzregeln konstruiert werden kann. Die Protokolle dieser Konstruktionen heißen *Ableitungen* und können als Beweise für die entsprechenden Elementbeziehungen

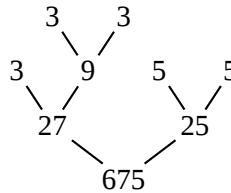
$x \in X$ aufgefasst werden (X ist die rekursiv definierte Menge). Betrachten Sie dazu noch einmal die rekursive Definition der Menge P :

$$\frac{}{3 \in P} \quad \frac{}{5 \in P} \quad \frac{x \in P \quad y \in P \quad z = x \cdot y}{z \in P}$$

Die folgende Ableitung zeigt, dass $675 \in P$:

- (1) $3 \in P$ mit Regel 1
- (2) $5 \in P$ mit Regel 2
- (3) $9 \in P$ mit Regel 3, (1), (1) und $9 = 3 \cdot 3$
- (4) $27 \in P$ mit Regel 3, (1), (3) und $27 = 3 \cdot 9$
- (5) $25 \in P$ mit Regel 3, (2), (2) und $25 = 5 \cdot 5$
- (6) $675 \in P$ mit Regel 3, (4), (5) und $675 = 27 \cdot 25$

Die Essenz dieser Ableitung kann durch einen sogenannten *Ableitungsbaum* übersichtlich dargestellt werden:



2.5 Konstruktion der natürlichen Zahlen

Wir zeigen jetzt, wie man ausgehend von der leeren Menge eine Menge N konstruieren kann, deren Elemente man als die natürlichen Zahlen auffassen kann. Die Menge N definieren wir rekursiv wie folgt:

$$\frac{}{\emptyset \in N} \quad \frac{x \in N}{\{x\} \in N}$$

Die natürlichen Zahlen entsprechen den Elemente von N wie folgt:

- 0 $\emptyset$
- 1 $\{\emptyset\}$
- 2 $\{\{\emptyset\}\}$
- ...

Als Nächstes definieren wir eine Additionsfunktion für N . Die Gleichungen

$$\begin{aligned} 0 + y &= y \\ (x + 1) + y &= (x + y) + 1 \end{aligned}$$

liefern uns die folgende rekursive Definition:

$$plus \in N \times N \rightarrow N$$

$$plus(\emptyset, y) = y$$

$$plus(\{x\}, y) = \{plus(x, y)\}$$

Bei der Definition einer Multiplikationsfunktion für N helfen uns die folgenden Gleichungen:

$$0 \cdot y = 0$$

$$(x + 1) \cdot y = x \cdot y + y$$

Damit bekommen wir:

$$mal \in N \times N \rightarrow N$$

$$mal(\emptyset, y) = \emptyset$$

$$mal(\{x\}, y) = plus(mal(x, y), y)$$

Mit den Gleichungen

$$x - 0 = x$$

$$(x + 1) - (y + 1) = x - y$$

können wir eine partielle Subtraktionsfunktion für N definieren:

$$minus \in N \times N \rightarrow N$$

$$minus(x, \emptyset) = x$$

$$minus(\{x\}, \{y\}) = minus(x, y)$$

Die Konstruktion der natürlichen Zahlen und ihrer Operationen zeigt eindrucksvoll, was man mit den grundlegenden mathematischen Konzepten Menge und Rekursion anstellen kann. In der Informatik bezeichnet man solche Konstruktionen als *Datenstrukturen*. Später werden wir mit ähnlichen Ideen viele interessante Datenstrukturen in Standard ML konstruieren.

Übungsaufgaben

Aufgabe 2.1 (Produkte und Summen) Geben Sie alle Elemente der folgenden Mengen an:

1. $\mathbb{B} \times \mathbb{B}$
2. $\mathbb{B} \uplus \mathbb{B}$
3. $\mathbb{B} \uplus \mathbb{B} \uplus \mathbb{B}$
4. $\mathbb{B} \uplus (\mathbb{B} \times \mathbb{B})$

Aufgabe 2.2 (Endliche Funktionen) Seien die Funktionen $f, g \in \mathbb{B}^2 \rightarrow \mathbb{B}$ wie folgt gegeben:

(x, y)	$f(x, y)$	$g(x, y)$
$(0, 0)$	0	0
$(0, 1)$	0	1
$(1, 0)$	0	1
$(1, 1)$	1	1

1. Beschreiben Sie f und g mit Lambda-Notation:

$$f = \lambda (x, y) \in \mathbb{B}^2 . \dots$$

$$g = \lambda (x, y) \in \mathbb{B}^2 . \dots$$

2. Beschreiben Sie f und g als Menge:

$$f = \{ \dots \}$$

$$g = \{ \dots \}$$

3. Beschreiben Sie die Adjunktion $f + \{ \langle 0, 0 \rangle \mapsto 1, \langle 0, 1 \rangle \mapsto 1 \}$ mit Lambda-Notation und als Menge.

4. Welche der folgenden Aussagen sind gültig:

$$\text{a) } f \in \mathbb{B}^2 \xrightarrow{\text{fin}} \mathbb{B}$$

$$\text{b) } f + g \in \mathbb{B}^2 \rightarrow \mathbb{B}$$

$$\text{c) } \mathbb{B}^2 \rightarrow \mathbb{B} = \mathbb{B}^2 \xrightarrow{\text{fin}} \mathbb{B}$$

$$\text{d) } \mathbb{B}^2 \xrightarrow{\text{fin}} \mathbb{B} = \mathbb{B}^2 \rightarrow \mathbb{B}$$

5. Geben Sie eine injektive Funktion in $\mathbb{B}^2 \rightarrow \mathbb{B}$ an, die nicht surjektiv ist.
6. Geben Sie eine injektive und surjektive Funktion in $\mathbb{B}^2 \rightarrow \mathbb{B}$ an.
7. Gibt es eine bijektive Funktion in $\mathbb{B}^2 \rightarrow \mathbb{B}$?

8. Geben Sie die Kardinalitäten der Mengen $\mathbb{B}^2 \rightarrow \mathbb{B}$ und $\mathbb{B}^2 \rightharpoonup \mathbb{B}$ an.
9. Seien X und Y endliche Mengen. Geben Sie die Kardinalitäten von $X \rightarrow Y$ und $X \rightharpoonup Y$ an (mithilfe von $|X|$ und $|Y|$).

Aufgabe 2.3 (Kaskadierte Funktionen) Sei die Funktion

$$f = \lambda (x, y) \in \mathbb{B}^2. \text{ if } x = y \text{ then } 1 \text{ else } 0$$

gegeben.

1. Es gibt genau eine Funktion $f' \in \mathbb{B} \rightarrow (\mathbb{B} \rightarrow \mathbb{B})$ mit

$$\forall x \in \mathbb{B} \forall y \in \mathbb{B}: f(x, y) = (f'(x))(y)$$

Geben Sie f' mit Lambda-Notation und als Menge an.

2. Es gibt genau eine Funktion

$$K \in (\mathbb{B} \rightarrow (\mathbb{B} \rightarrow \mathbb{B})) \rightarrow (\mathbb{B}^2 \rightarrow \mathbb{B})$$

mit

$$\forall f \in \mathbb{B} \rightarrow (\mathbb{B} \rightarrow \mathbb{B}) \forall x \in \mathbb{B} \forall y \in \mathbb{B}: (f(x))(y) = (K(f))(x, y)$$

- a) Geben Sie K mit Lambda-Notation an.
- b) Ist K bijektiv? Wenn ja, geben Sie die Umkehrfunktion von K an.

Aufgabe 2.4 (Rekursive Definition mit Inferenzregeln) Sie die Menge $P \subseteq \mathbb{N}^3$ rekursiv wie folgt definiert:

$$\frac{y \in \mathbb{N}}{(0, y, 0) \in P} \quad \frac{(x, y, z) \in P}{(x + 1, y, z + y) \in P}$$

1. Geben Sie eine Ableitung und einen Ableitungsbaum für die Aussage $(3, 5, 15) \in P$ an.
2. Beschreiben Sie P mit einer rekursiven Gleichung $P = \dots$.
3. Die Menge $p = \{ ((x, y), z) \mid (x, y, z) \in P \}$ ist eine Funktion in $\mathbb{N}^2 \rightarrow \mathbb{N}$. Beschreiben Sie p mit Lambda-Notation.

Aufgabe 2.5 (Listen) Wir wollen endliche Folgen von natürlichen Zahlen durch die folgenden Objekte repräsentieren:

1. Die leere Folge wird durch das Tupel $\langle \rangle$ repräsentiert.

2. Eine nichtleere Folge $x_1, \dots, x_n$ wird durch das Paar $\langle x_1, r \rangle$ repräsentiert, wobei r die Repräsentation der Folge $x_2, \dots, x_n$ ist.

Beispielsweise wird die Folge 2, 7, 4 durch das Objekt $\langle 2, \langle 7, \langle 4, \langle \rangle \rangle \rangle \rangle$ repräsentiert. Die Repräsentation einer Folge $x_1, \dots, x_n$ bezeichnen wir mit $[x_1, \dots, x_n]$.

1. Definieren Sie mithilfe von Inferenzregeln die Menge

$$L = \{ [x_1, \dots, x_n] \mid n \geq 0 \wedge x_1 \in \mathbb{N} \wedge \dots \wedge x_n \in \mathbb{N} \}$$

ohne die []-Notation zu benutzen. Die Elemente von L werden als Listen bezeichnet.

2. Definieren Sie die Funktion

$$\begin{aligned} length &\in L \rightarrow \mathbb{N} \\ length([x_1, \dots, x_n]) &= n \end{aligned}$$

ohne die []-Notation zu benutzen.

3. Definieren Sie die Funktion

$$\begin{aligned} sum &\in L \rightarrow \mathbb{N} \\ sum([x_1, \dots, x_n]) &= x_1 + \dots + x_n \end{aligned}$$

ohne die []-Notation zu benutzen.

4. Definieren Sie die Funktion

$$\begin{aligned} @ &\in L \times L \rightarrow L \\ [x_1, \dots, x_m]@ [y_1, \dots, y_n] &= [x_1, \dots, x_m, y_1, \dots, y_n] \end{aligned}$$

ohne die []-Notation zu benutzen.

5. Definieren Sie die Funktion

$$\begin{aligned} reverse &\in L \rightarrow L \\ reverse([x_1, \dots, x_n]) &= [x_n, \dots, x_1] \end{aligned}$$

ohne die []-Notation zu benutzen.