LEHRBUCH

Kurt-Ulrich Witt

Mathematische Grundlagen für die Informatik

Mengen, Logik, Rekursion

Springer Vieweg

Mathematische Grundlagen für die Informatik

Kurt-Ulrich Witt

Mathematische Grundlagen für die Informatik

Mengen, Logik, Rekursion

Springer Vieweg

Prof. Dr. Kurt-Ulrich Witt
Hochschule Bonn-Rhein-Sieg
St. Augustin, Deutschland
kurt-ulrich.witt@h-brs.de

ISBN 978-3-658-03078-0 ISBN 978-3-658-03079-7 (eBook)
DOI 10.1007/978-3-658-03079-7

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer Vieweg

Planung und Lektorat: Ulrike Schmickler-Hirzebruch | Barbara Gerlach

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

Springer Vieweg ist eine Marke von Springer DE. Springer DE ist Teil der Fachverlagsgruppe Springer Science+Business Media
www.springer-vieweg.de

Vorwort

Viele Studierende in Informatik-Studiengängen sind insbesondere am Anfang ihres Studiums überrascht über den Umfang und die Intensität der mathematischen Inhalte, mit denen sie sich auseinandersetzen müssen. Vorhandene Vorkenntnisse aus Schulen oder Berufsausbildungen helfen nicht unbedingt, da mathematische Begriffe, Methoden und Verfahren im Studium anders präsentiert und an die Studierenden andere Anforderungen für den Umgang damit gestellt werden. So werden z.B. in der Schule und in der Ausbildung die aussagenlogischen Verknüpfungen mithilfe von Wahrheitstafeln eingeführt und berechnet. In diesem Buch wird die Aussagenlogik nicht mehr oder weniger informell mit Wahrheitstafeln, sondern als formale Sprache eingeführt. Formale Sprachen wie Spezifikations-, Programmier-, Datenbank- und Formatierungssprachen sind von wesentlicher Bedeutung für die Informatik. Informatikerinnen und Informatiker müssen nicht nur solche Sprachen kennen und anwenden können, sondern sie müssen solche Sprachen auch selber entwerfen und implementieren können. Deshalb werden in diesem Buch schon direkt zu Beginn anhand der Sprache der Aussagenlogik die für die Definition formaler Sprachen wesentliche Begriffe wie Alphabet, Syntax und Semantik mathematisch präzise definiert und dabei Rekursion als Beschreibungsmethode verwendet.

Dieses Prinzip wird in diesem Buch grundsätzlich verfolgt. Zum einen werden schrittweise elementare Begriffe aus der Logik und der Mengenlehre, zu Relationen und Funktionen, zu Rechenstrukturen und zur Berechenbarkeit formal eingeführt und analysiert. Zum anderen werden für die Informatik bedeutende Problembeschreibungs- und Problemlösemethoden wie formale Notationen, Abstraktion, Schlussfolgerungsmechanismen, Induktion und Rekursion erläutert und angewendet. Die erwähnten Begriffe und Methoden bilden eine wesentliche Basis für das weitere Studium. Sowohl in Mathematik-Kursen, wie z.B. Analysis, Stochastik und Statistik, Algebra und Zahlentheorie, als auch in Kursen zu allen Bereichen der Informatik, wie z.B. Automatentheorie, Formale Sprachen, Berechenbarkeit, Datenstrukturen und Algorithmen, Programmierung, Datenbanksysteme, Betriebssysteme und Rechnernetze, werden diese Begriffe und Methoden benötigt und weiterentwickelt.

Das Buch richtet sich somit an Erstsemester-Studierende in Informatik- und Mathematik-Studiengängen. Es ist als Begleitlektüre zu entsprechenden Lehrveranstaltungen an Hochschulen aller Art und insbesondere zum Selbststudium geeignet. Jedes Kapitel beginnt mit einer seinen Inhalt motivierenden Einleitung und der Auflistung von Lernzielen, die durch das Studium des Kapitels erreicht werden sollen. Zusammenfassungen am Ende von Abschnitten oder am Ende von Kapiteln bieten Gelegenheit, den Stoff zu reflektieren. Die meisten Beweise sind vergleichsweise ausführlich und mit Querverweisen versehen, die die Zusammenhänge aufzeigen. Eingestreut sind viele Beispiele und über sechzig Aufgaben, deren Bearbeitung zur Festigung des Wissens und zum Üben der dar-

gestellten Methoden und Verfahren dienen. Zu fast allen Aufgaben sind am Ende des Buches oder im Text Musterlösungen aufgeführt. Die Aufgaben und Lösungen sind als integraler Bestandteil des Buches konzipiert. Wichtige Begriffe sind als Marginalien aufgeführt; der Platz zwischen den Marginalien bietet Raum für eigene Notizen.

Das Schreiben und das Publizieren eines solchen Buches ist nicht möglich ohne die Hilfe und ohne die Unterstützung von vielen Personen, von denen ich an dieser Stelle allerdings nur einige nennen kann: Als Erstes erwähne ich die Autoren der Publikationen, die ich im Literaturverzeichnis aufgeführt habe. Alle dort aufgeführten Werke habe ich für den einen oder anderen Aspekt verwendet. Ich kann sie allesamt für weitere ergänzende Studien empfehlen. Zu Dank verpflichtet bin ich auch vielen Studierenden, deren kritische Anmerkungen in meinen Lehrveranstaltungen zu Themen dieses Buches ich beim Schreiben berücksichtigt habe. Trotz dieser Hilfen wird das Buch Fehler und Unzulänglichkeiten enthalten. Diese verantworte ich allein – für Hinweise zu deren Beseitigung bin ich dankbar.

Die Publikation eines Buches ist nicht möglich ohne einen Verlag, der es herausgibt. Ich danke dem Springer-Verlag für die Bereitschaft zur Publikation und insbesondere Frau Schmickler-Hirzebruch für ihre Ermunterung zur und ihre Unterstützung bei der Publikation des Buches.

Mein größter Dank gilt allerdings meiner Familie für den Freiraum, den sie mir für das Schreiben dieses Buches gegeben hat.

Bedburg, im Juli 2013

K.-U. Witt

Inhaltsverzeichnis

1 Mengen und Logik

Ausgangspunkt unserer ersten Überlegungen soll der Begriff *Menge* sein. Bei welchen Gelegenheiten verwenden Sie diesen Begriff? Denken Sie einen Moment nach, bilden Sie zwei, drei Sätze mit diesem Begriff! Mir fallen spontan Sätze ein, wie „Letztes Wochenende, auf dem Altstadtfest, da war eine Menge los, und wir haben eine Menge Bier getrunken.“, „Von Popmusik versteht sie eine Menge.“, „Bis Du mit Deinem Studium fertig bist, wird noch eine Menge Wasser den Rhein runterlaufen.“, „Eine große Menschenmenge versammelte sich auf dem Marienplatz und wartete auf die Ankunft des neuen Deutschen Meisters.“

In welchem Sinn wird in diesen Sätzen der Begriff „Menge“ verwendet? Doch eher um auszudrücken, dass man nicht genau weiß, wie groß etwas ist oder wie viel von irgendetwas vorhanden ist. Genaueres kann oder will man nicht angeben; jedenfalls möchte man mitteilen, dass es sich um viel oder Großes handelt.

In der Mathematik und in der Informatik wird der Begriff Menge im Gegensatz zum umgangssprachlichen Gebrauch in einem präzisem Sinn benutzt. Er dient dazu, Objekte zu einer neuen Einheit zusammenzufassen, so dass diese Einheit als Ganzes betrachtet und weiterverwendet werden kann. Typische Beispiele in der Informatik sind Dateien und Datenbanken. Eine Kundendatenbank enthält etwa die für ein Unternehmen wichtigen Daten ihrer Kunden. Anfragen und Auswertungen einer Datenbank liefern als Ergebnis wieder Mengen. Um festzustellen, welche Kunden ihre Rechnungen bis zum Ende des letzten Quartals noch nicht bezahlt haben, müssen möglicherweise die Kundendatei mit der Bestelldatei und der Rechnungsdatei geeignet verknüpft werden, um die für das Versenden von Mahnungen notwendige Menge von Daten zu finden. Die Eigenschaften der Ergebnismenge müssen in einer formalen Sprache präzise beschrieben werden, damit das Datenbanksystem die gewünschte Ergebnismenge bestimmen kann. Solche Sprachen benutzen in aller Regel logische Ausdrücke.

So werden wir zunächst den Begriff der Menge präzisieren und kennen lernen, wie Mengen festgelegt werden können. Für formale Beschreibungen von Mengen eignen sich logische Ausdrücke. Deshalb werden wir uns mit der Aussagenlogik und mit der Prädikatenlogik beschäftigen. Dabei werden wir sehen, wie formale Sprachen festgelegt werden können. Formale Sprachen sind ein wesentliches Werkzeug in der Informatik. Informatikerinnen und Informatiker müssen diese Werkzeuge nicht nur geeignet anwenden können, sondern Sie müssen auch in der Lage sein, formale Sprachen selbst zu schaffen, und zwar so, dass Problemstellungen und Lösungsverfahren damit so beschrieben werden können, dass Sie mithilfe von Rechnersystemen gelöst bzw. ausgeführt werden können. Wir werden anhand der Aussagenlogik an einem einfachen Beispiel sehen, wie die Konstruktion einer formalen Sprache im Prinzip erfolgen kann.

Des Weiteren benutzen wir die logische Ausdrücke, um Beziehungen zwischen Mengen und Operationen auf Mengen zu definieren sowie Eigenschaften dieser Beziehungen und Operationen zu zeigen. Das ist z.B. wichtig für den oben

K.-U. Witt, *Mathematische Grundlagen für die Informatik*,
DOI 10.1007/978-3-658-03079-7_1, © Springer Fachmedien Wiesbaden 2013

bereits erwähnten Anwendungsbereich der Datenbanken. Hier müssen Mengen in Beziehung gesetzt und miteinander verknüpft werden. Ein Datenbanksystem muss mit Mengen rechnen können. Genau wie uns beim Rechnen mit Zahlen „nach Adam Riese“ bestimmte Eigenschaften helfen, wie z.B. das Ausklammern von gemeinsamen Faktoren, gibt es auch für das Rechnen mit Mengen Gesetze, die das Datenbanksystem benutzt, um korrekt und effizient Ergebnismengen zu berechnen. Mithilfe von logischen Schlussfolgerungen kann aus Fakten, die in einer Datenbank abgelegt sind, neues Wissen abgeleitet werden, und logische Ausdrücke steuern den Ablauf von Prozessen.

Durch Abstraktion gelangen wir am Schluss dieses Kapitels zu der Rechenstruktur einer *Booleschen Algebra*. Wir werden sehen, dass die vorher betrachtete Aussagenlogik und die ebenfalls vorher betrachteten Teilmengen einer Menge mit logischen Verknüpfungen bzw. mit Mengenverknüpfungen als Rechenoperationen spezielle Prototypen solcher Algebren sind.

Abstraktion ist eines *der* Problemlöse-Hilfsmittel in der Mathematik und in der Informatik. So ist man in aller Regel nicht an der Lösung eines sehr speziellen Problems interessiert, sondern an Methoden, Verfahren und Werkzeugen, die in möglichst vielen Anwendungsbereichen zur Lösung von Problemen eingesetzt werden können. Mit einem Datenbankmanagementsystem sollten Datenbanken nicht nur in einem Anwendungsbereich, sondern bei unterschiedlichen Anwendungen realisiert werden können – unabhängig von der der Art der Daten. Das Suchen von Elementen in Mengen, die Verknüpfung von Mengen und die Darstellung von Mengen muss unabhängig von den konkreten Elementen einer Menge gelöst werden. Ein Sortieralgorithmus sollte unabhängig sein vom Typ der Werte nach denen sortiert werden soll. Er sollte eine Menge von Daten korrekt und effizient sortieren unabhängig davon, ob es sich z.B. um Kundennamen, Datumsangaben oder Rechnungsnummern handelt.

1.1 Definition und Darstellung von Mengen

In diesem Kapitel wollen wir den Begriff der Menge in einer für die weiteren Betrachtungen hinreichenden Art und Weise präzisieren. Wir lernen, wie Mengen dargestellt und beschrieben werden, wir vereinbaren Bezeichner für gängige Zahlenmengen, und wir werden sehen, dass unser Mengenbegriff zu Schwierigkeiten führen kann.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- den Cantorschen Begriff der Menge verstehen,
- die Darstellungen von Mengen sowie die Bezeichner für die gängigen Zahlenmengen kennen,

- Mengen geeignet in aufzählender und beschreibender Darstellung beschreiben können,
- die Russelsche Antinomie erklären können.

1.1.1 Ein Mengenbegriff

Ausgangspunkt unserer Erklärung des Begriffes *Menge* soll die folgende Festlegung sein:[1]

> Eine *Menge* ist eine Zusammenfassung bestimmter, wohlunterschiedener Dinge unserer Anschauung oder unseres Denkens, welche *Elemente* der Menge genannt werden, zu einem Ganzen.

Menge
Element

Diese Festlegung ist mehr eine informelle Vereinbarung, denn eine präzise mathematische Definition. Sie setzt ein einheitliches Verständnis der darin verwendeten Begriffe voraus, und sie wird zu den schon angekündigten Schwierigkeiten führen. Eine axiomatische, „mathematisch saubere“ Einführung des Mengenbegriffs, welche diese Schwierigkeiten vermeidet, wäre aber im Hinblick darauf, wie wir diesen verwenden wollen, viel zu aufwändig. Immerhin haben zu Beginn des letzten Jahrhunderts eine Reihe von hervorragenden Logikern und Mathematikern viele Anstrengungen unternommen und geraume Zeit dafür benötigt, um zu einer formal zufrieden stellenden Fundierung dieses Begriffes zu gelangen. Für die gängigen Betrachtungen in Mathematik und Informatik reicht das Verständnis dieses Begriffes, welches wir mithilfe der folgenden weiteren Festlegungen bekommen werden, vollkommen aus.

Die Notation einer Menge, also „die Zusammenfassung von Dingen zu einem Ganzen“, erfolgt in der Art

$$\{\dots\}$$

Die Elemente der Menge werden durch die Mengenklammern $\{$ und $\}$ zu einem Ganzen zusammengefasst. „$\dots$“ legt eindeutig fest, welche Dinge Elemente der Menge sind. Um auf Mengen Bezug nehmen zu können, geben wir diesen in aller Regel einen Namen. Dazu verwenden wir zumeist große Buchstaben. Die Namensgebung erfolgt mithilfe eines Gleichheitszeichens:

$$M = \{\dots\}$$

Ist ein Ding a Element einer Menge M, dann schreiben wir $a \in M$. Ist ein Ding a kein Element einer Menge, dann schreiben wir $a \notin M$. Wenn wir ausdrücken wollen, dass mehrere Elemente zu einer Menge gehören, dann schreiben wir das auch, indem wir diese Elemente durch Kommata getrennt auflisten und dafür die Zugehörigkeit angeben. So schreiben wir z.B. anstelle von $a \in M$, $b \in M$, $c \in M$ kürzer $a, b, c \in M$. Entsprechend schreiben wir $a, b, c \notin M$, anstelle von $a \notin M$, $b \notin M$, $c \notin M$, falls a, b und c keine Elemente der Menge M sind.

Element

1 Diese Festlegung geht auf Georg Cantor (1845 - 1918) zurück, der als Begründer der Mengenlehre gilt und der wichtige Beiträge zur „modernen“ Mathematik geliefert hat.

Anschaulich kann man sich Mengen als Behälter, z.B. als Schachteln oder als Säcke, vorstellen. Der Behälter wird durch die Mengenklammern dargestellt. Genau wie Schachteln weitere Schachteln enthalten können, kann eine Menge weitere Mengen enthalten, und diese Mengen können wiederum Mengen enthalten. Und genau wie eine Schachtel leer sein kann, kann auch eine Menge leer sein. Aus der Art der oben vereinbarten Mengennotation folgt, dass die *leere Menge* durch $\{\}$ dargestellt wird. Die leere Menge wird auch durch das Symbol $\emptyset$ dargestellt. Ist die Menge M leer, so notieren wir $M = \emptyset$ oder $M = \{\}$. Offensichtlich gilt $a \notin \emptyset$ für jedes Ding a.

Leere Menge

Beispiel 1.1 **a)** Die Menge

$$A = \{\, 1, 2, 3, 4, 5 \,\}$$

enthält als Elemente die Zahlen 1, 2, 3, 4 und 5. Es gilt also z.B. $2, 5 \in A$ sowie $0, 6, 13 \notin A$.

b) Die Menge

$$B = \{\, 1, 2, \{\, 3, 4, 5, 6 \,\} \,\}$$

enthält drei Elemente: die Zahlen 1 und 2 sowie die Menge $\{\, 3, 4, 5, 6 \,\}$, die selbst vier Elemente enthält, nämlich die Zahlen 3, 4, 5 und 6. Machen Sie sich dies an der Schachtel-Metapher klar: Die Schachtel B enthält die Elemente 1, 2 sowie eine weitere Schachtel, die die Elemente 3, 4, 5 und 6 enthält. B kann in diese Schachtel nicht „hineinschauen“. Deutlich wird das, wenn wir der inneren Schachtel einen Namen geben, etwa $C = \{\, 3, 4, 5, 6 \,\}$. Dann ergibt sich die Darstellung $B = \{\, 1, 2, C \,\}$, woraus man unmittelbar sieht, das B drei und nicht sechs Elemente enthält. Es gilt $C \in B$, d.h. $\{\, 3, 4, 5, 6 \,\} \in \{\, 1, 2, \{\, 3, 4, 5, 6 \,\} \,\}$. Die Schachtel C ist ein Element der Schachtel B, und es gilt $3 \in C$, aber $3 \notin B$.

c) Die Gleichung $x^2 = -1$ besitzt in der Menge der reellen Zahlen keine Lösung, d.h. ihre Lösungsmenge $L = \{\, x \mid x \text{ ist reelle Zahl und } x^2 = -1 \,\}$ ist leer: $L = \emptyset$.

d) Die Menge

$$D = \{\{\}\}$$

enthält genau ein Element, nämlich die leere Menge. Somit ist die Menge D selbst nicht leer. Auch dies kann man sich mit der Schachtel-Metapher veranschaulichen. Die Schachtel D ist nicht leer, denn sie enthält ein Element, nämlich die leere Schachtel. Es gilt $\{\} \in D$. Wenn wir $E = \{\}$ setzen, dann ist $D = \{E\}$, wodurch auch in der mathematischen Notation deutlich wird, dass D nicht die leere Menge ist. □

Übungsaufgaben

1.1 a) Sei M eine Menge sowie $A = \{\, 3, 4, \{\, 5, 6 \,\} \,\}$ und $B = \{\, 5, 6 \,\}$. Setzen Sie $\in$ oder $\notin$ korrekt ein:

(1) $\emptyset \ldots \emptyset$	(2) $\emptyset \ldots \{\emptyset\}$	(3) $M \ldots \{\, \{\, 1, 2 \,\}, M \,\}$
(4) $1 \ldots \{\, \{\, 1, 2 \,\}, M \,\}$	(5) $3 \ldots A$	(6) $\{\, 4 \,\} \ldots A$
(7) $\{\, 5, 6 \,\} \ldots A$	(8) $\{\, B \,\} \ldots A$	(9) $B \ldots A$
(10) $6 \ldots A$	(11) $6 \ldots B$	(12) $\{5, 6\} \ldots B$

b) Bestimmen Sie, ob folgende Aussagen wahr oder falsch sind und begründen Sie Ihre Antworten!

(1) $\emptyset \in \emptyset$	(2) $\emptyset \in \{\emptyset\}$
(3) $\{\, a, b \,\} \in \{\, a, b, c, \{\, a, b \,\} \,\}$	(4) $\{\, a, b, c \,\} \in \{\, a, b, c, \{\, a, b \,\} \,\}$
(5) $\{\, a, b \,\} \in \{\{\ a, b \,\}\}$	(6) $a \in \{\, a, b, \{\, a, b \,\} \,\}$ □

1.1.2 Darstellung von Mengen

Unsere Festlegung des Mengenbegriffs besagt, dass die Elemente einer Menge bestimmt sein müssen. Dazu verwenden wir zwei Arten der Darstellung von Mengen: die *aufzählende* und die *beschreibende* Darstellung. Bei der aufzählenden Darstellung werden die Elemente der Menge explizit angegeben. In dieser Art und Weise haben wir bereits die Mengen im Beispiel 1.1 sowie in den Übungen 1.1 dargestellt.

Beispiel 1.2 Wir geben einige weitere Beispiele für die aufzählende Darstellung von Mengen an:

Aufzählende Darstellung von Mengen

$$A = \{\, 2, 3, 5, 7, 11 \,\}$$
$$B = \{\, 1, 2, \ldots, 50 \,\}$$
$$C = \{\, 1, 2, \ldots \,\}$$
$$D = \{\, 43, 44, \ldots \,\}$$

Die Menge A enthält fünf Elemente, nämlich die Primzahlen kleiner gleich 11. Bei den drei anderen Mengen wird ein Problem der aufzählenden Darstellung von Mengen offensichtlich. Falls sie viele Elemente enthalten oder falls sie unendlich viele Elemente enthalten, ist ihre komplette Aufzählung aufwändig bzw. unmöglich. In diesen Fällen helfen wir uns mit der „$\ldots$“-Schreibweise. Dabei muss durch die explizit angegebenen Elemente eindeutig klar sein, für welche Elemente „$\ldots$“ steht. In B sind offensichtlich die ganzen Zahlen von 1 bis 50 gemeint, in C die natürlichen Zahlen ohne die Null und in D die natürlichen Zahlen größer gleich 43. □

Bei der Menge $E = \{\, 3, 5, 7, \ldots \,\}$ könnte nicht klar sein, welche Menge gemeint ist: die Menge der positiven ungeraden Zahlen größer gleich 3 oder die

Menge der Primzahlen größer gleich 3. Die explizite Aufzählung eines weiteren Elementes könnte Aufklärung bringen: 9 im ersten Fall bzw. 11 im zweiten Fall.

Die allgemeine Form der aufzählenden Darstellung von Mengen ist also

$$M = \{ a_1, a_2, \ldots, a_n \}$$

für endliche Mengen sowie

$$M = \{ a_1, a_2, \ldots \}$$

für unendliche Mengen.

Übungsaufgaben

1.2 Geben Sie folgende Mengen in aufzählender Form an:

(1) M_1 = Menge aller nicht negativen ganzen Zahlen, die Kubikzahlen von ganzen Zahlen und kleiner als 100 sind.

(2) M_2 = Menge aller ganzen Zahlen zwischen 10 und 50, die durch 3 aber nicht durch 4 teilbar sind.

(3) M_3 = Menge aller Mengen, die man aus den drei Elementen a, b und c bilden kann,

(4) $M_4 = \{ x \mid x = \sqrt{y}$ und $0 \leq y \leq 100$ und $x \in \mathbb{N} \}$,

(5) $M_5 = \{ x \mid x$ ist ein positiver Teiler von $24 \}$,

(6) $M_6 = \{ x \mid x = 2$ und $x = 4 \}$,

(7) $M_7 = \{ x \mid x = 2$ oder $x \in \{ 1, 2, 3 \} \}$. □

Multimenge

Erfüllt die Menge $F = \{ 3, 4, 5, 3, 6 \}$ unsere Festlegung des Mengenbegriffs? Nein, denn Ihre Elemente sind nicht wohlunterschieden, da 3 mehr als einmal vorkommt. Wohlunterschieden bedeutet, dass jedes Element nur einmal in der Menge vorkommen darf. In der Informatik gibt es durchaus Anwendungen, in denen Mengen ein Element auch mehrfach enthalten können sollten. Man spricht dann von *Multimengen* oder *Bags*. Die Menge

$$M = \{ 1, 1, 2, 3, 3, 3, 3, 4, 5, 5, 5 \}$$

ist ein Beispiel einer Multimenge. Durch unterschiedliche Kenntlichmachung gleicher Elemente – etwa durch fortlaufende Indizierung oder durch Angabe der Häufigkeit – kann man diese unterscheidbar machen und so Multimengen durch Mengen darstellen. Nach Indizierung sieht die Menge M wie folgt aus

$$M = \{ 1_1, 1_2, 2_1, 3_1, 3_2, 3_3, 3_4, 4_1, 5_1, 5_2, 5_3 \}$$

sowie bei Angabe der Häufigkeit

$$M = \{\, 1(2), , 2(1), 3(4), 4(1), 5(3) \,\}$$

Die später folgenden Definitionen zu Teilmengen und Operationen auf Mengen können mithilfe solcher Kennzeichnungen auf Multimengen übertragen werden.

Wir betrachten im Folgenden in aller Regel keine Multimengen, sondern Mengen, die jedes Element genau einmal enthalten. Dabei sei beispielhaft auf eine Problematik hingewiesen, die entstehen kann, falls die Elemente Variablen sind und keine „konkreten“ Elemente (Konstanten). Sind etwa a, b, c und d Variable für ganze Zahlen, und bilden wir damit die Menge $M = \{\, a, b, c, d \,\}$, dann hängt der Inhalt dieser Menge von der Belegung der Variablen ab. Haben z.B. a und c beide den Wert 5 und b den Wert 1 und d den Wert 2, dann ist $M = \{\, 1, 2, 5 \,\}$, d.h. M enthält in diesem Fall drei Elemente. Für den Fall, dass alle Variablen denselben Wert haben, enthält M genau ein Element.

Beschreibende Darstellung von Mengen

Bei der beschreibenden Darstellung werden die Elemente nicht explizit aufgezählt, sondern es wird eine sie definierende Eigenschaft angegeben. Die allgemeine Form ist

$$M = \{\, x \mid p(x) \,\} \tag{1.1}$$

Dabei ist x ein Platzhalter (eine Variable) für die Elemente der Menge, und $p(x)$ ist eine für x informal oder formal angegebene Eigenschaft. Genau die Dinge, die die Eigenschaft erfüllen, sind Elemente der Menge. Auf Möglichkeiten die definierende Eigenschaft formal durch Prädikate anzugeben, gehen wir im Kapitel 1.7 ein.

Beispiel 1.3 Beispiele für die beschreibende Darstellung von Mengen sind:

$$\begin{aligned}
A &= \{\, x \mid x \text{ ist eine Primzahl kleiner gleich } 11 \,\} \\
G &= \{\, x \mid x \text{ ist eine positive ganze Zahl und } x + x = 10 \,\} \\
H &= \{\, (x, y) \mid x \text{ und } y \text{ sind positive ganze Zahlen und } x + y = 6 \,\} \\
T_{64} &= \{\, y \mid y \text{ ist ein positiver Teiler von } 64 \,\} \\
S &= \{\, \textit{st} \mid \textit{st} \text{ studiert Informatik an der Hochschule Bonn-Rhein-Sieg} \,\}
\end{aligned}$$

Dabei haben wir die Eigenschaften, welche die Elemente festlegen sollen, „halbformal“ ausgedrückt, d.h. umgangssprachlich unter Verwendung von mathematischen Ausdrücken. In aufzählender Darstellung gilt offensichtlich:

$$\begin{aligned}
A &= \{\, 2, 3, 5, 7, 11 \,\} \\
G &= \{\, 5 \,\} \\
H &= \{\, (0, 6), (1, 5), (2, 4), (3, 3), (4, 2), (5, 1), (6, 0) \,\} \\
T_{64} &= \{\, 1, 2, 4, 8, 16, 32, 64 \,\}
\end{aligned}$$

□

Übungsaufgaben

1.3 Geben Sie folgende Mengen in beschreibender Form an:
(1) $M_1 = \{\, 1, 2, 4, 8, 16, 32, 64 \,\}$,
(2) $M_2 = \{\, 4, 9, 25, 49, 121 \,\}$,
(3) $M_3 = \{\, 1, 8, 27, 64 \,\}$,
(4) $M_4 = \{\, 1, 3, 5, 7, \ldots \,\}$. □

Kardinalität einer Menge

Endliche Menge

Unendliche Menge

Enthält eine Menge M endlich viele Elemente, etwa m Stück, dann schreiben wir $|M| = m$ und nennen M eine *endliche Menge*. $|M|$ heißt die *Kardinalität* von M. Nicht endliche Mengen heißen *unendlich*, und wir notieren $|M| = \infty$.

Es gilt also z.B. $|A| = 5$, $|B| = 50$, $|D| = \infty$ für die Mengen A, B und D aus Beispiel 1.2 auf Seite 5 sowie $|G| = 1$, $|H| = |T_{64}| = 7$ für die Mengen G und H aus Beispiel 1.3. Offensichtlich gilt $|\emptyset| = 0$.

Im Kapitel 2.3 betrachten wir das Thema Mächtigkeit von Mengen noch detaillierter.

1.1.3 Bezeichner für Zahlenmengen

Gängige Zahlenmengen besitzen feste Bezeichner. Wir listen diese im Folgenden auf und verwenden sie so fortan.

Natürliche Zahlen

$\mathbb{N} = \{\, 1, 2, 3, \ldots \,\}$	natürliche Zahlen
$\mathbb{N}_0 = \{\, 0, 1, 2, 3, \ldots \,\}$	natürliche Zahlen mit 0
$\mathbb{N}_k = \{\, k, k+1, k+2, \ldots \,\}$	natürliche Zahlen ab k, $k \in \mathbb{N}_0$
$\mathbb{N}_{u,o} = \{\, u, u+1, u+2, \ldots, o \,\}$	natürliche Zahlen zwischen u und o, $u, o \in \mathbb{N}_0$, $u \leq o$
$\mathbb{P} = \{\, 2, 3, 5, 7, 11, \ldots \,\}$	Primzahlen

Für $\mathbb{N}_{u,o}$ schreiben wir auch $[\, u, o \,]$ und nennen dies das Intervall der natürlichen Zahlen von u bis o. Des Weiteren setzen wir $\mathbb{N}_{u,o} = \emptyset$, falls u größer als o ist.

Beispiele für die letzten Notationen sind

$$\begin{aligned} \mathbb{N}_{17} &= \{\, 17, 18, 19, \ldots \,\} \\ \mathbb{N}_{35,53} &= \{\, 35, 36, \ldots, 53 \,\} \\ &= [\, 35, 53 \,] \end{aligned}$$

Ganze Zahlen

$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$	ganze Zahlen
$\mathbb{G}_+ = \{0, 2, 4, \ldots\}$	nicht negative gerade Zahlen
$\mathbb{G}_- = \{-2, -4, \ldots\}$	negative gerade Zahlen
$\mathbb{G} = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$	gerade Zahlen
$\mathbb{U}_+ = \{1, 3, 5, \ldots\}$	positive ungerade Zahlen
$\mathbb{U}_- = \{-1, -3, -5 \ldots\}$	negative ungerade Zahlen
$\mathbb{U} = \{\ldots, -5, -3, -1, 1, 3, 5, \ldots\}$	ungerade Zahlen

Rationale, reelle und komplexe Zahlen

$\mathbb{Q} = \left\{ \frac{p}{q} \mid p \in \mathbb{Z} \text{ und } q \in \mathbb{N} \right\}$	rationale Zahlen (Brüche)
$\mathbb{Q}_+, \mathbb{Q}_-$	rationale Zahlen größer gleich/kleiner 0
$\mathbb{R}, \mathbb{R}_+, \mathbb{R}_-$	reelle Zahlen/größer gleich/kleiner 0
$\mathbb{C} = \{ a + bi \mid a, b \in \mathbb{R},\ i^2 = -1 \}$	komplexe Zahlen

Wie setzen an dieser Stelle die angegebenen Zahlenmengen in dem Sinne als bekannt voraus, als dass man sie aus der Schule und dem täglichen Leben kennt. In späteren Kapiteln geben wir formale Definitionen für die Mengen der natürlichen, ganzen und rationalen Zahlen an und betrachten einige ihrer wesentlichen Eigenschaften.

1.1.4 Russellsche Antinomie

Unsere Festlegung des Mengenbegriffs lässt es zu, dass Mengen Elemente von Mengen sein können (siehe bisherige Beispiele und Übungen). Kann eine Menge sich auch selbst enthalten? In aller Regel fallen uns nur Beispiele von Mengen ein, die sich nicht selbst enthalten, wie z.B. die Menge D aller rechtwinkligen Dreiecke in der Ebene. Offensichtlich ist D kein Element von D, denn die Menge D ist kein rechtwinkliges Dreieck. Betrachten wir nun die Menge D' aller Elemente, die keine rechtwinkligen Dreiecke sind. Da die Menge D' offensichtlich kein rechtwinkliges Dreieck ist, muss sie selbst zur Menge D' gehören, denn diese enthält ja alle Elemente, die keine rechtwinkligen Dreiecke sind.

Nun betrachten wir die Menge M, welche alle Mengen enthält. M ist selbst eine Menge, also muss M sich selbst enthalten, denn M enthält ja *alle* Mengen. Unsere auf der Cantorschen Definition basierende Festlegung des Mengenbegriffs

lässt diese Mengenbildungen zu. Nun betrachten wir die Menge M' aller Mengen, die sich nicht selbst enthalten, formal

$$M' = \{ A \mid A \notin A \}$$

und stellen die Frage, ob M' sich selbst enthält: $M' \in M'$? Gemäß der Cantorschen Mengendefintion muss bestimmt werden können, ob eine Ding zu einer Menge gehört oder nicht. Für die Frage, ob $M' \in M'$ gilt, muss also entweder die Antwort „ja“ oder die Antwort „nein“ zutreffen, und die richtige Antwort muss anhand der beschreibenden Eigenschaft entschieden werden können.

Bevor wir diese Frage beantworten, betrachten wir als Vorbereitung darauf das folgende Paradoxon:

> Sei S die Schlange, die all diejenigen Schlangen in den Schwanz beißt, die sich nicht selbst in den Schwanz beißen. Frage: Beißt S sich selbst in den Schwanz?

Es gibt zwei Möglichkeiten: S beißt sich selbst in den Schwanz oder nicht. Nehmen wir an, S beiße sich in den Schwanz. Dann gehört sie zu den Schlangen, die sich selber in den Schwanz beißen und die deshalb nicht von S in den Schwanz gebissen werden. Dies bedeutet aber, dass S nicht von S gebissen wird, also beißt S sich nicht selbst in den Schwanz. Nehmen wir an, S beiße sich nicht in den Schwanz. Dann gehört sie zu den Schlangen, die sich nicht selber in den Schwanz beißen. Diese werden aber gerade von S gebissen, also beißt S sich selbst in den Schwanz. In beiden möglichen Fällen führt die jeweilige Annahme zu einem Widerspruch, d.h. die Frage kann nicht beantwortet werden. Es liegt ein sogenanntes Paradoxon vor.

Kehren wir zur Frage „$M' \in M'$?“ zurück und gehen zu deren Beantwortung wie beim Schlangen-Paradoxon vor: Nehmen wir an, dass $M' \in M'$ gelte. Dann gehört M' zu den Mengen, die sich nicht selbst enthalten. Daraus folgt aber $M' \notin M'$. Nehmen wir an, dass $M' \notin M'$ gelte. Dann enthält M' sich nicht selbst, d.h. M' gehört zu den Mengen, die sich nicht selbst enthalten. Daraus folgt aber $M' \in M'$. Wir stellen also fest, dass auch diese Frage nicht beantwortet werden kann. Dieses Paradoxon ist bekannt als *Russelsche Antinomie*.[2] Es zeigt die Unzulänglichkeit der Cantorschen Mengendefinition. Eine Mengendefinition sollte in jedem Fall die eindeutige Beantwortung der Frage ermöglichen, ob ein Ding in einer Menge enthalten ist oder nicht. Die axiomatische Mengenlehre vermeidet Antinomien. Hierauf gehen wir, wie eingangs des Kapitels bereits angemerkt, nicht ein, da für unsere Zwecke im Folgenden die Cantorsche Mengendefinition ausreicht, Widersprüche treten nicht auf.

2 Benannt nach Bertrand Russell (1872 - 1970), britischer Logiker, Philosoph, Schriftsteller und Pazifist. Er lieferte wesentliche Beiträge zur Logik und Philosophie. Mit Abert Einstein initiierte er die Pugwash-Bewegung, mit Jean-Paul Sartre das Vietnam-Tribunal, das erste sogenannte Russelsche Tribunal. 1950 erhielt er den Nobelpreis für Literatur.

1.1.5 Zusammenfassung

Mengen sind ein grundlegendes, wesentliches Hilfsmittel in der Mathematik und in der Informatik, um Dinge zusammenzufassen und dieser Zusammenfassung einen Namen zu geben. Die Dinge, die zu einer Menge zusammengefasst werden, heißen Elemente der Menge. Mengen können auf zwei Arten dargestellt werden: aufzählend oder beschreibend.

Bei der aufzählenden Darstellung werden die Elemente der Menge explizit angegeben. Dies ist bei Mengen mit vielen Elementen und erst recht bei unendlichen Mengen unmöglich. Man hilft sich dann mit der „Pünktchennotation“, bei der einige Elemente angegeben werden. Aus den angegebenen Elementen und Informationen aus dem Zusammenhang muss bestimmt werden können, ob ein Ding Element der Menge ist oder nicht.

Bei der beschreibenden Darstellung werden die Eigenschaften, die die Elemente einer Menge haben, durch ein Prädikat angegeben. Prädikate werden in der Regel durch „halbformale“ Ausdrücke angegeben, die neben logischen (Teil-) Ausdrücken auch natürliche Sprachkomponenten enthalten.

Die „klassische“ Cantorsche Mengendefinition kann zu Paradoxien führen, was durch die Russelsche Antinomie deutlich wird.

1.2 Aussagenlogik

Aussagen- und Prädikatenlogik sind von grundlegender Bedeutung in der Informatik. Programm- und Prozessabläufe sind in aller Regel abhängig vom Erfülltoder Unerfülltsein von miteinander verknüpften Bedingungen. So ist z.B. die Steuerung eines Überdruckventils eines Heizkessels abhängig von seiner aktuellen Stellung, der Temperatur und des Drucks. Bei einer Datenbankabfrage müssen der erwarteten Antwort entsprechende Bedingungen formuliert werden wie z.B.: „Alle Kunden, die bis zum Ende des letzten Quartals ihre Rechnungen noch nicht bezahlt haben und keine Stammkunden sind, müssen eine Zahlungserinnerung bekommen.“ Wir werden, wie im Abschnitt 1.1.2 angekündigt, Prädikate verwenden, um darstellende Beschreibungen von Mengen zu formulieren (siehe (1.1) auf Seite 7). Außerdem benutzen wir Prädikate, um im Kapitel 1.9 Teilmengenbeziehungen und Mengenverknüpfungen zu definieren und deren Eigenschaften zu beweisen. Wir beginnen in diesem Kapitel mit der Aussagenlogik und behandeln im Kapitel 1.7 einführend die Prädikatenlogik. Wir wollen die Aussagenlogik – möglicherweise anders, als Sie diese bisher kennen gelernt haben – als eine Sprache betrachten. Sprachen, etwa formale Sprachen, wie Programmier- oder Dialogsprachen in der Informatik, oder natürliche Sprachen, wie die deutsche oder die englische Sprache, oder künstlerische Sprachen,

wie die Poesie oder die Musik, werden festgelegt durch

- ein Alphabet, welches ein endlicher Zeichenvorrat ist, aus dem die Wörter und Sätze einer Sprache zusammengesetzt sind,
- die Syntax, die festlegt, welche mit den Elementen des Alphabets gebildete Zeichenketten als Wörter oder Sätze zur Sprache gehören,
- die Semantik, welche den Wörtern und Sätzen der Sprache eine Bedeutung zuordnet.

Lernziele

Nach Durcharbeiten dieses Kapitels sollten Sie

- den syntaktischen Aufbau aussagenlogischer Formeln kennen,
- die Semantik aussagenlogischer Formeln erklären können,
- den Wahrheitswert aussagenlogischer Formeln sowohl mit dem Interpretationsoperator als auch mithilfe von Wahrheitstafeln berechnen können,
- die Definition der aussagenlogischen Operationen Negation, Disjunktion, Konjunktion, Subjunktion, Bijunktion, exklusives Oder kennen,
- aussagenlogische Grundbegriffe wie Erfüllbarkeit, Tautologie, Kontradiktion und Modell erläutern können,
- überprüfen können, ob eine Menge aussagenlogischer Formeln erfüllbar ist.

1.2.1 Alphabet der Aussagenlogik

Das Alphabet der Aussagenlogik besteht aus zwei Mengen:

Aussagenlogische Operatorsymbole

- aus der Menge der *aussagenlogischen Operatorsymbole*

$$O = \{\, \underline{0}, \underline{1}, \neg, \wedge, \vee, (,) \,\}$$

Aussagenlogische Variablen

- sowie aus der Menge $\{\, x, | \,\}$ zur Generierung der aussagenlogischen Variablen. Die Menge V der *aussagenlogischen Variablen* kann wie folgt rekursiv definiert werden:

  (1) x ist eine Variable: $x \in V$.

  (2) Falls α ein Variable ist, dann auch $\alpha|$, d.h., ist $\alpha \in V$, dann ist auch $\alpha| \in V$.

  Mithilfe dieser Regeln ergibt sich:

$$V = \{\, x, x|, x||, x|||, \ldots \,\}$$

  Wenn wir die Anzahl der Striche als Index notieren, dann ist

$$V = \{\, x_0, x_1, x_2, x_3, \ldots \,\}$$

  Wir werden in aller Regel nicht konkrete Elemente aus V, also indizierte x, in den aussagenlogischen Formeln verwenden, sondern dazu Variablenbezeichner benutzen, die bei Bedarf mit konkreten Elementen aus V bezeichnet

werden können. Als Variablenbezeichner verwenden wir kleine Buchstaben vom Ende des deutschen Alphabetes, z.B. p, q, r, v, x, y und z, bei Bedarf auch indiziert. Mit der Aussage $v \in V$ ist gemeint, dass v anstelle irgendeiner aussagenlogischen Variable $x_i \in V$ steht (v ist Platzhalter für x_i).

1.2.2 Syntax aussagenlogischer Formeln

Aussagenlogische Formeln

Die Sprache $\mathcal{A}$ der Aussagenlogik, deren Elemente – ihre Wörter – *aussagenlogische Formeln* heißen, ist durch folgende Syntax-Regeln festgelegt:

Aussagenlogische Konstantenbezeichner

(i) Die Operatorsymbole $\underline{0}, \underline{1} \in O$, die so genannten *aussagenlogischen Konstantenbezeichner*, sind Wörter in $\mathcal{A}$: $\underline{0}, \underline{1} \in \mathcal{A}$.

(ii) Jede aussagenlogische Variable ist auch eine aussagenlogische Formel: Für jedes $v \in V$ ist $v \in \mathcal{A}$.

(iii) Als Variablenbezeichner für aussagenlogische Formeln verwenden wir kleine Buchstaben vom Anfang des griechischen Alphabetes: $\alpha, \beta, \gamma, \ldots$, bei Bedarf auch indiziert, z.B. $\alpha_1, \alpha_2, \ldots$

Aus bereits vorhandenen aussagenlogischen Formeln werden mithilfe der Operator- und der Klammersymbole neue Formeln gebildet: Sind α,$\beta \in \mathcal{A}$, dann sind auch $(\alpha \wedge \beta), (\alpha \vee \beta), \neg\alpha \in \mathcal{A}$.[3]

(iv) Genau die gemäß den Regeln (i) - (iii) bildbaren Zeichenketten gehören zu $\mathcal{A}$.

Atomare Formeln

Zusammengesetzte Formeln

Literale

Konstanten und Variablen heißen auch *atomare Formeln*. Die unter Verwendung von Regel (iii) gebildeten Formeln heißen *zusammengesetzt*. Die Formeln der Gestalt v sowie die der Gestalt $\neg v$ mit $v \in V$ heißen *Literale*. Literale sind also Variable sowie mit dem Operator $\neg$ versehene Variable.

Beispiel 1.4 Es gilt:

a) $(p \wedge q) \in \mathcal{A}$.

b) $(((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)) \vee \underline{0}) \in \mathcal{A}$.

c) $p(\neg q \vee r) \notin \mathcal{A}$.

Wir zeigen, dass b) gilt:

(1) $\underline{0}, p, q, r \in \mathcal{A}$ gemäß Regel (i) bzw. Regel (ii).

(2) Gemäß (1) und Regel (iii) ist $\neg r \in \mathcal{A}$.

(3) Gemäß (1) und Regel (iii) ist $(q \wedge r) \in \mathcal{A}$.

(4) Gemäß (1,2) und Regel (iii) ist $(q \vee \neg r) \in \mathcal{A}$.

(5) Gemäß (4) und Regel (iii) ist $\neg(q \vee \neg r) \in \mathcal{A}$.

(6) Gemäß (1,3) und Regel (iii) ist $(p \vee (q \wedge r)) \in \mathcal{A}$.

3 Anstelle von $\neg\alpha$ ist auch die Notation $\overline{\alpha}$ üblich.

(7) Gemäß (4,6) und Regel (iii) ist $((p \vee (q \wedge r)) \wedge \neg(q \wedge \neg r)) \in \mathcal{A}$.

(8) Gemäß (1,7) und Regel (iii) ist $(((p \vee (q \wedge r)) \wedge \neg(q \wedge \neg r)) \vee \underline{0}) \in \mathcal{A}$.

Durch schrittweises Anwenden der Regeln (i) - (iii) haben wir die aussagenlogische Formel $(((p \vee (q \wedge r)) \wedge \neg(q \wedge \neg r)) \vee \underline{0})$ konstruiert.

Diese Formel enthält die vier Literale p, q, r sowie $\neg r$. □

Wir werden im Folgenden, falls dadurch keine Missverständnisse auftreten, die äußeren Klammern einer Aussage weglassen. Wir schreiben also $p \wedge q$ anstelle von $(p \wedge q)$ und $((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)) \vee \underline{0}$ anstelle von $(((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)) \vee \underline{0})$.

1.2.3 Semantik aussagenlogischer Formeln

Aussagenlogische Konstanten Wahrheitswerte

Die Bedeutung von aussagenlogischen Formeln wollen wir durch die Werte 0 für „falsch“ und 1 für „wahr“ angeben. Die Menge dieser beiden *aussagenlogischen Konstanten* bzw. *Wahrheitswerte* bezeichnen wir mit $\mathbb{B}$. Dabei legen wir auf $\mathbb{B} = \{ 0, 1 \}$ eine Ordnung fest: 0 sei kleiner als 1, d.h. das Maximum der beiden Wahrheitswerte ist 1, was wir durch $max\{ 0, 1 \} = 1$ ausdrücken, dass Minimum ist 0, d.h. $min\{ 0, 1 \} = 0$. Außerdem legen wir als Operationen auf $\mathbb{B}$ fest: $1-1 = 0$ sowie $1 - 0 = 1$.

Für diese Operationen gelten die beiden folgenden Beziehungen:

$$min\{ x, y \} = 1 - max\{ 1 - x, 1 - y \} \tag{1.2}$$
$$max\{ x, y \} = 1 - min\{ 1 - x, 1 - y \} \tag{1.3}$$

Wir können also eine der beiden Operationen *min* bzw. *max* durch die jeweils andere und die „$-$“-Operation ausdrücken, wir könnten also auf *min* oder auf *max* verzichten. Aus schreibtechnischen Gründen und der besseren Lesbarkeit wegen werden wir aber beide Operationen verwenden.

Abstrakte Maschine

Wir können $(\mathbb{B}, max, min, -)$ als eine *abstrakte Maschine* (auch *Rechenstruktur* oder *algebraische Struktur* genannt) auffassen, die die Werte 0 und 1 zur Verfügung stellt und darauf die Operationen max, min und „$-$“ ausführen kann. Wir benötigen jetzt noch eine Vorschrift, die festlegt, wie eine aussagenlogische Formel – abhängig von Eingaben – von dieser Maschine berechnet wird.

Übungsaufgaben

1.4 Verifizieren Sie die Beziehungen (1.2) und (1.3)! □

Interpretation aussagenlogischer Formeln

Wir haben die Syntax der aussagenlogischen Formeln rekursiv definiert: Die atomaren Formeln werden als gegeben angenommen und mit diesen und bereits konstruierten Formeln werden mithilfe von Operatoren und Klammern neue Formeln konstruiert. *Rekursion* ist eine Konstruktionmethode mit fundamentaler Bedeutung in der Informatik, insbesondere beim Entwurf von Datenstrukturen und Algorithmen sowie in der Programmierung. Wir werden in späteren Kapiteln noch ausführlicher auf Rekursion eingehen, im Kapitel 4 ist dieses Prinzip Grundlage für die Definition eines Berechenbarkeitsbegriffs und einer universellen Programmiersprache.

Rekursion

Wegen der rekursiven Definition ihrer Syntax ist es „natürlich“, die Semantik aussagenlogischer Formeln entsprechend rekursiv zu definieren: Zunächst muss den atomaren Formeln, also den Konstantenbezeichnern und Variablen einer Formel, ein Wahrheitswert zugewiesen werden, und daraus wird dann der Wahrheitswert der gesamten Formel entsprechend ihres induktiven Aufbaus rekursiv berechnet. Die Zuweisung von Wahrheitswerten zu Variablen einer Formel $\gamma \in \mathcal{A}$ geschieht mit einer *Belegung* $\mathcal{I}$ („Eingabe“), und die Berechnung des Wahrheitswertes von γ geschieht durch die *Interpretation* $\mathcal{I}^*$

Belegung
Interpretation

Sei $\gamma \in \mathcal{A}$ eine aussagenlogische Formel, dann sei V_γ die Menge der Variablen in γ. Diese Menge kann wie folgt (rekursiv) definiert werden:

(i) $V_\gamma = \{\}$, falls $\gamma \in \{\underline{0}, \underline{1}\}$,

(ii) $V_\gamma = \{\gamma\}$, falls $\gamma \in V$,

(iii) $V_\gamma = V_\alpha$, falls $\gamma = \neg\alpha$; $V_\gamma = V_\alpha \cup V_\beta$, falls $\gamma = \alpha \wedge \beta$ oder $\gamma = \alpha \vee \beta$.[4]

Jeder Variablen v in γ, also jedem $v \in V_\gamma$, wird mit der Belegung $\mathcal{I}$ genau ein Wahrheitswert zugewiesen. Formal: $\mathcal{I} : V_\gamma \to \mathbb{B}$ mit $\mathcal{I}(v) \in \mathbb{B}$. Für jede Variable $v \in V_\gamma$ gibt es zwei mögliche Belegungen: $\mathcal{I}(v) = 0$ oder $\mathcal{I}(v) = 1$. Ist n die Anzahl der Variablen in γ, also $|V_\gamma| = n$, dann gibt es 2^n mögliche Belegungen $\mathcal{I} : V_\gamma \to \mathbb{B}$. Diese fassen wir in der Menge

$$\mathcal{I}_\gamma = \mathbb{B}^{V_\gamma} = \{\, \mathcal{I} \mid \mathcal{I} : V_\gamma \to \mathbb{B} \,\}$$

zusammen.

Mit einer gewählten Belegung $\mathcal{I} \in \mathcal{I}_\gamma$ wird die Interpretation $\mathcal{I}^*(\gamma)$ der aussagenlogischen Formel $\gamma \in \mathcal{A}$ gemäß den folgenden Regeln berechnet (dabei erfolgt die Festlegung der Regeln rekursiv entsprechend der rekursiven Definitionen (i) - (iii) der Syntax der aussagenlogischen Formeln im vorigen Abschnitt):

(i) Für $\gamma \in \{\underline{0}, \underline{1}\}$ sei $\mathcal{I}^*(\underline{0}) = 0$ und $\mathcal{I}^*(\underline{1}) = 1$: Die Konstantenbezeichner werden unabhängig von der gegebenen Formel durch fest zugewiesene Wahrheitswerte interpretiert.

4 Für zwei Mengen A und B bedeutet $A \cup B$ die Vereinigung von A und B: Die Elemente von A und B werden zu einer Menge zusammengefasst; dabei werden mehrfach vorkommende Elemente natürlich nur einmal aufgeführt (siehe auch Kapitel 1.9).

(ii) $\mathcal{I}^*(v) = \mathcal{I}(v)$, für ein $\mathcal{I} \in \mathcal{I}_\gamma$: Die Variablen $v \in V_\gamma$ der Formel γ werden durch die gewählte Belegung $\mathcal{I}$ interpretiert.

(iii) Die Interpretation zusammengesetzter Formeln wird gemäß folgender Regeln berechnet:

Ist $\gamma = \alpha \wedge \beta$ mit $\alpha,\beta \in \mathcal{A}$, dann ist

$$\mathcal{I}^*(\gamma) = \mathcal{I}^*(\alpha \wedge \beta) = \mathit{min}\{\, \mathcal{I}^*(\alpha), \mathcal{I}^*(\beta)\,\}$$

Ist $\gamma = \alpha \vee \beta$ mit $\alpha,\beta \in \mathcal{A}$, dann ist

$$\mathcal{I}^*(\gamma) = \mathcal{I}^*(\alpha \vee \beta) = \mathit{max}\{\, \mathcal{I}^*(\alpha), \mathcal{I}^*(\beta)\,\}$$

Ist $\gamma = \neg\alpha$ mit $\alpha \in \mathcal{A}$, dann ist

$$\mathcal{I}^*(\gamma) = \mathcal{I}^*(\neg\alpha) = 1 - \mathcal{I}^*(\alpha)$$

Beispiel 1.5 Wir betrachten die Formel

$$\gamma = ((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)) \vee \underline{0}$$

aus Beispiel 1.4 auf Seite 13. Es ist $V_\gamma = \{\, p, q, r \,\}$. Wir wählen die Belegung $\mathcal{I}(p) = 1, \mathcal{I}(q) = 0$ sowie $\mathcal{I}(r) = 1$. Mit dieser Belegung ergibt sich gemäß den obigen Regeln folgende Interpretation:

$$\begin{aligned}
\mathcal{I}^*(\gamma) \;&=\; \mathcal{I}^*(((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)) \vee \underline{0}) \\
&= \mathit{max}\{\mathcal{I}^*((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)), \mathcal{I}^*(\underline{0})\} \\
&= \mathit{max}\{\mathit{min}\{\mathcal{I}^*(p \vee (q \wedge r)), \mathcal{I}^*(\neg(q \vee \neg r))\}, 0)\} \\
&= \mathit{max}\{\mathit{min}\{\mathit{max}\{\mathcal{I}^*(p), \mathcal{I}^*(q \wedge r)\}, 1 - \mathcal{I}^*(q \vee \neg r)\}, 0\} \\
&= \mathit{max}\{\mathit{min}\{\mathit{max}\{\mathcal{I}(p), \mathit{min}\{\mathcal{I}^*(q), \mathcal{I}^*(r)\}\}, 1 - \mathit{max}\{\mathcal{I}^*(q), \mathcal{I}^*(\neg r)\}\}, 0\} \\
&= \mathit{max}\{\mathit{min}\{\mathit{max}\{1, \mathit{min}\{\mathcal{I}(q), \mathcal{I}(r)\}\}, 1 - \mathit{max}\{\mathcal{I}(q), 1 - \mathcal{I}^*(r)\}\}, 0\} \\
&= \mathit{max}\{\mathit{min}\{\mathit{max}\{1, \mathit{min}\{0, 1\}\}, 1 - \mathit{max}\{0, 1 - \mathcal{I}(r)\}\}, 0\} \\
&= \mathit{max}\{\mathit{min}\{\mathit{max}\{1, \mathit{min}\{0, 1\}\}, 1 - \mathit{max}\{0, 1 - 1\}\}, 0\} \\
&= \mathit{max}\{\mathit{min}\{\mathit{max}\{1, 0\}, 1 - 0\}, 0\} \\
&= \mathit{max}\{\mathit{min}\{1, 1\}, 0\} \\
&= \mathit{max}\{1, 0\} \\
&= 1
\end{aligned}$$

Für die gewählte Belegung $\mathcal{I}$ ist der Wert der aussagenlogischen Formel also 1. Berechnen Sie den Wert für weitere der insgesamt $2^3 = 8$ möglichen Belegungen! □

Wir werden im Folgenden, da wir die Operatoren $\underline{0}$ und $\underline{1}$ fest mit den Werten 0 bzw. 1 belegt haben, äußerlich nicht mehr zwischen den Operatoren und ihren Werten unterscheiden und 0 anstelle von $\underline{0}$ bzw. 1 anstelle von $\underline{1}$ schreiben. Des Weiteren vereinbaren wir, dass der Operator $\neg$ stärker bindet, als der Operator $\wedge$,

und dieser stärker als $\vee$. Diese Vereinbarung hilft, Klammern einzusparen. Wir schreiben also z.B. $\alpha \wedge \beta \vee \gamma$ anstelle von $(\alpha \wedge \beta) \vee \gamma$, während im Allgemeinen $(\alpha \vee \beta) \wedge \gamma$ nicht als $\alpha \vee \beta \wedge \gamma$ geschrieben werden darf, denn letztere entspricht $\alpha \vee (\beta \wedge \gamma)$.

Programmiersprache

Wir können das Tripel $(\mathcal{A}, \mathbb{B}, \mathcal{I}^*)$ auch als eine *Programmiersprache* auffassen: Jedes $\alpha \in \mathcal{A}$ ist ein Programm, $\mathbb{B}$ enthält die möglichen Eingaben für die Variablen sowie die möglichen Ausgaben, die eine Interpretation in Abhängigkeit der Eingaben berechnen kann. $\mathcal{I}$ legt eine konkrete Eingabe fest, und $\mathcal{I}^*$ ist der Interpreter, der jedes Programm bei gegebener Eingabe ausführt. Die Maschine, auf der dieser Interpreter ausgeführt werden kann, benötigt als Maschinenbefehle *max*, *min* sowie $1 - 0$ und $1 - 1$, mit diesen können alle aussagenlogischen Formeln berechnet werden.

Am obigen Beispiel 1.5 erkennt man, dass die Berechnung einer Interpretation noch effizienter gestaltet werden kann. Im Beispiel werden „stur“ die Regeln schrittweise von links nach rechts ausgeführt. Ist man z.B. bei einem Ausdruck der Form *min*$\{\, \mathcal{I}^*(\alpha), 0 \,\}$ angekommen, so kann man diesen wegen der Definition von *min* sofort durch 0 ersetzen, ohne dass $\mathcal{I}^*(\alpha)$ noch ausgerechnet werden muss. Insbesondere, wenn α noch ein komplexer Ausdruck ist, wird dadurch viel Zeit bei der Berechnung der Interpretation eingespart. Überlegen Sie sich weitere Möglichkeiten zur Effizienzsteigerung bei der Auswertung! Wir werden an dieser Stelle nicht weiter darauf eingehen. Ein Gebiet der Informatik, welches sich mit solchen Fragestellungen beschäftigt ist der Compilerbau.

Wahrheitstafeln

Die Semantik aussagenlogischer Formeln und der Wert einer aussagenlogischen Formel kann auch mithilfe sogenannter *Wahrheitstafeln* festgelegt bzw. berechnet werden. Seien α,$\beta \in \mathcal{A}$ zwei aussagenlogische Formeln. Die folgenden Tabellen legen jeweils für alle möglichen Werte dieser beiden Formeln die Werte ihrer Verknüpfungen mit den Operationen $\neg$, $\wedge$ und $\vee$ fest:

α	$\neg\alpha$
1	0
0	1

α	β	$\alpha \wedge \beta$
1	1	1
1	0	0
0	1	0
0	0	0

α	β	$\alpha \vee \beta$
1	1	1
1	0	1
0	1	1
0	0	0

Jede Zeile legt den Wert für genau eine Belegung fest. Wir haben schon überlegt, dass eine aussagenlogische Formel mit n Variablen 2^n mögliche Belegungen besitzt. Daraus folgt, dass die Wahrheitstabelle für eine Formel mit n Variablen 2^n Zeilen umfasst.

Konjunktion

Wir sehen, dass die Verknüpfung zweier Formeln mit $\wedge$ genau dann 1 ergibt, wenn beide Formeln den Wert 1 besitzen. Diese Verknüpfung heißt *Und* -Verknüpfung oder *Konjunktion*. Die Verknüpfung zweier Formeln mit $\vee$ ergibt ge-

Disjunktion
Negation

nau dann 0, wenn beide Formeln den Wert 0 besitzen. Diese Verknüpfung heißt *Oder*-Verknüpfung oder *Disjunktion*. Die Verknüpfung $\neg$ heißt *Negation*.

Beispiel 1.6 Die Wahrheitstafel der Formel

$$\gamma = ((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)) \vee 0$$

aus Beispiel 1.4 b) (Seite 13) ist:

p	q	r	$\underline{0}$	$\neg r$	$q \wedge r$	$q \vee \neg r$	$\neg(q \vee \neg r)$	$p \vee (q \wedge r)$	$(p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)$	$((p \vee (q \wedge r)) \wedge \neg(q \vee \neg r)) \vee 0$
1	1	1	0	0	1	1	0	1	0	0
1	1	0	0	1	0	1	0	1	0	0
1	0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	1	0	1	0	0
0	1	1	0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0
0	0	0	0	1	0	1	0	0	0	0

Anstelle der Auswertung einer aussagenlogischen Formel durch $\mathcal{I}^*$ können also auch die Wahrheitstabellen zu ihrer Berechnung verwendet werden. Dabei enthält jede Zeile gerade die Berechnung für genau eine Belegung $\mathcal{I}$ der Variablen. So entspricht die dritte Zeile der obigen Tabelle der Belegung und der Berechung der Formel γ im Beispiel 1.5 auf Seite 16. □

Aussagenlogische Operationen

Subjunktion
Bijunktion
Exklusives Oder

Neben den bereits bekannten aussagenlogischen Operationen Negation, Konjunktion und Disjunktion führen wir noch drei weitere Operationen ein: die *Subjunktion*, die *Bijunktion* und das *exklusive Oder*. Seien $\alpha,\beta \in \mathcal{A}$ aussagenlogische Formeln, dann sind auch die Subjunktion $\alpha \to \beta$ (in Worten: „wenn α, dann β“ oder „aus α folgt β“), die Bijunktion $\alpha \leftrightarrow \beta$ (in Worten: „α genau dann, wenn β“) sowie das exklusive Oder $(\alpha \oplus \beta)$ (in Worten: „entweder α oder β“) aussagenlogische Formeln. Wir geben ihre Semantik in den folgenden Wahrheitstafeln an:

α	β	$\alpha \to \beta$
1	1	1
1	0	0
0	1	1
0	0	1

α	β	$\alpha \leftrightarrow \beta$
1	1	1
1	0	0
0	1	0
0	0	1

α	β	$\alpha \oplus \beta$
1	1	0
1	0	1
0	1	1
0	0	0

Rechnen Sie nach, dass die folgenden Folgerungen korrekt sind.

Folgerung 1.1 **a)** Für jede Belegung $\mathcal{I}$ der Variablen in den aussagenlogischen Formeln $\alpha,\beta \in \mathcal{A}$ gilt:

$$\mathcal{I}^*(\alpha \to \beta) = \mathcal{I}^*(\neg\alpha \vee \beta) \tag{1.4}$$
$$\mathcal{I}^*(\alpha \leftrightarrow \beta) = \mathcal{I}^*((\alpha \to \beta) \wedge (\beta \to \alpha)) \tag{1.5}$$
$$\mathcal{I}^*(\alpha \oplus \beta) = \mathcal{I}^*((\alpha \wedge \neg\beta) \vee (\neg\alpha \wedge \beta)) \tag{1.6}$$

Die Operationen Subjunktion, Bijunktion und exklusives Oder erweitern also die semantischen Möglichkeiten von $\mathcal{A}$ nicht, denn sie können mithilfe der bereits bekannten Operationen Negation, Konjunktion und Disjunktion ausgedrückt werden. Sie stellen syntaktische Hilfsmittel dar, um gegebenenfalls Ausdrücke kürzer darzustellen.

b) Wir kennen nun sechs verschiedene aussagenlogische Verknüpfungen. Wie viele zweistellige verschiedene aussagenlogische Verknüpfungen gibt es? Jede Wahrheitstafel für zweistellige Verknüpfungen besteht aus vier Zeilen, welche die vier möglichen Belegungen von zwei Variablen darstellen. Jede Belegung führt zu einem Ergebnis, welches jeweils 0 oder 1 ist. Damit gibt es $2^4 = 16$ mögliche Ergebnisse überhaupt, und damit gibt es genau sechzehn verschiedene zweistellige aussagenlogische Verknüpfungen. In a) haben wir bereits gesehen, dass wir mithilfe der Negation zweistellige Operationen durch andere zweistellige Operationen ausdrücken können. Im Abschnitt 1.4.2 werden wir uns mit der Frage beschäftigen, wie viele und welche zweistelligen Operationen ausreichen, um alle sechzehn zweistelligen Operationen darstellen zu können. □

Übungsaufgaben

1.5 Zeigen Sie, dass für jede Belegung $\mathcal{I}$ der aussagenlogischen Formeln $\alpha,\beta \in \mathcal{A}$

$$\mathcal{I}^*(\alpha \wedge \neg\beta) = \mathcal{I}^*(\neg(\alpha \to \beta)) \tag{1.7}$$

gilt! □

Es gilt

$$\begin{aligned}
\mathcal{I}^*(\alpha \wedge \neg\beta) &= min\{\, \mathcal{I}^*(\alpha), 1 - \mathcal{I}^*(\beta)\,\} \\
&= 1 - max\{\, 1 - \mathcal{I}^*(\alpha), \mathcal{I}^*(\beta)\,\} && \text{wegen (1.2)} \\
&= 1 - \mathcal{I}^*(\neg\alpha \vee \beta) \\
&= 1 - \mathcal{I}^*(\alpha \to \beta) && \text{wegen (1.4)} \\
&= \mathcal{I}^*(\neg(\alpha \to \beta))
\end{aligned}$$

Erfüllbarkeit aussagenlogischer Formeln

Für die folgenden Definitionen, die Begriffe für wichtige Eigenschaften aussagenlogischer Formeln festlegen, erweitern wir Belegungen von einzelnen Formeln auf Formelmengen. Sei $\mathcal{F}$ eine endliche Menge von Formeln aus $\mathcal{A}$, dann sei $V_\mathcal{F}$ die Menge der Variablen, die entsteht, wenn alle Variablen aller Formeln in $\mathcal{F}$ zu einer Menge zusammengefasst werden. Die Belegung $\mathcal{I} : V_\mathcal{F} \to \mathbb{B}$ ordnet dann jeder Variablen $v \in V_\mathcal{F}$ eine Belegung $\mathcal{I}(v) \in \mathbb{B}$ zu. Wir verallgemeinern entsprechend $\mathcal{I}_\gamma$ zu

$$\mathcal{I}_\mathcal{F} = \mathbb{B}^{V_\mathcal{F}} = \{\, \mathcal{I} \mid \mathcal{I} : V_\mathcal{F} \to \mathbb{B} \,\}$$

und nennen $\mathcal{I} \in \mathcal{I}_\mathcal{F}$ dann eine Belegung von $\mathcal{F}$.

Beispiel 1.7 Für die Formelmenge $\mathcal{F}_1 = \{\, p \vee q,\ q \wedge \neg r,\ (p \wedge q) \vee (q \to r) \,\}$ gilt $V_{\mathcal{F}_1} = \{\, p, q, r \,\}$. Wählt man etwa die Belegung $\mathcal{I}(p) = \mathcal{I}(q) = 1$ und $\mathcal{I}(r) = 0$, dann besitzen alle drei Formeln in $\mathcal{F}_1$ den Wert 1. □

Definition 1.1 Sei $\alpha \in \mathcal{A}$ eine aussagenlogische Formel und $\mathcal{F}$ eine endliche Menge aussagenlogischer Formeln aus $\mathcal{A}$.

Erfüllbarkeit **a)** α heißt *erfüllbar* genau dann, wenn eine Belegung $\mathcal{I}$ von α existiert mit $\mathcal{I}^*(\alpha) = 1$.

Tautologie **b)** α heißt *Tautologie* oder *allgemeingültig* (auch *gültig*) genau dann, wenn für jede Belegung $\mathcal{I}$ von α gilt $\mathcal{I}^*(\alpha) = 1$.

Kontradiktion **c)** α heißt *Kontradiktion* oder *widerspruchsvoll* (auch *unerfüllbar*) genau dann, wenn für jede Belegung $\mathcal{I}$ von α gilt $\mathcal{I}^*(\alpha) = 0$.

Modell **d)** $\mathcal{F}$ heißt erfüllbar genau dann, wenn es eine Belegung $\mathcal{I}$ von $\mathcal{F}$ gibt, so dass $\mathcal{I}^*(\gamma) = 1$ für alle $\gamma \in \mathcal{F}$ ist. $\mathcal{I}$ heißt ein *Modell* für $\mathcal{F}$. Gibt es zu $\mathcal{F}$ kein Modell,

Unerfüllbarkeit dann heißt $\mathcal{F}$ *unerfüllbar*. □

Beispiel 1.8 **a)** Die Formeln $p \wedge q$ und $(p \wedge q) \vee (q \to r)$ sowie die Formel aus Beispiel 1.7 sind erfüllbar aber keine Tautologien.

b) Die Formeln $p \vee \neg p$ und $(p \to q) \leftrightarrow (\neg p \vee q)$ sind Tautologien.

c) Die Formel $p \wedge \neg p$ ist eine Kontradiktion.

d) Die Menge $\mathcal{F}_1 = \{\, p \vee q, q \wedge \neg r, (p \wedge q) \vee (q \to r) \,\}$ ist erfüllbar, denn sie besitzt, wie bereits in Beispiel 1.7 festgestellt, das Modell $\mathcal{I}(p) = \mathcal{I}(q) = 1$, $\mathcal{I}(r) = 0$.

e) Die Menge $\mathcal{F}_2 = \{\, p, p \to q, \neg q \,\}$ ist unerfüllbar, denn sie besitzt kein Modell, da für eine Belegung $\mathcal{I}$, für die jede Formel in $\mathcal{F}_2$ den Wert 1 haben soll, $\mathcal{I}(p) = 1$ und $\mathcal{I}(q) = 0$ sein muss. Für diese Belegung ist aber $\mathcal{I}(p \to q) = 0$. Es gibt also keine Belegung, die $\mathcal{F}_2$ erfüllt. □

Folgerung 1.2 **a)** Eine Formel ist genau dann erfüllbar, wenn in der Ergebnisspalte ihrer Wahrheitstafel mindestens eine 1 vorkommt.

b) Eine Formel ist genau dann allgemeingültig, wenn in der Ergebnisspalte ihrer Wahrheitstafel nur Einsen vorkommen.

c) Eine Formel ist genau dann widerspruchsvoll, wenn in der Ergebnisspalte ihrer Wahrheitstafel nur Nullen vorkommen. □

Übungsaufgaben

1.6 (1) Zeigen Sie die Allgemeingültigkeit der Formel

$$(\alpha \rightarrow \beta) \rightarrow (\neg\beta \rightarrow \neg\alpha)$$

für α,$\beta \in \mathcal{A}$!

(2) Zeigen Sie, dass die Bauernregel „Wenn der Hahn kräht auf dem Mist, dann ändert sich das Wetter oder es bleibt, wie es ist“, eine Tautologie ist!

(3) Beweisen Sie die Beispiele 1.8 a - c) auf Seite 20! □

1.2.4 Zusammenfassung

Die Aussagenlogik kann als eine formale Sprache betrachtet werden. Aussagenlogische Konstanten und Variablen werden mit aussagenlogischen Operatoren zu aussagenlogischen Formeln verknüpft. Mithilfe der Wahrheitswerte 0 und 1 sowie darauf definierten Operationen min und „$-$“ oder max und „$-$“ kann mithilfe des Interpretationsoperators nach Belegung der Variablen eines aussagenlogischen Ausdrucks mit diesen Werten der Wahrheitswert des Ausdrucks berechnet werden. Alternativ kann der Wahrheitswert eines aussagenlogischen Ausdrucks mithilfe einer Wahrheitstabelle berechnet werden. Grundlage dafür ist in jedem Fall die Definition der Wahrheitswerte für grundlegende Operationen wie Negation, Disjunktion, Konjunktion, Subjunktion, Bijunktion und exklusives Oder. Einige dieser Operationen können mithilfe der anderen ausgedrückt werden.

Eine Menge von Formeln $\mathcal{F}$ heißt erfüllbar, wenn es eine Belegung der Variablen der Formeln gibt, die alle Formeln in $\mathcal{F}$ wahr machen. Eine solche Belegung heißt Modell für $\mathcal{F}$. Eine Formelmenge heißt allgemeingültig oder Tautologie, falls alle möglichen Belegungen Modelle sind. Besitzt eine Formelmenge kein Modell, dann heißt sie unerfüllbar oder Kontradiktion.

1.3 Logische Folgerungen und Implikationen

Für das Beweisen von mathematischen Behauptungen sowie für die logische Programmierung und die Wissensverarbeitung ist der logische Folgerungsbegriff elementar. Es geht darum, aus der Gültigkeit von Aussagen weitere gültige Aussagen abzuleiten. Wir werden Methoden für zwei grundsätzlich verschiedene Schlussweisen betrachten: semantische und syntaktische Folgerungen. Semantische Schlussfolgerungen basieren darauf, dass die Gültigkeit von Aussagen unter der Voraussetzung von gegebenen Aussagen mithilfe von Interpretationen oder Wahrheitstafeln berechnet wird. Syntaktische Schlussfolgerungen basieren auf Ableitungsregeln, die als korrekte semantische Schlussfolgerungen angenommen werden. Die Ableitung einer Formel aus einer Formelmenge geschieht dann mithilfe der Ableitungsregeln durch einen Ersetzungsmechanismus, bei dem Formelmengen durch Formelmengen ersetzt werden, ohne deren Semantik zu betrachten.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- die Begriffe logische Folgerung und Implikation erläutern sowie deren Aquivalenz beweisen können,
- Methoden für die Berechnung logischer Folgerungen sowie elementare logische Folgerungen kennen und anwenden können,
- wissen, was eine axiomatisierbare, logische Theorie ist,
- die Grundidee logischer Kalküle erklären können,
- Widerspruchsfreiheit und Vollständigkeit als wesentliche Qualtitätskriterien von Kalkülen begreifen.

1.3.1 Logische Folgerung

Logische Folgerung

Definition 1.2 Sei $\alpha \in \mathcal{A}$ eine aussagenlogische Formel und $\mathcal{F}$ eine endliche Menge aussagenlogischer Formeln aus $\mathcal{A}$. α heißt *logische Folgerung* von $\mathcal{F}$ genau dann, wenn $\mathcal{I}^*(\alpha) = 1$ für jedes Modell $\mathcal{I}$ von $\mathcal{F}$ ist. Wir schreiben $\mathcal{F} \models \alpha$ und sprechen „aus $\mathcal{F}$ folgt α (logisch)“. □

Beispiel 1.9 **a)** Für $\mathcal{F} = \{ p, q \}$ gilt $\mathcal{F} \models p \wedge q$, denn für jedes Modell $\mathcal{I}$ von $\mathcal{F}$ muss gelten $\mathcal{I}(p) = 1$ und $\mathcal{I}(q) = 1$, damit ist aber auch $\mathcal{I}^*(p \wedge q) = 1$.

b) Für $\mathcal{F} = \{ p \rightarrow q, q \rightarrow r \}$ gilt $\mathcal{F} \models p \rightarrow r$, denn $\mathcal{F}$ besitzt die folgenden vier Modelle

$$\begin{array}{llll} (1) & \mathcal{I}(p) = 1 & \mathcal{I}(q) = 1 & \mathcal{I}(r) = 1 \\ (2) & \mathcal{I}(p) = 0 & \mathcal{I}(q) = 1 & \mathcal{I}(r) = 1 \\ (3) & \mathcal{I}(p) = 0 & \mathcal{I}(q) = 0 & \mathcal{I}(r) = 1 \\ (4) & \mathcal{I}(p) = 0 & \mathcal{I}(q) = 0 & \mathcal{I}(r) = 0 \end{array}$$

und für alle diese Modelle gilt jeweils $\mathcal{I}^*(p \to r) = 1$. Aus den Formeln $p \to q$ und $q \to r$ folgt also logisch die Formel $p \to r$.

c) Für $\mathcal{F} = \{p \to r, q \vee r\}$ gilt $\mathcal{F} \not\models p \wedge r$, denn die Belegung $\mathcal{I}(p) = 0$, $\mathcal{I}(q) = \mathcal{I}(r) = 1$ ist ein Modell von $\mathcal{F}$, aber es ist $\mathcal{I}^*(p \wedge r) = 0$ für diese Belegung. Aus $\mathcal{F}$ kann also die Formel $p \wedge q$ nicht logisch gefolgert werden. □

Der folgende Satz macht eine Aussage über den Zusammenhang von Allgemeingültigkeit und Unerfüllbarkeit von Formeln. Diese Aussage ist wichtig für die Programmierung von automatischen Beweisern, welche die Gültigkeit von Formeln nachweisen sollen.

Satz 1.1 Sei $\mathcal{F} = \{ \alpha_1, \alpha_2, \ldots, \alpha_n \}$ eine Menge aussagenlogischer Formeln und $\beta \in \mathcal{A}$. Dann gilt $\mathcal{F} \models \beta$ genau dann, wenn $\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\beta \}$ unerfüllbar ist.

Beweis Wir beweisen die Behauptung in zwei Schritten, indem wir zeigen:

(i) Wenn $\mathcal{F} \models \beta$ gilt, dann ist $\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\beta \}$ unerfüllbar.

(ii) Wenn $\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\beta \}$ unerfüllbar ist, dann gilt $\mathcal{F} \models \beta$.

Zu (i): Sei $\mathcal{I}$ ein Modell für $\mathcal{F}$, d.h. $\mathcal{I}^*(\alpha_i) = 1$ für alle i. Da $\mathcal{F} \models \beta$ gilt, ist $\mathcal{I}^*(\beta) = 1$ für $\mathcal{I}$, d.h. $\mathcal{I}^*(\neg\beta) = 0$ für $\mathcal{I}$. Somit gibt es also keine Belegung $\mathcal{I}$, für die $\mathcal{I}^*(\alpha_i) = 1$ für alle i und $\mathcal{I}^*(\neg\beta) = 1$ ist. Daraus folgt, dass $\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\beta \}$ unerfüllbar ist.

Zu (ii): Sei $\mathcal{I}$ ein Modell für $\mathcal{F}$, es ist also $\mathcal{I}^*(\alpha_i) = 1$ für alle i. Da $\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\beta \}$ unerfüllbar ist, kann $\mathcal{I}$ kein Modell für $\neg\beta$ sein, d.h. es gilt $\mathcal{I}^*(\neg\beta) = 0$ und damit gilt $\mathcal{I}^*(\beta) = 1$ für die Belegung $\mathcal{I}$. Hieraus folgt, dass jedes Modell von $\mathcal{F}$ ein Modell von β ist, d.h. $\mathcal{F} \models \beta$ gilt. □

Der Satz gibt im Übrigen eine Beweismethode an: Um zu zeigen, dass die Behauptung β aus den Voraussetzungen $\mathcal{F}$ logisch folgt, kann man zeigen, dass die Negation der Behauptung mit den Voraussetzungen unvereinbar ist.

Folgerung 1.3 **a)** Eine Formel $\beta \in \mathcal{A}$ ist allgemeingültig genau dann, wenn $\neg\beta$ unerfüllbar ist.

b) $\beta \in \mathcal{A}$ ist allgemeingültig genau dann, wenn $\emptyset \models \beta$ gilt.[5]

c) Ist $\mathcal{F}$ eine unerfüllbare Menge von Formeln, dann gilt $\mathcal{F} \models \beta$ für jede Formel $\beta \in \mathcal{A}$. Dies bedeutet, dass man aus jeder unerfüllbaren Formelmenge jede beliebige Formel folgern kann.

Beweis **a)** folgt unmittelbar aus Definition 1.1 b) und c).

b) folgt aus a) und Satz 1.1, wir brauchen nur $\mathcal{F} = \emptyset$ zu wählen.

c) Sei $\mathcal{F} = \{\alpha_1, \ldots, \alpha_n\}$ unerfüllbar, d.h. $\mathcal{F}$ besitzt kein Modell. Es folgt, dass die Formelmenge $\{\alpha_1, \ldots, \alpha_n, \neg\beta\}$ unerfüllbar bleibt, denn selbst, wenn $\neg\beta$ erfüllbar wäre, wären die erfüllenden Belegungen kein Modell für $\mathcal{F}$, da $\mathcal{F}$

5 Anstelle von $\emptyset \models \beta$ schreibt man auch $\models \beta$.

kein Modell besitzt. Aus der Unerfüllbarkeit der Formelmenge $\{\alpha_1, \ldots, \alpha_n, \neg\beta\}$ folgt mit dem Satz 1.1 die zu zeigende Behauptung $\mathcal{F} \models \beta$. □

Der nächste Satz gibt weitere Beweismethoden an: die *Deduktion* und die *Modus ponens-Regel*.

Deduktionstheorem

Satz 1.2 **a)** Für jede Menge $\mathcal{F} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ aussagenlogischer Formeln und für alle Formeln $\beta, \gamma \in \mathcal{A}$ gilt

$$\{\alpha_1, \alpha_2, \ldots, \alpha_n, \beta\} \models \gamma \text{ genau dann, wenn } \mathcal{F} \models (\beta \to \gamma)$$

gilt.

Modus ponens-Regel

b) Für alle Formeln $\alpha, \beta \in \mathcal{A}$ gilt $\{\alpha, \alpha \to \beta\} \models \beta$.

Beweis **a)** Es gilt:[6]

$\{\alpha_1, \ldots, \alpha_n\} \models \beta \to \gamma$

gdw. $\{\alpha_1, \ldots, \alpha_n, \neg(\beta \to \gamma)\}$ unerfüllbar	wegen Satz 1.1
gdw. $\{\alpha_1, \ldots, \alpha_n, \beta \wedge \neg\gamma\}$ unerfüllbar	wegen Gleichung (1.7), S. 19
gdw. $\{\alpha_1, \ldots, \alpha_n, \beta, \neg\gamma\}$ unerfüllbar	wegen Beispiel 1.9 a)
gdw. $\{\alpha_1, \ldots, \alpha_n, \beta\} \models \gamma$	wegen Satz 1.1

b) Es gilt mit Satz 1.1:

$$\{\alpha, \alpha \to \beta\} \models \beta \text{ gdw. } \{\alpha, \alpha \to \beta, \neg\beta\} \text{ unerfüllbar}$$

Da wegen Beispiel 1.8 e) $\{\alpha, \alpha \to \beta, \neg\beta\}$ unerfüllbar ist, ist die Behauptung gezeigt. □

Satz 1.3 Es seien $\mathcal{F}$ eine Menge aussagenlogischer Formeln und $\alpha \in \mathcal{A}$. Dann gelten die folgenden Aussagen:

a) Gilt $\mathcal{F} \models \alpha$, dann auch $\mathcal{F} \cup \{\beta\} \models \alpha$ für alle Formeln $\beta \in \mathcal{A}$.[7]

b) Gilt $\mathcal{F} \models \alpha$ und ist $\beta \in \mathcal{A}$ allgemeingültig, dann gilt $\mathcal{F} - \{\beta\} \models \alpha$.[8] □

Übungsaufgaben

1.7 Beweisen Sie Satz 1.3! □

6 Wir kürzen „genau dann, wenn“ durch „gdw.“ ab.
7 $\mathcal{F} \cup \{\beta\}$ bedeutet, dass die Formel β zur Formelmenge $\mathcal{F}$ hinzugefügt wird.
8 $\mathcal{F} - \{\beta\}$ bedeutet, dass aus der Menge $\mathcal{F}$ das Element β entfernt wird. Ist $\beta \notin \mathcal{F}$, dann ist $\mathcal{F} = \mathcal{F} - \{\beta\}$.

1.3.2 Implikation

Neben der logischen Folgerung betrachten wir als weiteren Folgerungsbegriff die Implikation.

Implikation

Definition 1.3 Gilt für aussagenlogische Formeln $\alpha_1, \alpha_2, \ldots, \alpha_n$ und β, dass die Subjunktion $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ eine Tautologie ist, dann heißt diese Subjunktion *Implikation*, und wir schreiben $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \Rightarrow \beta$ und sprechen „$\alpha_1, \alpha_2, \ldots, \alpha_n$ implizieren β“. □

Beispiel 1.10 Es seien $\alpha, \beta, \gamma \in \mathcal{A}$, dann gelten die folgenden Implikationen (rechnen Sie nach!):

a) *Abschwächung der Nachbedingung*: $\alpha \Rightarrow (\alpha \vee \beta)$.

Abschwächung der Nachbedingung

b) *Verschärfung der Vorbedingung*: $(\alpha \wedge \beta) \Rightarrow \alpha$ (vergleiche Satz 1.3 a).

Verschärfung der Vorbedingung

c) *Kettenschluss*: $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \gamma) \Rightarrow (\alpha \rightarrow \gamma)$. □

Kettenschluss

Bemerkung 1.1 Auf den ersten Blick scheinen die Symbole „$\rightarrow$“ und „$\Rightarrow$“ dasselbe zu bedeuten. Das ist aber nicht der Fall. Die Subjunktion $\rightarrow$ ist eine boolesche Operation, die zwei logische Formeln miteinander verknüpft. Die Implikation $\Rightarrow$ ist eine Aussage über die Subjunktion zweier aussagenlogischer Formeln. $\rightarrow$ ist ein Symbol in der Sprache der Aussagenlogik; das Symbol $\Rightarrow$ verwenden wir *metasprachlich*, um eine Aussage über eine Eigenschaft aussagenlogischer Formeln zu machen. □

Der folgende Satz besagt, dass die logische Folgerung und die Implikation in dem Sinne äquivalente Folgerungsbegriffe sind, als dass die logische Folgerung

$$\{ \alpha_1, \alpha_2, \ldots, \alpha_n \} \models \beta$$

auch nachgewiesen werden kann, indem man zeigt, dass die Implikation

$$(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \Rightarrow \beta$$

gilt, und umgekehrt.

Satz 1.4 Für die aussagenlogischen Formeln $\alpha_1, \alpha_2, \ldots, \alpha_n$ und β gilt

$$\{ \alpha_1, \alpha_2, \ldots, \alpha_n \} \models \beta \quad \text{genau dann, wenn} \quad (\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \Rightarrow \beta$$

gilt.

Beweis Wir setzen $\mathcal{F} = \{ \alpha_1, \alpha_2, \ldots, \alpha_n \}$ und zeigen:

(i) Wenn $\mathcal{F} \models \beta$ gilt, dann gilt auch $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \Rightarrow \beta$.

(ii) Wenn $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \Rightarrow \beta$ gilt, dann auch $\mathcal{F} \models \beta$.

Zu (i): Für alle Modelle $\mathcal{I}$ von $\mathcal{F}$ ist $\mathcal{I}^*(\alpha_i) = 1$ für $1 \leq i \leq n$. Für diese Modelle muss, da $\mathcal{F} \models \beta$ vorausgesetzt ist, $\mathcal{I}^*(\beta) = 1$ sein. Für diese Belegungen ist die Subjunktion $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ immer wahr. Für Belegungen von $\mathcal{F}$, die keine Modelle sind, ist $\mathcal{I}^*(\alpha_i) = 0$ für mindestens ein i. Dann gilt $\mathcal{I}^*(\alpha_1 \wedge \alpha_2 \wedge$

$\ldots \wedge \alpha_n) = 0$ und damit ist die Subjunktion $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ ebenfalls wahr und zwar unabhängig vom Wert von β. Die Subjunktion $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ ist also in jedem Falle wahr und damit eine Implikation, was zu zeigen war.

Zu (ii): $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$ ist eine Tautologie genau dann, wenn entweder alle α_i und β den Wert 1 haben oder mindestens ein α_i den Wert 0 hat. Hieraus folgt, dass in beiden Fällen die Formelmenge $\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\beta \}$ unerfüllbar ist. Gemäß Satz 1.1 folgt daraus, dass $\mathcal{F} \models \beta$ gilt, was zu zeigen war. □

1.3.3 Kalküle

Die beiden bisher vorgestellten Folgerungsbegriffe, *logische Folgerung* (Definition 1.2) und *Implikation* (Definition 1.3), sind semantische Folgerungsbegriffe. Das soll heißen, dass in beiden Fällen die Werte der beteiligten Formeln berechnet werden müssen um zu entscheiden, ob die gegebenen Formeln eine logische Folgerung bzw. eine Implikation bilden. Es ist also notwendig, die Interpretationen der beteiligten Formeln bzw. die Wahrheitstafel der betreffenden Subjunktion zu betrachten.

Wir wollen nun eine andere Möglichkeit der Folgerung, die syntaktische Folgerung, betrachten. Syntaktische Folgerung heißt, dass eine Folgerung vorgenommen wird, ohne die Semantik der beteiligten Formeln, sei es durch Interpretationen oder sei es durch die Wahrheitstafel, zu berechnen. Die Folgerung geschieht, indem in einer Formel Teilformeln durch andere Formeln ersetzt werden. Die Ersetzung von Formeln geschieht dabei mithilfe sogenannter Ableitungsregeln, auch Inferenzregeln genannt.

Ableitungsregel
Inferenzregel

Definition 1.4 **a)** Seien $\alpha_1, \alpha_2, \ldots, \alpha_n$ und β ausagenlogische Formeln, für die die Implikation $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \Rightarrow \beta$ gilt. Dann heißt

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\beta}$$

Prämisse
Konklusion

Ableitungs- oder *Inferenzregel*. Die Formelmenge $\{\alpha_1, \ldots, \alpha_n\}$ heißt *Prämisse* und γ heißt *Konklusion* dieser Inferenzregel.

b) Sei γ eine aussagenlogische Formel, $\mathcal{F} = \{ \alpha_1, \alpha_2, \ldots, \alpha_n \}$ eine Menge aussagenlogischer Formeln, $\{ \beta_1, \beta_2, \ldots, \beta_k \}$ irgendeine Auswahl von Formeln aus $\mathcal{F}$ und $\{ \gamma_1, \gamma_2, \ldots, \gamma_m \}$ die Menge der nicht ausgewählten Formeln aus $\mathcal{F}$ sowie

$$\frac{\beta_1, \beta_2, \ldots, \beta_k}{\gamma} \tag{1.8}$$

eine Inferenzregel, dann heißt $\{ \gamma_1, \gamma_2, \ldots, \gamma_m, \gamma \}$ *ableitbar* aus $\mathcal{F}$, und wir schreiben

$$\mathcal{F} \vdash \{\gamma_1, \gamma_2, \ldots, \gamma_m, \gamma\}$$

In $\mathcal{F}$ wird quasi die Prämisse der Inferenzregel (1.8) durch deren Konklusion ersetzt.

c) Eine aussagenlogische Formel γ ist ableitbar aus einer Menge $\mathcal{F}$ von aussagenlogischen Formeln, falls es Mengen aussagenlogischer Formeln $\mathcal{F}_1$, $\mathcal{F}_2$, ..., $\mathcal{F}_r$, $r \geq 0$, gibt mit: **Logische Ableitung**

$$\mathcal{F} \vdash \mathcal{F}_1 \vdash \mathcal{F}_2 \vdash \ldots \vdash \mathcal{F}_r \vdash \{\gamma\}$$

Wir notieren dann $\mathcal{F} \vdash \gamma$ und sagen, dass γ *logisch aus* $\mathcal{F}$ *ableitbar* ist. □

Beispiel 1.11 Es seien $\alpha, \beta, \gamma \in \mathcal{A}$. a) - d) sind Beispiele für Ableitungsregeln. e) ist ein Beispiel für eine logische Ableitung mithilfe der Regeln a) und b).

a) *Modus ponens-Regel:* **Modus ponens**

$$\frac{\alpha, \alpha \rightarrow \beta}{\beta}$$

(siehe auch Satz 1.2 b).

b) *Modus tollens-Regel:* **Modus tollens**

$$\frac{\alpha \rightarrow \beta,\ \neg\beta}{\neg\alpha}$$

c) *Reductio ad absurdum-Regel:* **Reductio ad absurdum**

$$\frac{(\gamma \vee \alpha) \rightarrow \beta,\ (\gamma \vee \alpha) \rightarrow \neg\beta}{\neg\alpha}$$

d) *Kettenschluss:* **Kettenschluss**

$$\frac{\alpha \rightarrow \beta, \beta \rightarrow \gamma}{\alpha \rightarrow \gamma}$$

(siehe auch Beispiel 1.10 c) auf Seite 25).

e) Es gilt

$$\{\, \alpha \rightarrow \beta, \neg\beta, \neg\alpha \rightarrow \gamma \,\} \vdash \{\, \neg\alpha, \neg\alpha \rightarrow \gamma \,\} \vdash \{\gamma\}$$

und damit

$$\{\, \alpha \rightarrow \beta, \neg\beta, \neg\alpha \rightarrow \gamma \,\} \vdash \gamma$$

Die erste Ableitung erfolgt mithilfe der Modus tollens-Regel, die zweite mithilfe der Modus ponens-Regel. □

Die logische Ableitung geschieht also, indem eine Menge von aussagenlogischen Formeln aufgrund von Inferenzregeln oder bereits durchgeführten logischen Ableitungen verändert wird. Dabei wird die Semantik der Formeln bei keinem Ableitungsschritt betrachtet. Die korrekte Semantik wird nur – einmalig – für die benutzten Inferenzregeln vorausgesetzt (Definition 1.4 a). Solche syntaktischen Ableitungssysteme werden auch *Kalküle* genannt. Kalküle sind gut geeignet für die Programmierung von logischen Schlussfolgerungsmechanismen auf Rechnern. Dabei sollte ein aussagenlogischer Kalkül den folgenden beiden Qualitätskriterien mindestens genügen: **Logischer Kalkül**

- *Widerspruchsfreiheit* (auch *Korrektheit*): Jede mit dem Kalkül ableitbare Formel ist eine logische Folgerung. Ist also $\mathcal{F}$ eine Menge aussagenlogischer Formeln, α eine aussagenlogische Formel und gilt $\mathcal{F} \vdash \alpha$, dann folgt, dass auch $\mathcal{F} \models \alpha$ gilt. Der Kalkül kann also keine Folgerungen produzieren, die semantisch nicht korrekt sind.

Widerspruchsfreiheit Korrektheit

Mit Folgerung 1.3 (siehe Seite 23) gilt für einen widerspruchsfreien Kalkül: Alle ableitbaren Formeln sind allgemeingültig, d.h., gilt $\vdash \alpha$, dann gilt auch $\models \alpha$.

Vollständigkeit

- *Vollständigkeit:* Jede logische Folgerung ist auch mit dem Kalkül ableitbar. Ist also $\mathcal{F}$ eine Menge aussagenlogischer Formeln, α eine aussagenlogische Formel und gilt $\mathcal{F} \models \alpha$, dann folgt, dass auch $\mathcal{F} \vdash \alpha$ gilt. Der Kalkül kann also alle logischen Folgerungen syntaktisch ableiten.

  Mit Folgerung 1.3 gilt für einen vollständigen Kalkül: Alle allgemeingültigen Formeln sind ableitbar, d.h., gilt $\models \alpha$, dann gilt auch $\vdash \alpha$.

Für einen widerspruchsfreien und vollständigen aussagenlogischen Kalkül gilt also $\mathcal{F} \vdash \alpha$ genau dann, wenn $\mathcal{F} \models \alpha$ gilt, bzw. $\vdash \alpha$ gilt genau dann, wenn $\models \alpha$ gilt.

In Kapitel 1.5 stellen wir einen widerspruchsfreien und vollständigen Kalkül für die Aussagenlogik vor.

1.3.4 Theorien

In der Logik, der Mathematik und in vielen Bereichen der Künstlichen Intelligenz (Wissensverarbeitung, Robotik) ist man daran interessiert, aus möglichst wenigen Grundannahmen, den sogenannten Axiomen, weitere Aussagen zu schlussfolgern, um zu einem umfangreichen Wissen, einer Theorie, zu gelangen.

Ein klassisches Beispiel dafür ist die Euklidische Geometrie. Euklid machte fünf fundamentale Annahmen über Punkte und Linien in der Ebene. Auf dieser Basis ist durch logische Schlussfolgerungen ein immenses Wissen entstanden, das wiederum im Verlauf der Jahrhunderte Grundlage für vielfältige wissenschaftliche Erkenntnisse und technische Entwicklungen gewesen ist.

Abschluss unter logischer Folgerung

Definition 1.5 **a)** Sei $\mathcal{T}$ eine Menge aussagenlogischer Formeln. $\mathcal{T}$ heißt unter *logischer Folgerung abgeschlossen* genau dann, wenn gilt: Ist $\alpha \in \mathcal{A}$ und gilt $\mathcal{T} \models \alpha$, dann ist $\alpha \in \mathcal{T}$.

Theorie

b) Eine unter logischer Folgerung abgeschlossene Menge $\mathcal{T}$ logischer Formeln heißt eine *Theorie*. □

Wie oben erwähnt, ist man an axiomatisierbaren Theorien interessiert, bei denen aus gegebenen (angenommenen) Grundaussagen alle gültigen Aussagen durch logische Schlussfolgerung berechnet werden können.

Definition 1.6 **a)** Eine Theorie $\mathcal{T}$ heißt *axiomatisierbar* genau dann, wenn eine Menge $\mathcal{A}_\mathcal{T}$ von Formeln existiert mit $\mathcal{T} = \{\alpha \mid \mathcal{A}_\mathcal{T} \models \alpha\}$.

Axiomatisierbare Theorie

b) $\mathcal{A}_\mathcal{T}$ ist die Menge der *Axiome* von $\mathcal{T}$.

Axiome

c) Ist $\mathcal{A}_\mathcal{T}$ endlich, dann heißt $\mathcal{T}$ *endlich axiomatisierbar*. □

Die Euklidische Geometrie ist ein Beispiel für eine endlich axiomatisierbare Theorie. Die Arithmetik natürlicher Zahlen ist eine nicht endlich axiomatisierbare Theorie. Sie basiert auf den Peano-Axiomen (siehe Abschnitt 3.1.1). Die Anzahl der Peano-Axiome ist zwar endlich, aber eines davon, das so genannte Induktionsaxiom, ist ein Schema, das bei seiner Anwendung eine Annahme über alle natürlichen Zahlen zugrunde legt.

1.3.5 Zusammenfassung

Für das Beweisen von Behauptungen in Mathematik und Informatik sowie für das Ableiten von Wissen aus Wissensbasen (Logikprogrammierung, Expertensysteme, Wissensmanagementsysteme) sind Schlussfolgerungsmechanismen von Bedeutung. Eine Formel kann aus einer Formelmenge logisch gefolgert werden, wenn alle Modelle der Formelmenge auch Modelle der Formel sind. Durch Implikationen, das sind immer wahre Subjunktionen, ist ein weiterer semantischer Folgerungsbegriff gegeben. Die logische Folgerung und die Implikation sind semantische Folgerungsbegriffe, weil die Folgerung auf Interpretationen bzw. auf Wahrheitstafeln beruht, d.h. die Belegungen der Formelmengen und Formeln werden betrachtet. Logische Folgerung und Implikation sind äquivalente Folgerungsbegriffe.

Mithilfe von Kalkülen werden Formeln aus gegebenen Formelmengen basierend auf semantisch als gültig bewiesenen Inferenzregeln mithilfe eines syntaktischen Ableitungsbegriffes hergeleitet. Ohne Belegungen und Interpretationen oder Wahrheitstafeln zu betrachten, werden Prämissen durch Konklusionen substituiert. Bedeutende Inferenzregeln sind Modus ponens, Modus tollens, Reductio ad absurdum und der Kettenschluss. Für Kalküle sind Widerspruchsfreiheit (Korrektheit) und Vollständigkeit wesentliche Anforderungen. Ein Kalkül ist korrekt, wenn jede aus einer Formelmenge $\mathcal{F}$ mit dem Kalkül, also syntaktisch ableitbare Formel α auch eine logische Folgerung aus $\mathcal{F}$ ist: Gilt $\mathcal{F} \vdash \alpha$, dann gilt auch $\mathcal{F} \models \alpha$. Ein Kalkül ist vollständig, wenn jede Formel α, die logisch aus einer Formelmenge $\mathcal{F}$, also semantisch gefolgert werden kann, auch mit dem Kalkül abgeleitet werden kann: Gilt $\mathcal{F} \models \alpha$, dann gilt auch $\mathcal{F} \vdash \alpha$.

Eine unter logischer Folgerung abgeschlossene Menge von Formeln nennt man eine Theorie. Können alle Formeln der Theorie aus einer Teilmenge von Formeln abgeleitet werden, dann heißt die Theorie axiomatisierbar, und die Elemente der Teilmenge heißen Axiome der Theorie.

1.4 Äquivalenzen, Basen und Normalformen

In diesem Kapitel betrachten wir weitere (letztendlich alle) zweistelligen aussagenlogischen Verknüpfungen und greifen die Frage auf, ob es Teilmengen davon gibt, mit denen alle diese ausgedrückt werden können. Ausgangspunkt dafür ist der Begriff der Äquivalenz von aussagenlogsichen Formeln.

Für bestimmte Betrachtungen und Anwendungen ist es von Vorteil, wenn aussagenlogische Formeln eine feste syntaktische Struktur besitzen. Wir betrachten zwei solcher Formate: die disjunktive und die konjunktive Normalform.

Lernziele

Nach Durcharbeiten dieses Kapitels sollten Sie

- Definitionen für die semantische Äquivalenz aussagenlogischer Formeln erklären können,
- die Äquivalenz aussagenlogischer Aussagen sowohl semantisch als auch mithilfe syntatkischer Substitutionen beweisen können,
- „gängige“ Äquivalenzen kennen,
- den Begriff der aussagenlogischen Basis erklären können,
- aussagenlogische Basen kennen und beweisen können,
- disjunktive und konjunktive Normalformen erklären und aussagenlogische Formeln in diese Formen umwandeln können.

1.4.1 Aussagenlogische Äquivalenzen

Interessant für „das Rechnen“ mit aussagenlogischen Formeln ist, in ihnen Teilformeln durch gleichwertige Teilformeln ersetzen zu können, um „kalkülmäßig“ – also rein syntaktisch – Formeln in gleichwertige Formeln transformieren zu können. Dazu müssen wir zunächst die Gleichwertigkeit oder Äquivalenz von aussagenlogischen Formeln definieren.

Aussagenlogische Äquivalenz

Definition 1.7 Zwei aussagenlogische Formeln $\alpha,\beta \in \mathcal{A}$ heißen *logisch äquivalent*, falls für jede Belegung $\mathcal{I}$ von α und β gilt: $\mathcal{I}^*(\alpha) = \mathcal{I}^*(\beta)$. Schreibweise: $\alpha \equiv \beta$. □

Beispiel 1.12 Aus Folgerung 1.1 a) (Seite 18) folgt unmittelbar, dass für $\alpha,\beta \in \mathcal{A}$ folgende Äquivalenzen gelten:

$$\alpha \to \beta \equiv \neg\alpha \vee \beta$$
$$\alpha \leftrightarrow \beta \equiv (\alpha \to \beta) \wedge (\beta \to \alpha)$$
$$\alpha \oplus \beta \equiv (\alpha \wedge \neg\beta) \vee (\neg\alpha \wedge \beta)$$

Der folgende Satz listet wichtige aussagenlogische Äquivalenzen auf.

Satz 1.5 Es seien $\alpha, \beta, \gamma \in \mathcal{A}$ aussagenlogische Formeln. Dann gelten die folgenden Äquivalenzen:

Kommutativität

$$\alpha \vee \beta \equiv \beta \vee \alpha$$
$$\alpha \wedge \beta \equiv \beta \wedge \alpha$$
$$\alpha \oplus \beta \equiv \beta \oplus \alpha$$
$$\alpha \leftrightarrow \beta \equiv \beta \leftrightarrow \alpha$$

Assozitivität

$$\alpha \vee (\beta \vee \gamma) \equiv (\alpha \vee \beta) \vee \gamma$$
$$\alpha \wedge (\beta \wedge \gamma) \equiv (\alpha \wedge \beta) \wedge \gamma$$
$$\alpha \oplus (\beta \oplus \gamma) \equiv (\alpha \oplus \beta) \oplus \gamma$$
$$\alpha \leftrightarrow (\beta \leftrightarrow \gamma) \equiv (\alpha \leftrightarrow \beta) \leftrightarrow \gamma$$

Distributivität

$$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$
$$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

De Morgansche Regeln

$$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$$
$$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$$

Einführung der Negation

$$\neg\alpha \equiv \alpha \rightarrow 0$$
$$\neg\alpha \equiv \alpha \leftrightarrow 0$$
$$\neg\alpha \equiv \alpha \oplus 1$$

Doppelte Negation

$$\neg\neg\alpha \equiv \alpha$$

Idempotenz

$$\alpha \vee \alpha \equiv \alpha$$
$$\alpha \wedge \alpha \equiv \alpha$$
$$\alpha \oplus 0 \equiv \alpha$$

Absorption

$$1 \wedge \alpha \equiv \alpha$$
$$0 \vee \alpha \equiv \alpha$$
$$1 \rightarrow \alpha \equiv \alpha$$
$$1 \leftrightarrow \alpha \equiv \alpha$$
$$\alpha \vee (\alpha \wedge \beta) \equiv \alpha$$
$$\alpha \wedge (\alpha \vee \beta) \equiv \alpha$$

Tautologien

$$1 \vee \alpha \equiv 1$$
$$\neg\alpha \vee \alpha \equiv 1$$
$$\alpha \rightarrow \alpha \equiv 1$$
$$\alpha \rightarrow 1 \equiv 1$$
$$0 \rightarrow \alpha \equiv 1$$
$$\alpha \leftrightarrow \alpha \equiv 1$$

Unerfüllbarkeitsregeln

$$0 \wedge \alpha \equiv 0$$
$$\neg\alpha \wedge \alpha \equiv$$
$$\alpha \oplus \alpha \equiv 0$$

Kontraposition

$$\alpha \rightarrow \beta \equiv \neg\beta \rightarrow \neg\alpha$$

Übungsaufgaben

1.8 Beweisen Sie die im Satz 1.5 aufgelisteten logischen Äquivalenzen! □

Äquivalenz

Definition 1.8 Gilt für aussagenlogische Formeln α und β, dass die Bijunktion $\alpha \leftrightarrow \beta$ eine Tautologie ist, dann heißt diese Bijunktion eine *Äquivalenz*, und wir schreiben $\alpha \Leftrightarrow \beta$. □

Den Zusammenhang zwischen logischer Äquivalenz (siehe Definition 1.7) und Äquivalenz (Definition 1.8) macht der folgende Satz klar (vergleiche auch Satz 1.4 auf Seite 25 sowie Bemerkung 1.1).

Satz 1.6 Seien $\alpha,\beta \in \mathcal{A}$ aussagenlogische Formeln, dann gilt

$$\alpha \equiv \beta \text{ genau dann, wenn } \alpha \Leftrightarrow \beta$$

gilt. □

Folgerung 1.4 Eine aussagenlogische Formel $\alpha \in \mathcal{A}$ ist allgemeingültig genau dann, wenn $\alpha \equiv 1$ (oder $\alpha \Leftrightarrow 1$) gilt. □

Übungsaufgaben

1.9 Beweisen Sie Satz 1.6! □

Wir greifen nun die Bemerkung eingangs dieses Abschnitts auf und führen die Substitution von Formeln ein.

Definition 1.9 Es sei $\beta \in \mathcal{A}$ eine Teilformel der Formel $\alpha \in \mathcal{A}$ und $\beta' \in \mathcal{A}$ irgendeine Formel, dann ist $\alpha\,[\,\beta \leftarrowtail \beta'\,]$ die Formel, die entsteht, wenn in α jedes Vorkommen von β durch β' ersetzt wird. □

Beispiel 1.13 Es sei $\alpha = (p \to q) \vee (q \to r)$, $\beta = q \to r$ und $\beta' = \neg q \vee r$. Dann ist $\alpha\,[\,\beta \leftarrowtail \beta'\,] = (p \to q) \vee (\neg q \vee r)$. □

Es ist offensichtlich, dass die Semantik einer Formel α nicht verändert wird, wenn in ihr Teilformeln durch äquivalente Formeln ersetzt werden.

Satz 1.7 Sei $\beta \in \mathcal{A}$ eine Teilformel von α und $\beta' \in \mathcal{A}$ mit $\beta \equiv \beta'$, dann gilt: $\alpha \equiv \alpha\,[\,\beta \leftarrowtail \beta'\,]$. □

Der Satz 1.7 bildet zusammen mit bekannten Äquivalenzen, wie z.B. den in Satz 1.5 aufgelisteten, die Möglichkeit, die Äquivalenz von aussagenlogischen Formeln bzw die Allgemeingültigkeit von aussagenlogischen Formeln rein syntaktisch durch Ersetzen von Teilformeln durch äquivalente nachzuweisen.

Beispiel 1.14 Seien $\alpha, \beta,\gamma \in \mathcal{A}$. Wir zeigen rein syntaktisch, dass die aussagenlogische Formel $(\alpha \to \beta) \vee (\beta \to \gamma)$ allgemeingültig ist:
$(\alpha \to \beta) \vee (\beta \to \gamma)$

$\equiv (\neg\alpha \vee \beta) \vee (\neg\beta \vee \gamma)$	gemäß Definition der Subjunktion
$\equiv \neg\alpha \vee (\beta \vee \neg\beta) \vee \gamma$	wegen der Assoziativität der Disjunktion
$\equiv \neg\alpha \vee 1 \vee \gamma$	wegen der Tautologieregel
$\equiv 1$	wegen der Tautologieregel

1.4.2 Aussagenlogische Basen

Wir kennen bisher folgende fünf zweistellige Verknüpfungen: Disjunktion $\vee$, Konjunktion $\wedge$, Subjunktion $\rightarrow$, Bijunktion $\leftrightarrow$ und Exklusives Oder $\oplus$. Für die weiteren Betrachtungen führen wir für $\alpha,\beta \in \mathcal{A}$ noch zwei weitere Verknüpfungen ein: $\alpha \uparrow \beta$ (*NAND*) und $\alpha \downarrow \beta$ (*NOR*) definiert durch

NAND
NOR

α	β	$\alpha \uparrow \beta$
1	1	0
1	0	1
0	1	1
0	0	1

bzw.

α	β	$\alpha \downarrow \beta$
1	1	0
1	0	0
0	1	0
0	0	1

Folgerung 1.5 Für aussagenlogische Formeln $\alpha,\beta \in \mathcal{A}$ gilt:

a) $\alpha \uparrow \beta \equiv \neg(\alpha \wedge \beta)$.

b) $\alpha \downarrow \beta \equiv \neg(\alpha \vee \beta)$. □

Übungsaufgaben

1.10 Beweisen Sie die beiden Äquivalenzen in Folgerung 1.5! □

In Folgerung 1.1 b) (Seite 18) haben wir bereits überlegt, wie viele zweistellige Verknüpfungen es geben kann. Der folgende Satz verallgemeinert diese Aussage und beantwortet die Frage, wie viele n-stellige Verknüpfungen es prinzipiell geben kann.

Satz 1.8 Es gibt 2^{2^n} n-stellige aussagenlogische Verknüpfungen.

Beweis Eine Wahrheitstafel besitzt für n aussagenlogische Variablen 2^n Zeilen, da jede Variable die Werte 0 oder 1 annehmen kann. Jede dieser 2^n Zeilen kann als Ergebnis (in der letzten Spalte) ebenfalls die Werte 0 oder 1 haben. Das bedeutet, dass es genau 2^{2^n} verschiedene Ergebnisspalten gibt. Also gibt es prinzipiell 2^{2^n} verschiedene n-stellige aussagenlogische Verknüpfungen. □

Folgerung 1.6 Als Spezialfall gilt für $n = 2$, dass es, wie wir bereits in Folgerung 1.1 b) festgestellt haben, $2^{2^2} = 16$ verschiedene zweistellige aussagenlogische Verknüpfungen gibt. □

Wir gehen nun auf die bereits in Folgerung 1.1 b) gestellte Frage ein, ob man tatsächlich 16 verschiedene zweistellige Operationen braucht oder ob man einige Operationen mithilfe anderer äquivalent darstellen kann. Nicht nur theoretisch, sondern auch aus praktischen Gründen ist diese Fragestellung von Interesse. So

ist z.B. im Hinblick auf den Entwurf logischer Schaltungen die Frage nach einer minimalen Anzahl logischer Grundbausteine interessant, mit der alle logischen Schaltungen realisierbar sind.

Beispiel 1.15 **a)** Es seien $\alpha,\beta \in \mathcal{A}$ aussagenlogische Formeln. Aus Folgerung 1.1 a) (Seite 18) kennen wir folgende Äquivalenzen:

$$\begin{aligned}\alpha \to \beta &\equiv \neg\alpha \vee \beta\\ \alpha \leftrightarrow \beta &\equiv (\alpha \to \beta) \wedge (\beta \to \alpha)\\ &\equiv (\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)\\ \alpha \oplus \beta &\equiv (\alpha \wedge \neg\beta) \vee (\neg\alpha \wedge \beta)\end{aligned}$$

Subjunktion, Bijunktion und exklusives Oder sind also durch Negation, Disjunktion und Konjunktion darstellbar.

b) Mithilfe der Doppelten Negation und der De Morganschen Regeln (siehe Satz 1.5, Seite 31) gelten die folgenden Äquivalenzen:

$$\alpha \wedge \beta \equiv \neg\neg(\alpha \wedge \beta) \equiv \neg(\neg\alpha \vee \neg\beta) \tag{1.9}$$
$$\alpha \vee \beta \equiv \neg\neg(\alpha \vee \beta) \equiv \neg(\neg\alpha \wedge \neg\beta) \tag{1.10}$$

Die Konjunktion lässt sich also durch Negation und Disjunktion, die Disjunktion durch Negation und Konjunktion darstellen.

c) Mithilfe von Idempotenz (siehe Satz 1.5) und Folgerung 1.5 b) (Seite 34) gilt

$$\neg\alpha \equiv \neg(\alpha \vee \alpha) \equiv \alpha \downarrow \alpha \tag{1.11}$$

Die Negation lässt sich also durch *NOR* ausdrücken.

d) Mithilfe von Äquivalenz (1.9), Folgerung 1.5 b) und der Äquivalenz (1.11) gilt

$$\alpha \wedge \beta \equiv \neg(\neg\alpha \vee \neg\beta) \equiv \neg\alpha \downarrow \neg\beta \equiv (\alpha \downarrow \alpha) \downarrow (\beta \downarrow \beta)$$

Die Konjunktion lässt sich also alleine durch *NOR* ausdrücken.

Übungsaufgaben

1.11 Zeigen Sie:

(1) Die Negation lässt sich alleine durch *NAND* ausdrücken.

(2) Die Konjunktion lässt sich alleine durch *NAND* ausdrücken.

(3) Die Disjunktion lässt sich alleine durch *NOR* ausdrücken.

(4) Die Disjunktion lässt sich alleine durch *NAND* ausdrücken. □

Es sei $\mathbb{B}^{(4)}$ die Menge aller zweistelligen aussagenlogischen Verknüpfungen und $\mathbb{B}^{(4)^*} = \mathbb{B}^{(4)} \cup \{\neg\}$ diese Menge einschließlich Negation.

Definition 1.10 Sei $\mathcal{O} \subseteq \mathbb{B}^{(4)^*}$ eine Auswahl von aussagenlogischen Verknüpfungen.

a) Eine zweistellige aussagenlogische Verknüpfung $\circ \in \mathbb{B}^{(4)}$ heißt *definierbar* durch $\mathcal{O}$ genau dann, wenn für $\alpha, \beta, \gamma \in \mathcal{A}$ gilt:

(1) Ist $\alpha \circ \beta \equiv \gamma$, dann

(2) sind γ und die Teilformeln α und β allein mit Operatoren aus $\mathcal{O}$ zusammengesetzt.

Die Negation $\neg$ ist *definierbar* durch $\mathcal{O}$ genau dann, wenn für $\alpha \in \mathcal{A}$ gilt: $\neg\alpha \equiv \gamma$ und γ enthält nur α als Teilformel, die allein mit Operatoren aus $\mathcal{O}$ zusammengesetzt ist.

Aussagenlogische Basis

b) Die Menge $\mathcal{O}$ heißt *aussagenlogische Basis*, falls jede Verknüpfung aus $\mathbb{B}^{(4)^*}$ durch $\mathcal{O}$ definierbar ist. □

Auf die oben gestellten Fragen gibt nun der folgende Satz eine Antwort.

Satz 1.9 Die folgenden Mengen aussagenlogischer Verknüpfungen bilden aussagenlogische Basen:[9]

Boolesche Basis	$\{\neg, \vee, \wedge\}$
De Morgan-Basis	$\{\neg, \vee\}$ und $\{\neg, \wedge\}$
Frege-Basis	$\{\neg, \rightarrow\}$
NOR-Basis	$\{\downarrow\}$
NAND-Basis	$\{\uparrow\}$

Übungsaufgaben

1.12 Beweisen Sie Satz 1.9!

9 Die Frege-Basis ist benannt nach dem Mathematiker, Logiker und Philosoph Gottlob Frege (1848 - 1925). Dieser lieferte bedeutende Beträge zur mathematischen Grundlagenforschung und beeinflusste durch seine sprachanalytischen Untersuchungen Philosophie und Linguistik. Frege gilt als Begründer der modernen Logik. In seiner Arbeit „Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens“ stellte er als Erster einen leistungsfähigen Kalkül für die Prädikatenlogik vor.
Die Boolesche Basis ist nach George Boole und die De Morgan-Basis ist nach Augustus de Morgan benannt, zu beiden werden in Kapitel 1.10 bzw. in Kapitel 1.9 noch Angaben gemacht.

1.13 Die dreistellige aussagenlogische Verknüpfung $ifte(\alpha,\beta,\gamma)$ sei durch folgende Wahrheitstabelle definiert:

α	β	γ	$ifte(\alpha,\beta,\gamma)$
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	0

(1) Überlegen Sie, dass diese Verknüpfung die aus Programmiersprachen bekannte Selektionsanweisung

if α **then** β **else** γ **endif**

widerspiegelt!

(2) Verifizieren Sie, dass

$$ifte(\alpha,\beta,\gamma) \equiv (\alpha \rightarrow \beta) \wedge (\neg\alpha \rightarrow \gamma)$$

ist!

(3) Zeigen Sie, dass die Negation sowie alle zweistelligen aussagenlogischen Verknüfungen durch die Verknüpfung $ifte$ definierbar sind, wenn für die Operanden auch Konstanten zugelassen sind! □

1.4.3 Disjunktive und konjunktive Normalform

Sowohl für theoretische Betrachtungen als auch für praktische Anwendungen kann es von Vorteil sein, wenn aussagenlogische Formeln eine einheitliche syntaktische Struktur besitzen. Als Beispiele, die auch in späteren Abschnitten des Buches von Bedeutung sind, führen wir in diesem Abschnitt die disjunktive und die konjunktive Normalform ein.

Disjunktive Normalform

Definition 1.11 **a)** Eine aussagenlogische Formel $\alpha \in \mathcal{A}$ ist in *disjunktiver Normaform* (*DNF*), falls gilt: $\alpha = \alpha_1 \vee \ldots \vee \alpha_n$ mit $\alpha_i = \alpha_{i1} \wedge \ldots \wedge \alpha_{ik_i}$, $1 \leq i \leq n$, wobei alle α_{ij}, $1 \leq j \leq k_i$, $1 \leq i \leq n$, Literale sind.

Konjunktive Normalform

b) Eine aussagenlogische Formel $\alpha \in \mathcal{A}$ ist in *konjunktiver Normaform* (*KNF*), falls gilt $\alpha = \alpha_1 \wedge \ldots \wedge \alpha_n$ mit $\alpha_i = \alpha_{i1} \vee \ldots \vee \alpha_{ik_i}$, $1 \leq i \leq n$, wobei alle α_{ij}, $1 \leq j \leq k_i$, $1 \leq i \leq n$, Literale sind. Die konjunktiv verknüpften Teilformeln α_i, $1 \leq i \leq m$, heißen *Klauseln* von α. □

Klausel

Beispiel 1.16 Die Formel

$$\alpha = (p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q) \vee (p \wedge q \wedge r) \vee (\neg q \wedge \neg r)$$

ist ein Beispiel für eine Formel in disjunktiver Normalform. Die Formel

$$\beta = (\neg p \vee q \vee r) \wedge (\neg p \vee \neg q) \wedge (\neg p \vee q \vee \neg r)$$

ist ein Beispiel für eine Formel in konjunktiver Normalform. □

Eine Formel in α disjunktiver Normalform

$$\alpha = \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{k_i} \alpha_{ij} \right)$$

ist also eine Disjunktion von Konjunktionen von Literalen, eine Formel α in konjunktiver Normalform

$$\alpha = \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{k_i} \alpha_{ij} \right)$$

entsprechend eine Konjunktion von Disjunktionen von Literalen.

Satz 1.10 **a)** Jede aussagenlogische Formel lässt sich in eine äquivalente aussagenlogische Formel in konjunktiver Normalform transformieren.

b) Jede aussagenlogische Formel lässt sich in eine äquivalente aussagenlogische Formel in disjunktiver Normalform transformieren. □

Übungsaufgaben

1.14 Beweisen Sie Satz 1.10! □

Wir zeigen die Behauptung a): Mit folgendem Verfahren können aussagenlogische Formeln $\alpha \in \mathcal{A}$ äquivalent in konjunktive Normalform transformiert werden (dabei nennen wir die Zwischenergebnisse der Transformation nach Schritt (i) α_i):

(1) Ersetze jedes Vorkommen von 1 in α durch $p \vee \neg p$ und jedes Vorkommen von 0 durch $q \wedge \neg q$, wobei p und q zwei neue Variablen sind, d.h. $p, q \notin V_\alpha$.

(2) Ersetze in α_1 jedes Vorkommen einer Teilformel $\neg\neg\beta$ in α_1 durch β bis keine solche Formel mehr vorkommt.

(3) Ersetze in α_2 jedes Vorkommen einer Teilformel $\neg(\beta \wedge \gamma)$ durch $(\neg\beta \vee \neg\gamma)$ und jedes Vorkommen einer Teilformel $\neg(\beta \vee \gamma)$ durch $(\neg\beta \wedge \neg\gamma)$, bis keine solche Formeln mehr vorkommen.

(4) Ersetze in α_3 jedes Vorkommen einer Teilformel $(\beta \vee (\gamma \wedge \delta))$ durch $((\beta \vee \gamma) \wedge (\beta \vee \delta))$ und jedes Vorkommen einer Teilformel $((\beta \wedge \gamma) \vee \delta)$ durch $((\beta \vee \delta) \wedge (\gamma \vee \delta))$, bis keine solche Formeln mehr vorkommen.

Da nur äquivalente Umformungen verwendet werden, ist $\alpha_4 \equiv \alpha$, und α_4 ist in konjunktiver Normalform.

Ein Verfahren zur Transformation einer Formel in disjunktive Normalform lässt sich in analoger Art und Weise angeben. □

Beispiel 1.17 Wir wollen die Formel $\alpha = ((\alpha \rightarrow \beta) \rightarrow \gamma) \vee \delta$ in konjunktive Normalform transformieren. Zunächst wenden wir die Äquivalenz $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$ an und erhalten

$$\alpha \equiv ((\neg\alpha \vee \beta) \rightarrow \gamma) \vee \delta \equiv (\neg(\neg\alpha \vee \beta) \vee \gamma) \vee \delta$$

Mit der Regel (3) erhalten wir

$$\alpha \equiv ((\alpha \wedge \neg\beta) \vee \gamma) \vee \delta$$

und durch zweimaliges Anwenden von (4) erhalten wir die konjunktive Normalform:

$$\begin{aligned} \alpha &\equiv ((\alpha \vee \gamma) \wedge (\neg\beta \vee \gamma)) \vee \delta \\ &\equiv (\alpha \vee \gamma \vee \delta) \wedge (\neg\beta \vee \gamma \vee \delta) \end{aligned}$$

□

Falls die Wahrheitstafel einer aussagenlogischen Formel α mit n Variablen bekannt ist, dann kann man aus dieser eine zu α äquivalente Formel in konjunktiver Normalform konstruieren. Sei dazu $V_\alpha = \{a_1, \ldots, a_n\}$. Wir betrachten nun die Belegungen $\mathcal{I}(a_i)$, $1 \leq i \leq n$, für die $\mathcal{I}^*(\alpha) = 0$ ist, d.h. die Belegungen, die keine Modelle für α sind. Für jede solche Belegung $\mathcal{I}$ bilden wir eine Klausel, die alle Variablen von α enthält und zwar a_i, falls $\mathcal{I}(a_i) = 0$ bzw. $\neg a_i$, falls $\mathcal{I}(a_i) = 1$ ist. Durch das Negieren der Variablen, für die $\mathcal{I}(a_i) = 1$ ist, wird erreicht, dass die Klausel den Wert 0 bekommt, wie die Formel α selbst für diese Belegung. Für alle anderen Belegungen wird die Klausel wahr, also auch für alle Modelle von α. Die so konstruierten Klauseln, auch *Maxterme* genannt, werden konjunktiv verknüpft. Wir nennen die so konstruierte konjunktive Normalform die *kanonische KNF* von α. Es folgt, dass die Formel α und ihre kanonische konjunktive Normalform äquivalent sind.

Maxterm

Kanonische KNF

Beispiel 1.18 Wenden wir dieses Verfahren auf die in Übung 1.13 für die Verknüpfung $ifte$ angegebene Wahrheitstafel an, dann erhalten wir folgende Klauseln

$$\begin{aligned} \neg\alpha \vee \beta \vee \neg\gamma &\quad \text{für die 3. Zeile} \\ \neg\alpha \vee \beta \vee \gamma &\quad \text{für die 4. Zeile} \\ \alpha \vee \neg\beta \vee \gamma &\quad \text{für die 6. Zeile} \\ \alpha \vee \beta \vee \gamma &\quad \text{für die 8. Zeile} \end{aligned}$$

und damit die zu $ifte$ äquivalente kanonische konjunktive Normalform:

$$ifte(\alpha,\beta,\gamma) \equiv (\neg\alpha \vee \beta \vee \neg\gamma) \wedge (\neg\alpha \vee \beta \vee \gamma) \wedge (\alpha \vee \neg\beta \vee \gamma) \wedge (\alpha \vee \beta \vee \gamma)$$

Durch Ausklammern erhält man

$$ifte(\alpha,\beta,\gamma) \equiv (\neg\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

und damit eine weitere zu $ifte$ äquivalente Formel in konjunktiver Normalform, die man im Übrigen auch erhält, wenn man auf die Darstellung von $ifte$ in Übung 1.13 (2) die beiden Äquivalenzen $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$ und $\neg\alpha \rightarrow \gamma \equiv \alpha \vee \gamma$ anwendet. □

Bemerkung 1.2 Aus dem Beispiel folgt unmittelbar, dass Formeln verschiedene, selbstverständlich alle untereinander äquivalente Darstellungen in konjunktiver Normalform haben können. Im Gegensatz dazu ist die kanonische KNF eindeutig, abgesehen von kommutativer Vertauschung der Maxterme und kommutativer Vertauschung der Literale innerhalb der Maxterme. □

Kanonische DNF

In analoger Art und Weise kann man aus der Wahrheitstafel einer Formel α mit n Variablen $a_1, \ldots, a_n$ eine diskuntktive Normalform, entsprechend *kanonische disjunktive Normalform* genannt, konstruieren. Wir betrachten jetzt die Belegungen $\mathcal{I}(a_i)$, $1 \leq i \leq n$, für die $\mathcal{I}^*(\alpha) = 1$ ist, d.h. die Belegungen, die Modelle für α sind. Für jede solche Belegung $\mathcal{I}$ bilden wir eine Konjunktion, die alle Variablen von α enthält und zwar a_i, falls $\mathcal{I}(a_i) = 1$ bzw. $\neg a_i$, falls $\mathcal{I}(a_i) = 0$ ist. Durch das Negieren der Variablen, für die $\mathcal{I}(a_i) = 0$ ist, wird erreicht, dass die Klausel den Wert 1 bekommt, wie die Formel α selbst für diese Belegung. Für alle anderen Belegungen wird die Konjunktion falsch, also auch für alle Belegungen von α, die keine Modelle sind. Die so konstruierten Konjunktionen, auch *Minterme* genannt, werden dann disjunktiv zur kanonischen DNF von α verknüpft; α und ihre kanonische DNF sind äquivalent.

Minterm

Beispiel 1.19 Wenden wir dieses Verfahren ebenfalls auf die in Übung 1.13 für die Verknüpfung $ifte$ angegebene Wahrheitstafel an, dann erhalten wir folgende Konjunktionen

$$\begin{aligned} \alpha \wedge \beta \wedge \gamma &\quad \text{für die 1. Zeile} \\ \alpha \wedge \beta \wedge \neg\gamma &\quad \text{für die 2. Zeile} \\ \neg\alpha \wedge \beta \wedge \gamma &\quad \text{für die 5. Zeile} \\ \neg\alpha \wedge \neg\beta \wedge \gamma &\quad \text{für die 7. Zeile} \end{aligned}$$

und damit die zu $ifte$ äquivalente kanonische disjunktive Normalform:

$$ifte(\alpha,\beta,\gamma) \equiv (\alpha \wedge \beta \wedge \gamma) \vee (\alpha \wedge \beta \wedge \neg\gamma) \vee (\neg\alpha \wedge \beta \wedge \gamma) \vee (\neg\alpha \wedge \neg\beta \wedge \gamma)$$

□

Satz 1.11 Ersetzt man in einer aussagenlogischen Formel $\alpha \in \mathcal{A}$, in der nur die Operatoren $\neg$, $\vee$ und $\wedge$ vorkommen, jedes Vorkommen von $\wedge$ durch $\vee$, jedes Vorkommen von $\vee$ durch $\wedge$, jedes Vorkommen von 0 durch 1, jedes Vorkommen von 1 durch 0 sowie jedes Vorkommen eines Literals durch seine Negation, dann gilt für die so entstehende Formel β: $\alpha \equiv \neg\beta$. □

Dualitätsprinzip der Aussagenlogik

Übungsaufgaben

1.15 Beweisen Sie Satz 1.11! □

Folgerung 1.7 Sei $\alpha \in \mathcal{A}$ eine aussagenlogische Formel. Sei *dnf*(α) die kanonische disjunktive und *knf*(α) die kanonische konjunktive Normalform zu α, dann gilt

$$\textit{dnf}(\alpha) \equiv \neg\textit{knf}(\neg\alpha) \text{ sowie } \textit{knf}(\alpha) \equiv \neg\textit{dnf}(\neg\alpha)$$

Übungsaufgaben

1.16 a) Verifizieren Sie Folgerung 1.7 anhand der Beispiele 1.18 und 1.19!

b) Beweisen Sie Folgerung 1.7! □

Wir wollen noch eine bestimmte Art der konjunktiven Normalform betrachten, nämlich solche in denen alle Klauseln genau drei Literale enthalten.

Definition 1.12 Eine Formel $\alpha \in \mathcal{A}$ in KNF ist in *3KNF* genau dann, wenn alle Klauseln von α genau drei Literale enthalten. □

3KNF

Wir geben ein Verfahren an, mit dem eine Formel in KNF eine Formel in 3KNF transformiert werden kann: Sei $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_n$ mit $\alpha_i = (\, \alpha_{i1} \vee \ldots \vee \alpha_{ik_i} \,)$, $1 \leq i \leq n$, gegeben. Für jede Klausel α_i betrachten wir folgende vier Fälle:

(1) Ist $k_i = 1$, d.h. $\alpha_i = (\, \alpha_{i1} \,)$ besteht aus genau einem Literal, dann benötigen wir zwei neue Variablen p_i und q, und wir ersetzen α_i durch

$$\begin{aligned}\beta_i = {} & (\, \alpha_{i1} \vee p_i \vee q_i \,) \wedge (\, \alpha_{i1} \vee \neg p_i \vee q_i \,) \\ & \wedge (\, \alpha_{i1} \vee p_i \vee \neg q_i \,) \wedge (\, \alpha_{i1} \vee \neg p_i \vee \neg q_i \,)\end{aligned}$$

(2) Ist $k_i = 2$, d.h. $\alpha_i = (\, \alpha_{i1} \vee \alpha_{i2} \,)$ besteht aus zwei Literalen, dann benötigen wir eine neue Variable p_i und ersetzen α_i durch

$$\beta_i = (\, \alpha_{i1} \vee \alpha_{i2} \vee p_i \,) \wedge (\, \alpha_{i1} \vee \alpha_{i2} \vee \neg p_i \,)$$

(3) Ist $k_i = 3$, dann ist nichts zu tun: $\beta_i = \alpha_i$.

(4) Ist $k_i > 3$, dann benötigen wir $k_i - 3$ neue Variable $p_{i,1}, \ldots, p_{i,k_i-3}$ und ersetzen α_i durch

$$\beta_i = (\, \alpha_{i1} \vee \alpha_{i2} \vee p_{i,1} \,) \wedge (\, \neg p_{i,1} \vee \alpha_{i3} \vee p_{i,2} \,) \wedge \\ \ldots \wedge (\, \neg p_{i,k_i-3} \vee \alpha_{i,k_i-1} \vee \alpha_{i,k_i} \,)$$

Beispiel 1.20 Wir betrachten die KNF-Formel

$$\alpha = p \wedge (\, p \vee q \,) \wedge (\, p \vee q \vee r \,) \wedge (\, p \vee q \vee r \vee s \vee t \,)$$

Fall (1) trifft auf die Klausel p zu. Wir führen die Variablen a und b und ersetzen die Klausel durch

$$(\, p \vee a \vee b \,) \wedge (\, p \vee \neg a \vee b \,) \wedge (\, p \vee a \vee \neg b \,) \wedge (\, p \vee \neg a \vee \neg b \,)$$

Auf die Klausel $p \vee q$ trifft Fall (2) zu. Wir führen die Variable c ein und ersetzen die Klausel durch

$$(\, p \vee q \vee c \,) \wedge (\, p \vee q \vee \neg c \,)$$

Die Klausel $p \vee q \vee r$ braucht nicht ersetzt zu werden. Auf die Klausel $(\, p \vee q \vee r \vee s \vee t \,)$ trifft Fall (4) zu, die Anzahl der Literale ist 5. Wir führen deshalb 2 neue Variable d und e ein und ersetzen die Klausel durch

$$(\, p \vee q \vee d \,) \wedge (\, \neg d \vee r \vee e \,) \wedge (\, \neg e \vee s \vee t \,)$$

Wir erhalten also insgesamt die Formel

$$\begin{aligned} \alpha \equiv\ & (\, p \vee a \vee b \,) \wedge (\, p \vee \neg a \vee b \,) \wedge (\, p \vee a \vee \neg b \,) \wedge (\, p \vee \neg a \vee \neg b \,) \\ & \wedge (\, p \vee q \vee c \,) \wedge (\, p \vee q \vee \neg c \,) \\ & \wedge (\, p \vee q \vee r \,) \\ & \wedge (\, p \vee q \vee d \,) \wedge (\, \neg d \vee r \vee e \,) \wedge (\, \neg e \vee s \vee t \,) \end{aligned}$$

als Ergebnis der Transformation der Formel α in 3KNF. □

Satz 1.12 Sei $\alpha \in \mathcal{A}$ in KNF und β die Formel in 3KNF, die aus α durch Anwenden des obigen Verfahrens resultiert. Dann ist α erfüllbar genau dann, wenn β erfüllbar ist.

Beweis Wir betrachten die Fälle (1), (2) und (4), zu (3) ist nichts zu zeigen. Zu (1): Ist $\alpha_i = (\, \alpha_{i1} \,)$, dann ist

$$\begin{aligned} \beta_i = & (\, \alpha_{i1} \vee p_i \vee q_i \,) \wedge (\, \alpha_{i1} \vee \neg p_i \vee q_i \,) \\ & \wedge (\, \alpha_{i1} \vee p_i \vee \neg q_i \,) \wedge (\, \alpha_{i1} \vee \neg p_i \vee \neg q_i \,) \end{aligned}$$

Die Belegung $\mathcal{I}(\alpha_{i1}) = 1$ erfüllt α_i und β_i. Die Belegung $\mathcal{I}(\alpha_{i1}) = 0$ erfüllt α_i nicht, und für diese Belegung ist genau eine Klausel von β_i nicht erfüllt für jede Wahl $\mathcal{I}(p_i), \mathcal{I}(q_i) \in \{0, 1\}$, d.h. β_i ist für $\mathcal{I}(\alpha_{i1}) = 0$ nicht erfüllbar.

Zu (2): Es gilt mit bekannten Äquivalenzen

$$\begin{aligned}\alpha &= (\,\alpha_{i1} \vee \alpha_{i2}\,)\\ &\equiv (\,\alpha_{i1} \vee \alpha_{i2}\,) \vee 0\\ &\equiv (\,\alpha_{i1} \vee \alpha_{i2}\,) \vee (\,p_i \wedge \neg p_i\,)\\ &\equiv (\,\alpha_{i1} \vee \alpha_{i2} \vee p_i\,) \wedge (\,\alpha_{i1} \vee \alpha_{i2} \vee \neg p_i\,)\\ &= \beta\end{aligned}$$

und damit $\alpha_i \equiv \beta_i$, d.h. α_i ist erfüllbar genau dann, wenn β_i erfüllbar ist.

Zu (4): Sei α_i erfüllbar, dann gilt $\mathcal{I}(\alpha_{ij}) = 1$ für mindestens ein j, $1 \leq j \leq k_i$. Wir betrachten zwei Fälle:

1. Falls $\mathcal{I}(\alpha_{i1}) = 1$ oder $\mathcal{I}(\alpha_{i2}) = 1$ ist, dann ist β_i erfüllbar für $\mathcal{I}(p_{ij}) = 0$, $1 \leq j \leq k_i - 3$.

2. Falls $\mathcal{I}(\alpha_{ij}) = 1$ für ein j mit $3 \leq j \leq k_i$ ist, so ist β_i ebenfalls erfüllbar und zwar für $\mathcal{I}(p_{ir}) = 1$, $1 \leq r \leq j - 2$, und $\mathcal{I}(p_{ir}) = 0$, $j - 1 \leq r \leq k_i - 3$.

β_i ist also in allen Füllen erfüllbar, in denen α_i erfüllbar ist.

Sei nun β_i erfüllbar, auch hierfür unterscheiden wir zwei Fälle:

1. Ist $\mathcal{I}(p_{i1}) = 0$, so muss $\mathcal{I}(\alpha_{i1}) = 1$ oder $\mathcal{I}(\alpha_{i2}) = 1$ gewählt werden, damit β_i erfüllt ist.

2. $\mathcal{I}(p_{i1}) = 1$, so muss $\mathcal{I}(\alpha_{ij}) = 1$ für mindestens ein j, $3 \leq j \leq k_i$, gewählt werden, damit β_i erfüllt ist.

Insgesamt gilt also, dass für mindenstens ein j, $1 \leq j \leq k_i$, $\mathcal{I}(\alpha_{ij}) = 1$ ist, was ausreicht, damit $\mathcal{I}^*(\alpha_i) = 1$ wird. Das heißt, dass jede erfüllende Belegung von β_i eine erfüllende Belegung für α_i. □

1.4.4 Zusammenfassung

Zwei aussagenlogische Formeln heißen äquivalent, wenn ihre Interpretationen für jede ihrer Belegungen übereinstimmen. Die Semantik einer aussagenlogischen Formel ist invariant gegenüber der Substitution von Teilformeln durch äquivalente Formeln. Äquivalenzen können verwendet werden, um syntaktisch die Äquivalenz, die Allgemeingültigkeit oder die Unerfüllbarkeit von Formeln nachzuweisen.

Eine aussagenlogische Basis ist eine Menge von aussagenlogischen Verknüpfungen, mit denen alle aussagenlogische Verknüpfungen dargestellt werden können. Bekannte aussagenlogischen Basen sind die Boolesche Basis, mit der wir die

Sprache der Aussagenlogik eingeführt haben, die de Morgan-Basen, die Frege-Basis sowie die *NOR*- und die *NAND*-Basis; die beiden letztgenannten Basen enthalten nur ein Element: die *NOR*- bzw. die *NAND*-Verknüpfung.

Jede aussagenlogische Formel lässt sich in eine äquivalente Formel in disjunktiver und in eine äquivalente Formel in konjunktiver Normalform transformieren. Aus der Wahrheitstafel einer Formel lassen sich ihre kanonische disjunktive und ihre kanonische konjunktive Normalform bestimmen. Diese sind jeweils bis auf Vertauschungen von Literalen innerhalb von Mintermen bzw. innerhalb von Maxtermen und bis auf Vertauschung der Minterme bzw. Vertauschung der Maxterme untereinander eindeutig. Formeln in konjunktiver Normalform lassen sich erfüllbarkeitsäquivalent in 3KNF transformieren.

Das Dualitätsprinzip der Aussagenlogik besagt, dass (auf der Basis der de Morgan-Regeln) disjunktive und konjunktive Normalformen schematisch ineinander transformiert werden können.

1.5 Resolutionskalkül

Im Abschnitt 1.3.3 haben wir grundlegende Begriffe für Kalküle und wesentliche Eigenschaften von Kalkülen kennen gelernt. Wir wollen nun kurz auf den Resolutionskalkül eingehen, der in der Logischen Programmierung und in der Künstlichen Intelligenz eine wichtige Rolle spielt. Der Resolutionskalkül ist ein widerspruchsfreier und vollständiger Kalkül, mit dem die (Un-) Erfüllbarkeit einer Menge von Klauseln nachgewiesen werden kann.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- den Resolutionskalkül erklären und anwenden können.

1.5.1 Klauselmengen

Klausel

Definition 1.13 Sei

$$\alpha = (p_{11} \vee \ldots \vee p_{1k_1}) \wedge \ldots \wedge (p_{n1} \vee \ldots \vee p_{nk_n})$$

die in konjunktiver Normalform gegebene aussagenlogische Formel $\alpha \in \mathcal{A}$. Dann heißen die Mengen $\{p_{i1}, \ldots, p_{ik_i}\}$, $1 \leq i \leq n$, der jeweils disjunktiv verknüpften Literale die *Klauseln* von α, und die Menge ihrer Klauseln

Klauselmenge

$$M_\alpha = \{\, \{\, p_{11}, \ldots, p_{1k_1} \,\}, \ldots, \{\, p_{n1}, \ldots, p_{nk_n} \,\} \,\}$$

heißt *Klauselmenge* von α. Klauseln, die eine Variable und ihre Negation enthalten, heißen *trivial*.

Triviale Klauseln

Um leere Klauseln von leeren Klauselmengen zu unterscheiden, notieren wir erstere mit dem Symbol $\diamond$ und letztere wie üblich mit dem Symbol $\emptyset$ für leere Mengen. □

Beispiel 1.21 **a)** Für die Formel β aus Beispiel 1.16 (Seite 37) gilt:

$$M_\beta = \{ \{ \neg p, q, r \}, \{ \neg p, \neg q \}, \{ \neg p, q, \neg r \} \}$$

□

Übungsaufgaben

1.17 Geben Sie die Klauselmenge der Formel

$$\alpha = (p \vee \neg q) \wedge (p \vee \neg r \vee q \vee \neg r \vee q) \wedge (p \vee \neg q \vee r) \wedge (\neg q \vee p)$$

an. □

Satz 1.13 Seien α_1 und α_2 zwei äquivalente aussagenlogische *KNF*-Formeln mit $V_{\alpha_1} = V_{\alpha_2}$. Dann gilt $M_{\alpha_1} = M_{\alpha_2}$. □

Beispiel 1.22 Die folgenden drei Formeln sind äquivalent:

$$\alpha_1 = (p \vee p) \wedge (q \vee \neg r), \quad \alpha_2 = (\neg r \vee q) \wedge p, \alpha_3 = p \wedge (q \vee \neg r \vee q)$$

Sie besitzen dieselben Variablenmengen, und es gilt $M_{\alpha_1} = M_{\alpha_2} = M_{\alpha_3} = \{ \{p\}, \{ q, \neg r \} \}$. □

Eine Klauselmenge abstrahiert von äquivalenten Umformungen mit Kommutativitäts-, Assoziativitäts- und Distributivitätsregeln.

Da es sich bei der Klauselmenge einer Formel also nur um eine andere Darstellung dieser Formel handelt, können die Begriffe Belegung, Interpretation, Ableitungsregel, (Un-) Erfüllbarkeit, Folgerung und Äquivalenz auf Klauselmengen entsprechend übertragen werden.

Folgerung 1.8 **a)** Die leere Klausel $\diamond$ ist unerfüllbar.

b) Die leere Klauselmenge $\emptyset$ ist allgemeingültig.

c) Sei M eine Klauselmenge und K eine triviale Klausel mit $K \in M$, dann ist $M \equiv M - \{K\}$.

Beweis **a)** Eine Klausel ist erfüllbar, wenn es eine Belegung der Variablen dieser Klausel gibt, die mindestens ein Literal wahr macht. Da die leere Klausel keine Literale enthält, kann es auch keine Belegung für ihre Variablen geben, die ein Literal wahr macht.

b) Eine Klauselmenge ist allgemeingültig, wenn jede Belegung der Variablen der Klauselmenge jede Klausel wahr macht. Da die leere Klauselmenge $\emptyset$ keine Klauseln enthält, müssen auch keine Klauseln wahr gemacht werden.

c) K enthalte die Literale p und $\neg p$. Dann kann die Disjunktion, die K repräsentiert äquivalent in die Formel $\alpha = \beta \vee p \vee \neg p$ umgeformt werden, wobei β aus einer Disjunktion aller anderen Literale von K besteht. Offensichtlich ist α allgemeingültig und deshalb irrelevant bei der Berechnung aller Interpretationen von M, da M aus einer Konjunktion von Klauseln besteht. □

Bemerkung 1.3 Wegen Folgerung 1.8 c) können wir im Folgenden davon ausgehen, dass KNF-Formeln und Klauselmengen keine trivialen Klauseln enthalten. □

1.5.2 Der Resolutionsoperator

In Vorbereitung auf die folgenden Überlegungen betrachten wir zunächst ein Beispiel.

Beispiel 1.23 Es sei

$$\alpha = (p \vee q \vee \neg r) \wedge (r \vee \neg s)$$

Es ist also

$$M_\alpha = \{ \{ p, q, \neg r \}, \{ r, \neg s \} \}$$

Ist α bzw. M_α erfüllbar? M_α ist erfüllbar genau dann, wenn beide Klauseln $K_1 = \{ p, q, \neg r \}$ und $K_2 = \{ r, \neg s \}$ erfüllbar sind. Beide Klauseln sind erfüllbar genau dann, wenn $\mathcal{I}(p) = 1$ oder $\mathcal{I}(q) = 1$ und in jedem Fall $\mathcal{I}(s) = 0$ gesetzt wird. Die Belegung des Literals r ist für die Erfüllbarkeit von K_1 und K_2 unerheblich, da r in K_1 und die Negation $\neg r$ in K_2 auftritt. r und $\neg r$ „neutralisieren“ sich quasi hinsichtlich der Erfüllbarkeit von M_α. Aus unserer Überlegung folgt, dass M_α genau dann erfüllbar ist, wenn die Klauselmenge

$$M'_\alpha = \{ \{ p, q, \neg r \}, \{ r, \neg s \}, \{ p, q, \neg s \} \}$$

erfüllbar ist, es ist also $M_\alpha \equiv M'_\alpha$. □

Diese beispielhafte Überlegung ist die Grundlage für folgende Ableitungsregel, auf welcher der Resolutionskalkül basiert.

Resolution

Definition 1.14 Die *Resolution* erfolgt mithilfe der Ableitungregel

$$\frac{p_1 \vee \ldots \vee p_m \vee r,\ q_1 \vee \ldots \vee q_n \vee \neg r}{p_1 \vee \ldots \vee p_m \vee q_1 \vee \ldots \vee q_n}$$

oder in „Klauselnotation“

$$\frac{\{ p_1, \ldots, p_m, r \},\ \{ q_1, \ldots, q_n, \neg r \}}{\{ p_1, \ldots, p_m, q_1, \ldots, q_n \}}$$

Aus schreibtechnischen Gründen führen wir eine neue Schreibweise ein: Für die Klauselmengen

$$K_1 = \{ p_1, \ldots, p_m, r \} \text{ und } K_2 = \{ q_1, \ldots, q_n, \neg r \}$$

ist

Resolvente

$$K = \{ p_1, \ldots, p_m, q_1, \ldots, q_n \}$$

eine *Resolvente* von K_1 und K_2. Schreibweise: $K = Res(K_1, K_2)$.

r und $\neg r$ sind die für diese Resolution *passenden Literale*. □

Passende Literale

Beispiel 1.24 **a)** Es seien $K_1 = \{ p, q, \neg r \}$ und $K_2 = \{ r, \neg s \}$ die beiden Klauseln aus Beispiel 1.23, dann ist $Res(K_1, K_2) = \{ p, q, \neg s \}$ die einzige Resolvente von K_1 und K_2.

b) Es seien $K_1 = \{ p, \neg q, r \}$ und $K_2 = \{ q, \neg r \}$ Klauseln, dann sind $Res(K_1, K_2) = \{ p, r, \neg r \}$ sowie $Res(K_2, K_1) = \{ p, q, \neg q \}$ mögliche Resolventen von K_1 und K_2.

c) Die Resolvente der Klauseln $K_1 = \{p\}$ und $K_2 = \{\neg p\}$ ist leer: $Res(K_1, K_2) = \diamond$. □

Bemerkung 1.4 Zu beachten ist, dass die Definition 1.14 immer nur das Resolvieren genau eines Paares von passenden Literalen zulässt. Zwei Klauseln mit mehr als einem Paar passender Klauseln können also mehrere verschiedene Resolventen haben (siehe Beispiel 1.24 b). Diese sind allerdings trivial und könnten gemäß Folgerung 1.8 c) aus einer Klauselmenge entfernt werden. □

Satz 1.14 Sei $\alpha \in \mathcal{A}$ in konjunktiver Normalform, $K_1, K_2 \in M_\alpha$ seien Klauseln von α und $K = Res(K_1, K_2)$ eine Resolvente von K_1 und K_2. Dann gilt $M_\alpha \equiv M_\alpha \cup \{K\}$. Dabei ist $M_\alpha \cup \{K\}$ die Menge von Klauseln bestehend aus allen Klauseln von M_α und der Klausel K.

Resolutionslemma

Beweis Es sei $K_1 = \{ p_1, \ldots, p_m, r \}$ und $K_2 = \{ q_1, \ldots, q_n, \neg r \}$ sowie $K = Res(K_1, K_2) = \{ p_1, \ldots, p_m, q_1, \ldots, q_n \}$. Es ist offensichtlich, dass dann, wenn $\mathcal{I}$ ein Modell für $M_\alpha \cup \{K\}$ ist, $\mathcal{I}$ auch ein Modell für M_α ist. Ist umgekehrt $\mathcal{I}$ ein Modell für M_α, dann ist $\mathcal{I}$ auch ein Modell für jede Klausel in M_α, also auch für die Klauseln K_1 und K_2. Wir betrachten zwei Fälle: (1) $\mathcal{I}(r) = 1$ sowie (2) $\mathcal{I}(r) = 0$.

Zu (1): Dann ist $\mathcal{I}(\neg r) = 0$ und da $\mathcal{I}$ ein Modell für K_2 ist, muss $\mathcal{I}$ ein Modell für $\{ q_1, \ldots, q_n \}$ sein, und damit ist $\mathcal{I}$ ein Modell für $K = \{ p_1, \ldots, p_m, q_1, \ldots, q_n \}$.

Zu (2): Dann muss, da $\mathcal{I}$ ein Modell für K_1 ist, $\mathcal{I}$ auch ein Modell für $\{ p_1, \ldots, p_m \}$ sein, und damit ist $\mathcal{I}$ ein Modell für $K = \{ p_1, \ldots, p_m, q_1, \ldots, q_n \}$.

In jedem Fall ist also ein Modell $\mathcal{I}$ von M_α auch ein Modell für K und damit für $M_\alpha \cup \{ K \}$. □

Wir definieren nun die fortgesetzte Anwendung des Resolutionsoperators Res auf eine Klauselmenge.

Definition 1.15 Sei M_α die Klauselmenge einer aussagenlogischen Formel $\alpha \in \mathcal{A}$ in konjunktiver Normalform. Dann sei

$$Res(M_\alpha) = M_\alpha \cup \{\, Res(K_1, K_2) \mid K_1, K_2 \in M_\alpha \,\}$$

Dabei bedeutet $M_\alpha \cup \{\, Res(K_1, K_2) \mid K_1, K_2 \in M_\alpha \,\}$, dass zur Klauselmenge M_α alle möglichen Resolventen von allen möglichen Paaren von Klauseln aus M_α hinzugefügt werden. Wir wenden nun den Operator Res wiederholt auf M_α an:

$$\begin{aligned} Res^0(M_\alpha) &= M_\alpha \\ Res^{n+1}(M_\alpha) &= Res(Res^n(M_\alpha)),\ n \geq 0 \end{aligned}$$

$Res(M_\alpha)$ bedeutet also die Anwendung des Operators Res auf alle Paare von Klauseln aus M_α, und für $k \in \mathbb{N}_0$ bedeutet $Res^k(M_\alpha)$, dass der Res-Operator k-mal angewendet wird, zunächst auf M_α, dann auf das Ergebnis dieser Anwendung, dann auf dessen Ergebnis usw.:

$$Res^k(M_\alpha) = \underbrace{Res(Res(\ldots Res}_{k\text{-mal}}(M_\alpha)\ldots))$$

Aus dem Resolutionslemma (Satz 1.14, Seite 47) folgt unmittelbar

Folgerung 1.9 Sei M_α die Klauselmenge einer aussagenlogischen Formel $\alpha \in \mathcal{A}$ in konjunktiver Normalform. Dann gilt

a) $M_\alpha \equiv Res^i(M_\alpha)$ für alle $i \geq 0$,

b) $Res^i(M_\alpha) \equiv Res^j(M_\alpha)$ für alle $i, j \geq 0$. □

Beispiel 1.25 Wir betrachten die Formel

$$\alpha = (\neg r \vee p \vee q) \wedge (p \vee q \vee r) \wedge (\neg q \vee p)$$

Es ist also $M_\alpha = \{\{p, q, \neg r\}, \{p, q, r\}, \{p, \neg q\}\}$. Es gilt:

$$\begin{aligned} Res(M_\alpha) &= \{\{p, q, \neg r\}, \{p, q, r\}, \{p, \neg q\}, \{p, q\}, \{p, \neg r\}, \{p, r\}\} \\ Res^2(M_\alpha) &= Res(Res(M_\alpha)) \\ &= \{\{p, q, \neg r\}, \{p, q, r\}, \{p, \neg q\}, \{p, q\}, \{p, \neg r\}, \{p, r\}, \{p\}\} \\ Res^3(M_\alpha) &= Res(Res^2(M_\alpha)) \\ &= \{\{p, q, \neg r\}, \{p, q, r\}, \{p, \neg q\}, \{p, q\}, \{p, \neg r\}, \{p, r\}, \{p\}\} \end{aligned}$$

Es ist also $Res^3(M_\alpha) = Res^2(M_\alpha)$ und damit $Res^l(M_\alpha) = Res^2(M_\alpha)$ für alle $l \geq 2$. Nach zweimaligem Anwenden des Operators wird die Klauselmenge stationär, d.h. verändert sich nicht mehr. □

Dass die fortgesetzte Anwendung des Operators Res auf eine Klauselmenge nach endlich vielen Schritten stationär wird, d.h. keine neuen Klauseln mehr produziert, ist einsichtig, denn bei jeder Anwendung von Res kommen höchstens Klauseln hinzu, die ein Literal weniger enthalten. Da es nur endlich viele Klauseln und in jeder Klausel nur endlich viele Literale gibt, muss dieser Prozess stoppen. Es gilt der folgende Satz.

Satz 1.15 Sei M_α Klauselmenge einer aussagenlogischen Formel $\alpha \in \mathcal{A}$ in konjunktiver Normalform. Dann gibt es ein $t \in \mathbb{N}_0$, so dass $Res^t(M_\alpha) = Res^l(M_\alpha)$ ist für alle $l \geq t$. □

Die Klauselmenge $Res^t(M_\alpha)$, für die $Res^t(M_\alpha) = Res^l(M_\alpha)$ für alle $l \geq t$ gilt, bezeichnen wir mit $Res^*(M_\alpha)$.

Beispiel 1.26 Im obigen Beispiel 1.25 gilt: $Res^*(M_\alpha) = Res^2(M_\alpha)$. □

Aus dem obigen Satz 1.15 und dem Resolutionslemma, Satz 1.14 (Seite 47), folgt unmittelbar

Folgerung 1.10 Sei M_α Klauselmenge einer aussagenlogischen Formel $\alpha \in \mathcal{A}$ in konjunktiver Normalform, dann ist

a) $M_\alpha \equiv Res^*(M_\alpha)$,

b) M_α (un-) erfüllbar genau dann, wenn $Res^*(M_\alpha)$ (un-) erfüllbar ist. □

Als Vorbereitung auf den nächsten Satz betrachten wir folgendes Beispiel.

Beispiel 1.27 Für die Formel

$$\alpha = (p \vee q \vee \neg r) \wedge \neg p \wedge (p \vee q \vee r) \wedge (p \vee \neg q)$$

mit der Klauselmenge

$$M_\alpha = \{\{p, q, \neg r\}, \{\neg p\}, \{p, q, r\}, \{p, \neg q\}\}$$

gilt:

$$\begin{aligned} Res(M_\alpha) = \{&\{p, q, \neg r\}, \{\neg p\}, \{p, q, r\}, \{p, \neg q\}, \\ &\{q, \neg r\}, \{p, q\}, \{p, \neg r\}, \{q, r\}, \{\neg q\}, \{p, r\}\} \\ Res^2(M_\alpha) = Res(Res(M_\alpha)) \\ = \{&\{p, q, \neg r\}, \{\neg p\}, \{p, q, r\}, \{p, \neg q\}, \\ &\{q, \neg r\}, \{p, q\}, \{p, \neg r\}, \{q, r\}, \{\neg q\}, \{p, r\} \\ &\{q\}, \{\neg r\}, \{r\}, \{p\}\} \end{aligned}$$

Diese Klauselmenge enthält die beiden Klauseln r und $\neg r$ – im Übrigen auch noch p und $\neg p$ sowie q und $\neg q$ –, d.h. die entsprechende Formel hat die Gestalt

$$\alpha' = \ldots \wedge \neg p \wedge \ldots \wedge \neg q \wedge \ldots \wedge q \wedge \neg r \wedge r \wedge p$$

Alleine wegen der Teilformel $r \wedge \neg r$ und ebenso wegen der Teilformeln $p \wedge \neg p$ und $\neg q \wedge q$ ist die Formel α' und damit die Klauselmenge $Res^2(M_\alpha)$ unerfüllbar, denn r und $\neg r$ sind widersprüchlich (siehe Satz 1.5, Seite 31). Werden r und $\neg r$ resolviert, dann entsteht die leere Klausel: $Res(\{r\}, \{\neg r\}) = \diamond$ (gleiches gilt natürlich auch für p und $\neg p$ sowie für q und $\neg q$). Es folgt $\diamond \in Res^3(M_\alpha)$. □

Aus dem Beispiel können wir ableiten, dass $Res^k(M_\alpha)$ für ein $k \geq 1$ erstmalig die leere Klausel enthält, falls in $Res^{k-1}(M_\alpha)$ eine Klausel genau aus einem

Literal besteht und eine weitere Klausel genau aus dessen Negation. Damit ist $Res^{k-1}(M_\alpha)$ eine unerfüllbare Klauselmenge. Wegen Folgerung 1.10 b) (Seite 49) ist damit auch M_α und damit α unerfüllbar.

Im folgenden Satz fassen wir dieses Ergebnis zusammen, und wir zeigen im Beweis des Satzes, dass der Resolutionskalkül hinsichtlich des Nachweises der Unerfüllbarkeit einer aussagenlogischen Formel korrekt und vollständig ist (siehe Abschnitt 1.3.3).

Resolutionssatz der Aussagenlogik

Satz 1.16 Sei M_α die Klauselmenge der aussagenlogischen Formel $\alpha \in \mathcal{A}$ in konjunktiver Normalform. Dann gilt: M_α (und damit α) ist unerfüllbar genau dann, wenn $\diamond \in Res^*(M_\alpha)$ ist.

Beweis Wir zeigen zuerst die Korrektheit, d.h., ist $\diamond \in Res^*(M_\alpha)$, dann ist M_α unerfüllbar, womit gezeigt ist, dass der Resolutionskalkül keine erfüllbare Formel als unerfüllbar ermittelt. Sei also $\diamond \in Res^*(M_\alpha)$. Wie wir oben im Anschluss an Beispiel 1.27 überlegt haben, kann die leere Klausel nur durch Resolution zweier Klauseln $K_1 = \{ l \}$ und $K_2 = \{ \neg l \}$, wobei l ein Literal ist, entstehen. Gemäß Folgerung 1.9 auf Seite 48 gilt $M_\alpha \equiv Res^i(M_\alpha) \equiv Res^j(M_\alpha)$ für alle $i, j \geq 0$. Ist also $\diamond \in Res^*(M_\alpha)$, dann muss es ein $k \geq 1$ geben mit $\diamond \in Res^k(M_\alpha)$ und $K_1, K_2 \in Res^{k-1}(M_\alpha)$. Da es keine Belegung gibt, die sowohl K_1 als auch K_2 erfüllt, ist $Res^{k-1}(M_\alpha)$ unerfüllbar und damit, da $Res^k(M_\alpha) \equiv M_\alpha$ ist, ist auch M_α unerfüllbar.

Wir zeigen nun die Vollständigkeit, d.h., ist α unerfüllbar, dann ist $\diamond \in Res^*(M_\alpha)$, womit gezeigt ist, dass alle unerfüllbaren Formeln durch den Resolutionskalkül auch als solche erkannt werden. Wir zeigen die Behauptung mithilfe einer so genannten vollständigen Induktion über die Anzahl n der atomaren Formeln in einer Klauselmenge. Vollständige Induktion bedeutet, dass man die Behauptung zunächst für einen Anfangswert für n zeigt, dann annimmt, dass die Behauptung für n gilt, und schließlich mithilfe dieser Annahme zeigt, dass die Behauptung dann auch für $n + 1$ gilt. Auf vollständige Induktion gehen wir im Kapitel 3.2 noch ausführlich ein.

Sei also $n = 1$, d.h. α enthält genau eine Variable p. Da M_α unerfüllbar ist, müssen in M_α die Klauseln $\{ p \}$ und $\{ \neg p \}$ vorkommen. Diese resolvieren zur leeren Klausel, also ist $\diamond \in Res^*(M_\alpha)$. Wir nehmen nun an, dass für jede unerfüllbare Klauselmenge M_α mit n atomaren Formeln $p_1, \ldots, p_n$ gilt, dass $\diamond \in Res^*(M_\alpha)$ ist. Sei nun M_β eine Klauselmenge mit den atomaren Formeln $p_1, \ldots, p_{n+1}$. Wir bilden aus M_β zwei Klauselmengen M'_β und M''_β wie folgt: M'_β entsteht aus M_β durch Streichen jedes Vorkommens von p_{n+1} in einer Klausel sowie durch Streichen aller Klauseln, in denen $\neg p_{n+1}$ vorkommt. M'_β ist somit äquivalent zu der Klauselmenge, die entsteht, wenn man in M_β die atomare Formel p_{n+1} fest mit 0 belegt. M''_β entsteht aus M_β in analoger Weise, nur mit vertauschten Rollen von p_{n+1} und $\neg p_{n+1}$. M'_β und M''_β müssen notwendigerweise unerfüllbar sein. Denn, wenn wir annehmen, dass M'_β erfüllbar ist, dann gibt es

eine erfüllende Belegung $\mathcal{I} : \{ p_1, \ldots, p_n \} \to \{ 0, 1 \}$. Dann ist $\mathcal{I}'$ mit

$$\mathcal{I}'(x) = \begin{cases} \mathcal{I}(x), & \text{falls } x \in \{ p_1, \ldots, p_n \} \\ 0, & \text{falls } x = p_{n+1} \end{cases}$$

ein Modell für M_β, was aber einen Widerspruch zur Unerfüllbarkeit von M_β bedeutet. In analoger Weise führt die Annahme, dass M''_β erfüllbar ist, ebenfalls zu diesem Widerspruch. M'_β und M''_β sind also unerfüllbare Klauselmengen mit n atomaren Formeln. Wegen der Induktionsannahme, dass für jede unerfüllbare Klauselmenge M_α mit n atomaren Formeln $\diamond \in Res^*(M_\alpha)$ gilt, ist also $\diamond \in Res^*(M'_\beta)$ und $\diamond \in Res^*(M''_\beta)$.

Aus $\diamond \in Res^*(M'_\beta)$ folgt, dass es Klauseln $K_1, \ldots, K_r$ gibt mit $K_r = \diamond$, und es ist $K_i \in M'_\beta$ oder K_i ist Resolvent zweier Klauseln K_a undK_b mit $a, b < i$ und $1 \leq i \leq r$. Gleichermaßen muss es eine solche Folge $K'_1, \ldots, K'_s$ für M''_β geben.

Einige Klauseln K_i sind aus Klauseln in M_β durch Streichen von p_{n+1} entstanden. Wir machen dieses Streichen rückgängig, $K_i \cup \{ p_{n+1} \}$, und berücksichtigen p_{n+1} beim Resolvieren. Aus der Folge $K_1, \ldots, K_r$ entsteht dann eine neue Folge, und es folgt, dass $\diamond \in Res^*(M_\beta)$ oder dass $\{ p_{n+1} \} \in Res^*(M_\beta)$ ist. Analog folgt durch Wiedereinfügen von $\neg p_{n+1}$ in die Klauseln von $K'_1, \ldots, K'_s$, aus denen wir dieses Literal gestrichen haben, dass $\diamond \in Res^*(M_\beta)$ oder dass $\{ \neg p_{n+1} \} \in Res^*(M_\beta)$ ist. Ist $\diamond \in Res^*(M_\beta)$, dann ist nichts mehr zu zeigen. Ist $\{ p_{n+1} \} \in Res^*(M_\beta)$ und $\{ \neg p_{n+1} \} \in Res^*(M_\beta)$, dann gilt nach dem nächsten Resolutionsschritt $\diamond \in Res^*(M_\beta)$. Damit haben wir insgesamt die Behauptung gezeigt. □

1.5.3 Das Resolutionsverfahren

Resolutionsverfahren

Der Resolutionssatz ist die Grundlage für das *Resolutionsverfahren*: Gegeben sei eine Formel $\alpha \in \mathcal{A}$ in *KNF*.

1. Bilde die Klauselmenge M_α zu α.
2. Wende den Resolutionsoperator Res fortgesetzt auf M_α an, bis ein t erreicht ist, so dass $Res^l(M_\alpha) = Res^t(M_\alpha)$ für $l \geq t$, d.h. bestimme $Res^*(M_\alpha)$. Solch ein t existiert gemäß Satz 1.15 (Seite 49).
3. Falls $\diamond \in Res^*(M_\alpha)$ ist, dann ist α unerfüllbar, sonst erfüllbar.

Das Verfahren muss nicht immer so lange ausgeführt werden, bis die erneute Anwendung des Res-Operators keine neue Klauseln mehr erzeugt. Der Operator braucht nicht mehr angewendet zu werden, falls die leere Klausel $\diamond$ bereits erzeugt wurde. Zu diesem Zeitpunkt steht bereits fest, dass α unerfüllbar ist.

Definition 1.16 Eine *Deduktion der leeren Klausel* aus einer Klauselmenge M_α, $\alpha \in \mathcal{A}$ in *KNF*, ist eine Folge $K_1, K_2, \ldots, K_t$ von Klauseln, so dass gilt:

(1) K_t ist die leere Klausel und

(2) K_i, $1 \leq i \leq t$, ist entweder eine Klausel aus M_α ($K_i \in M_\alpha$) oder eine Resolvente von Klauseln K_r, K_s ($K_i = Res(K_r, K_s)$) mit $r, s \leq i$. □

Aus dem Resolutionssatz 1.16 (Seite 50) folgt unmittelbar

Folgerung 1.11 Eine Formel $\alpha \in \mathcal{A}$ in *KNF* ist unerfüllbar genau dann, wenn eine Deduktion der leeren Klausel aus M_α möglich ist. □

Deduktion der leeren Klausel

Beispiel 1.28 Wir betrachten die Formel (siehe auch Beispiel 1.27, Seite 49)

$$\alpha = (\neg r \vee p \vee q) \wedge \neg p \wedge (p \vee q \vee r) \wedge (\neg q \vee p)$$

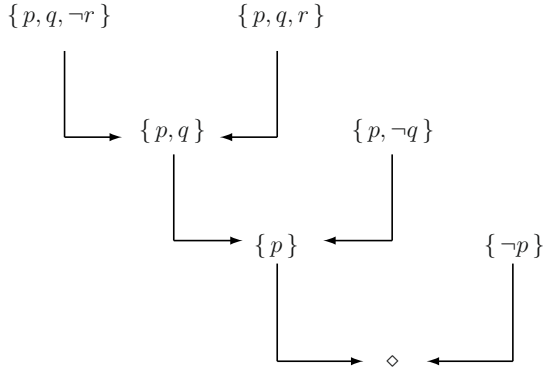
Es ist also $M_\alpha = \{\{p, q, \neg r\}, \{\neg p\}, \{p, q, r\}, \{p, \neg q\}\}$. Es gilt:

$K_1 = \{p, q, \neg r\}$	(Klausel aus M_α)
$K_2 = \{p, q, r\}$	(Klausel aus M_α)
$K_3 = \{p, q\}$	(Resolvente von K_1 und K_2)
$K_4 = \{p, \neg q\}$	(Klausel aus M_α)
$K_5 = \{p\}$	(Resolvente von K_3 und K_4)
$K_6 = \{\neg p\}$	(Klausel aus M_α)
$K_7 = \diamond$	(Resolvente von K_5 und K_6)

Die Klauselfolge $K_1, \ldots, K_7$ erfüllt die Bedingungen von Definition 1.16. Es existiert also eine Deduktion der leeren Klausel aus M_α, α ist somit gemäß Folgerung 1.11 unerfüllbar. □

Resolutionsgraph

Eine Deduktion kann mithilfe eines *Resolutionsgraphen* dargestellt werden. Im Folgenden ist der Resolutionsgraph für die Deduktion in Beispiel 1.28 gezeichnet.



Übungsaufgaben

1.18 Sei

$$\alpha = (p \vee \neg q \vee r) \wedge (q \vee r) \wedge (\neg p \vee r) \wedge (q \vee \neg r) \wedge \neg r$$

(1) Geben Sie M_α an!

(2) Berechnen Sie $Res^*(M_\alpha)$!

(3) Ist α erfüllbar?

(4) Falls α unerfüllbar ist, dann geben Sie eine Deduktion der leeren Klausel an und zeichnen Sie den entsprechenden Resolutionsgraphen!

1.19 Präsident I weiß:

a) „Wenn Professor W eine Vorlesung hält, dann kommen viele Studierende, falls kein schönes Wetter ist."

b) „Wenn Professor W eine Vorlesung hält, dann ist schlechtes Wetter."

c) I vermutet, dass die Aussage „Wenn Professor W eine Vorlesung hält, dann kommen viele Studierende." zutrifft.

Können Sie bestätigen, dass die Vermutung von Präsident I zutrifft?

Gehen Sie wie folgt vor:

(1) Benutzen Sie die Variablen P, S und W für die Aussagen
- P: „Professor W hält eine Vorlesung."
- S: „Viele Studierende kommen."
- W: „Es ist schönes Wetter."

(2) Formulieren Sie die Aussagen a) - c) durch geeignete aussagenlogische Verknüpfungen dieser Variablen!

(3) Transformieren Sie diese Verknüpfungen in Disjunktionen!

(4) Geben Sie eine Klauselmenge an, die die Vermutung von Präsident I beschreibt!

(5) Wenden Sie das Resolutionsverfahren an, und beantworten Sie mithilfe des Ergebnisses die Frage, ob die Vermutung von Präsident I zutrifft! □

1.5.4 Zusammenfassung

Das Resolutionsverfahren ist ein korrekter und vollständiger Kalkül, mit dem die Erfüllbarkeit einer aussagenlogischen Formel gezeigt werden kann. Die Formel muss in konjunktiver Normalform vorliegen, und diese wird in Form einer Klauselmenge dargestellt. Die Klauseln enthalten die jeweils disjunktiv verknüpften Literale der Formel. Der Resolutionskalkül basiert auf einer Inferenzregel, welche zwei Klauseln, die ein passendes Paar von Literalen enthalten (eine Klausel

enthält die Variable, die andere deren Negation) zu einer neuen Klausel vereinigt, wobei das passende Paar aus der Vereinigung ausgeschlossen wird. So werden schrittweise, sich „neutralisierende“ Literale ausgeschlossen. Eine gegebene aussagenlogische Formel ist unerfüllbar genau dann, wenn die wiederholte Anwendung der Resolution die leere Klausel erzeugt. Die Deduktion von Klauseln kann mithilfe eines Resolutionsgraphen dargestellt werden.

1.6 Hornlogik

Erfüllbarkeitsproblem der Aussagenlogik

Trotz aller Kalküle bleibt das *Erfüllbarkeitproblem der Aussagenlogik*, auch SAT-Problem[10] genannt, d.h. das Problem zu entscheiden, ob eine aussagenlogische Formel erfüllbar ist oder nicht, gemäß dem derzeitigen Stand der Erkenntnis ein schwieriges Problem. Nach diesem Erkenntnisstand gibt es kein deterministisches Entscheidungsverfahren, welches im Allgemeinen für eine aussagenlogische Formel mit n Variablen in weniger als größenordnungsmäßig 2^n Schritten feststellt, ob die Formel erfüllbar ist oder nicht (d h. im schlimmsten Fall muss die komplette Wahrheitstafel ausgerechnet werden).

Logikprogrammierung

Deduktive Datenbanken

Manche praktische Anwendungen erfordern allerdings gar nicht, dass die gesamte Sprache $\mathcal{A}$ der aussagenlogischen Formeln zur Verfügung steht, sondern kommen mit einer Untermenge davon aus. Solche Anwendungen sind z.B. die *Logikprogrammierung* sowie *deduktive Datenbanken*. Eine solche Datenbank besteht aus Fakten und Schlussregeln, mit denen aus den Fakten darin implizit vorhandenes Wissen abgeleitet werden kann. Betrachten wir als Beispiel ein Autobahnnetz, welches die Großstädte eines Landes miteinander verbindet. Als Fakten könnte die Datenbank Aussagen über die unmittelbare Verbindung von zwei Städten durch (mindestens) eine Autobahnstrecke enthalten. Die Aussage $c(A, B)$ soll z.B. bedeuten, dass von der Stadt A nach der Stadt B eine direkte Autobahnverbindung existiert. Die Datenbank enthält neben diesem extensionalen Wissen auch intensionales Wissen, nämlich z.B. dass, wenn B von A direkt über eine Autobahn erreicht werden kann, A auch direkt von B über eine Autobahn erreicht werden kann, oder dass es, wenn B von A und C von B erreicht werden kann, dann auch (indirekt) C von A. Mit folgenden Prädikaten könnten alle Erreichbarkeiten rekursiv beschrieben werden:

$$c(x, y) \rightarrow e(x, y) \tag{1.12}$$

$$e(x, y) \rightarrow e(y, x) \tag{1.13}$$

$$e(x, y), e(y, z) \rightarrow e(x, z) \tag{1.14}$$

$e(x, y)$ steht dabei für die Relation „y ist von x aus erreichbar.“ Die Subjunktion (1.12) drückt (die Rekursionsanfänge) aus: y ist von x erreichbar, wenn es eine

10 SAT steht für *Satisfiability*.

direkte Verbindung von x nach y gibt. Die Subjunktion (1.13) drückt aus, dass die Erreichbarkeitsrelation symmetrisch ist, d.h. wenn y von x aus erreichbar ist, dann auch x von y, und (1.14) beschreibt die Transitivität dieser Relation, d.h. die indirekten Verbindungen. Die Fakten über die direkten Verbindungen beschreiben zusammen mit den drei Subjunktionen die transitive Hülle der Relation e und damit alle auf den Fakten basierenden Autobahnverbindungen. Die transitive Hülle der Relation e enthält alle Verbindungen, sowohl die direkten als auch alle indirekt über mehrere Zwischenstationen möglichen Erreichbarkeiten. Die Begriffe Relation, Symmetrie, Transitivität und reflexive Hülle von Relationen werden in Kapitel 2.1 noch ausführlich behandelt.

Lernziele

Nach Durcharbeiten dieses Kapitels sollten Sie

- den syntaktischen Aufbau von Hornformeln kennen,
- ein Entscheidungsverfahren für die Erfüllbarkeit von Hornformeln erläutern und anwenden können,
- wissen, was ein kleinstes Modell einer aussagenlogischen Formel ist.

1.6.1 Hornformeln und Hornklauseln

Logische Formeln $\alpha \in \mathcal{A}$, die in oben beispielhaft genannten Anwendungen von Bedeutung sind, haben die folgenden Formen:

- $\alpha = p \in V$: α ist eine (atomare) Aussage, die ein *Faktum* beschreibt. α ist äquivalent zur Formel $1 \rightarrow p$. In deduktiven Datenbanken bilden diese Fakten das *extensionale Wissen*.

  Fakten

  Extensionales Wissen

- $\alpha = p \vee \neg q_1 \vee \ldots \vee \neg q_k$, $p, q_i \in V$, $1 \leq i \leq k$. Die Formel α ist äquivalent zu der Formel $(q_1 \wedge \ldots \wedge q_k) \rightarrow p$. α ist also eine Schlussfolgerung, die beschreibt, dass von den Fakten $q_1, \ldots, q_k$ auf die Aussage p geschlossen werden kann. Diese Schlussfolgerungen stellen das *intensionale Wissen* der Datenbank bereit.

  Intensionales Wissen

- $\alpha = \neg q_1 \vee \ldots \vee \neg q_k$, $q_i \in V$, $1 \leq i \leq k$. α ist äquivalent zu der Formel $0 \vee \neg q_1 \vee \ldots \vee \neg q_k$ und damit zu $(q_1 \wedge \ldots \wedge q_k) \rightarrow 0$. α beschreibt also, dass nicht alle Aussagen q_i, $1 \leq i \leq k$, zutreffen.

Wie wir sehen, lassen sich alle diese aussagenlogischen Formeln als Subjunktionen darstellen:

$$\begin{aligned} 1 &\rightarrow p \\ (q_1 \wedge \ldots \wedge q_k) &\rightarrow p \\ (q_1 \wedge \ldots \wedge q_k) &\rightarrow 0 \end{aligned}$$

Die folgende Definition fasst diese Fälle zusammen und zeichnet dadurch eine Untermenge von $\mathcal{A}$ aus.

Hornformel
Hornklausel

Definition 1.17 Eine aussagenlogische Formel $\alpha \in \mathcal{A}$ in konjunktiver Normalform heißt *Hornformel*[11] genau dann, wenn jede Klausel höchstens eine nicht negierte Variable enthält. Solche Klauseln werden *Hornklauseln* genannt. Mit $\mathcal{HF}$ bezeichnen wir die Menge aller Hornformeln in $\mathcal{A}$. □

Beispiel 1.29 Die Formel

$$\alpha = (p \vee \neg q) \wedge (\neg p \vee \neg q \vee \neg r \vee s) \wedge (\neg q \vee \neg r) \wedge \neg p \wedge r$$

ist eine Hornformel, in Klauselform

$$\alpha = \{ \{ p, \neg q \}, \{ \neg p, \neg q, \neg r, s \}, \{ \neg q, \neg r \}, \{ \neg p \}, \{ r \} \}$$

und als Subjunktionen

$$\alpha = (q \rightarrow p) \wedge ((p \wedge q \wedge r) \rightarrow p) \wedge ((q \wedge r) \rightarrow 0) \wedge (p \rightarrow 0) \wedge (1 \rightarrow r)$$

Die Formel $(\neg p \vee \neg q \vee s) \wedge (p \vee \neg q \vee \neg r \vee q)$ ist keine Hornformel, da in der zweiten Klausel zwei nicht negierte Variablen vorkommen. □

1.6.2 Erfüllbarkeit von Hornformeln

Das Erfüllbarkeitsproblem für Hornformeln, $HORNSAT$, ist deutlich effizienter lösbar als das Erfüllbarkeitsproblem SAT im Allgemeinen. Während, wie eingangs des Kapitels erwähnt, beim derzeitigen Kenntnisstand die Erfüllbarkeit einer Formel $\alpha \in \mathcal{A}$ im Allgemeinen exponentiellen Aufwand abhängig von der Anzahl der Variablen in α erfordert, gibt es Verfahren, mit denen die Erfüllbarkeit von Hornformeln $\beta \in \mathcal{HF}$ in polynomieller Zeit geprüft werden kann (etwa in der Größenordnung $m \cdot n$, wenn m die Anzahl der Klauseln und n die Anzahl der Literale in β sind).

Das Schema eines Verfahrens, mit dem geprüft werden kann, ob eine Formel $\alpha \in \mathcal{HF}$ erfüllbar ist, ist in Abbildung 1 gegeben.

Beispiel 1.30 Gegeben sei die Hornformel

$$\alpha = p \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg s \vee \neg p) \wedge (\neg t \vee s)$$

Wir wandeln die Klauseln um in Subjunktionen:

$$\begin{aligned} 1 &\rightarrow p \\ p &\rightarrow q \\ p \wedge q &\rightarrow r \\ s \wedge p &\rightarrow 0 \\ t &\rightarrow s \end{aligned} \tag{1.15}$$

11 Die Hornlogik ist benannt nach Alfred Horn (1918 – 2001), einem amerikanischen Mathematiker, der diese Variante der Aussagenlogik einführte. Die Hornlogik bildet die Grundlage für das logische Programmieren und logische Programmiersprachen, wie z.B. die Sprache PROLOG („Programming in Logic“).

(1) Enthält α eine Teilformel $1 \to p$, dann markiere alle Vorkommen von p in α.

(2) Führe die folgenden Schritte so lange aus, bis diese nicht mehr anwendbar sind:

   (i) Ist $(q_1 \wedge \cdots \wedge q_k) \to p$ eine Teilformel von α und alle q_i, $1 \leq i \leq k$, sind bereits markiert und p ist nicht markiert, dann markiere jedes Vorkommen von p in α.

   (ii) Ist $(q_1 \wedge \cdots \wedge q_k) \to 0$ eine Teilformel von α und alle q_i, $1 \leq i \leq k$, sind bereits markiert, dann stoppe das Verfahren – die Formel α ist unerfüllbar.

(3) Stoppe das Verfahren – die Formel α ist erfüllbar, und die Belegung $\mathcal{I}(q) = 1$ für alle markierten Variablen q und $\mathcal{I}(p) = 0$ für alle unmarkierten Variablen p in α ist ein Modell für α.

Abb. 1: Entscheidungsverfahren für die Erfüllbarkeit von Hornformeln

Wegen der Klausel (1.15) erhalten wir mit Verfahrensschritt (1) folgende roten Markierungen:

$$\begin{aligned} 1 &\to p \\ p &\to q \\ p \wedge q &\to r \\ s \wedge p &\to 0 \\ t &\to s \end{aligned} \tag{1.16}$$

Anwendung von Verfahrensschritt (2i) führt wegen der Klausel (1.16) zu folgenden blauen Markierungen:

$$\begin{aligned} 1 &\to p \\ p &\to q \\ p \wedge q &\to r \\ s \wedge p &\to 0 \\ t &\to s \end{aligned} \tag{1.17}$$

Anwendung von Verfahrensschritt (2i) führt wegen der Klausel (1.17) zu folgender grünen Markierung:

$$\begin{aligned} 1 &\to p \\ p &\to q \\ p \wedge q &\to r \\ s \wedge p &\to 0 \\ t &\to s \end{aligned}$$

Es ist kein Verfahrensschritt mehr anwendbar, die Formel ist erfüllbar, und die Belegung $\mathcal{I}(p) = \mathcal{I}(q) = \mathcal{I}(r) = 1, \mathcal{I}(s) = \mathcal{I}(t) = 0$ ist ein Modell für α. □

Übungsaufgaben

1.20 Testen Sie mit dem vorgestellten Verfahren die Erfüllbarkeit der folgenden Hornformeln:

(1) $\alpha = (r \vee \neg p \vee \neg q) \wedge (\neg s \vee t) \wedge (\neg p \vee \neg t) \wedge p \wedge (\neg p \vee q)$

(2) $\beta = p \wedge (\neg p \vee q) \wedge (\neg q \vee r \vee \neg s) \wedge (\neg p \vee \neg r) \wedge s$ □

Satz 1.17 Das in Abbildung 1 angegebene Verfahren entscheidet in korrekter Weise, ob eine Hornformel $\alpha \in \mathcal{HF}$ erfüllbar ist oder nicht. Werden die dabei markierten Variablen mit 1 belegt, dann stellt diese Belegung ein Modell für α dar. □

Folgerung 1.12 Enthält eine Formel $\alpha \in \mathcal{HF}$ keine Klausel der Art $(\neg q_1 \vee \ldots \vee \neg q_k)$[12], $k \geq 1$, dann ist α erfüllbar. □

1.6.3 Kleinste Modelle

Kleinstes Modell

Im Unterschied zu aussagenlogischen Formeln besitzen Hornformeln eindeutige *kleinste Modelle*. $\mathcal{I} \in \mathcal{I}_\alpha$ ist ein kleinstes Modell für eine Formel α, falls für jede Variable $v \in V_\alpha$ und jedes weitere Modell $\mathcal{I}' \in \mathcal{I}_\alpha$ gilt: Ist $\mathcal{I}'^*(v) = 1$, dann ist auch $\mathcal{I}^*(v) = 1$.

Beispiel 1.31 **a)** Die Formel $p \vee q$ ist keine Hornformel. Sie besitzt zwei kleinste Modelle: $\mathcal{I}_1$ mit $\mathcal{I}_1(p) = 1$ und $\mathcal{I}_1(q) = 0$ sowie $\mathcal{I}_2$ mit $\mathcal{I}_2(p) = 0$ und $\mathcal{I}_2(q) = 1$.

b) Die in Beispiel 1.30 und Übung 1.20 (1) gefundenen Modelle sind kleinste Modelle, denn mindestens die Variablen p, q und r (in beiden Fällen auch nicht mehr) müssen mit 1 belegt werden.

c) Betrachten wir die Hornformel mit den folgenden Klauseln:

$$1 \to s \qquad r \to s \qquad r \to t \qquad s \to q \qquad q \wedge t \to p$$

Dann ist $\mathcal{I}(s) = \mathcal{I}(q) = 1$ das kleinste Modell. □

12 Als Subjunktion geschrieben: $q_1 \wedge \ldots \wedge q_k \to 0$.

1.6.4 Zusammenfassung

Für die Anwendung von Logiken in der Praxis, z.B. in der Logikprogrammierung, der Wissensverarbeitung und der Künstlichen Intelligenz, ist es von wesentlicher Bedeutung, dass die Erfüllbarkeit von Formeln sowie Modelle effizient berechnet werden können. Mit der Hornlogik steht eine solche Logik zur Verfügung. Sie ist die Basis für viele Anwendungen. Hornformeln sind aussagenlogische Formeln in konjunktiver Normalform, deren Klauseln, Hornklauseln genannt, jeweils höchstens eine nicht negierte Variable enthalten. Erfüllbare Hornformeln besitzen zudem im Gegensatz zu aussagenlogischen Formeln ein kleinstes Modell.

1.7 Prädikatenlogik

Die Aussagenlogik ermöglicht die Verknüpfung von elementaren und zusammengesetzten Aussagen zu neuen zusammengesetzten Aussagen. Die Belegung der Variablen mit einem Wahrheitswert erlaubt die Berechnung des Wahrheitswertes der gesamten Aussage. Die Aussagenlogik ist allerdings zu arm, um z.B. zu beschreiben, dass die Addition natürlicher Zahlen kommutativ ist: „Für alle natürlichen Zahlen x und y gilt $x+y = y+x$“. x und y sind hier keine aussagenlogischen Variablen, die als Werte die Wahrheitswerte 0 und 1 annehmen können, sondern Variable für andere Werte wie z.B. Zahlenwerte. Operationssymbole wie $+$ drücken keine logische Verknüpfung aus, und das Symbol $=$ drückt eine Relation aus. Außerdem haben wir – umgangssprachlich – noch ausgedrückt, dass die Beziehung *für alle* x und y gelten soll.

Wir erweitern nun die Sprache der Aussagenlogik so, dass wir solche und weitere Eigenschaften ausdrücken können. Diese neue Sprache heißt Prädikatenlogik (erster Stufe). Wir gehen dabei in gleicher Weise wie bei der Definition der Aussagenlogik vor, indem wir nacheinander das Alphabet, die Syntax und schließlich die Semantik der Prädikatenlogik festlegen. Dabei werden wir nicht ganz so streng formal vorgehen, wie wir das bei der Aussagenlogik getan haben. Mithilfe von prädikatenlogischen Notationen werden wir dann in der Folge beschreibende Darstellungen von Mengen angeben. Ein Grund für die Beschäftigung mit Aussagen- und Prädikatenlogik ist, in der beschreibenden Darstellung $M = \{ x \mid p(x) \}$ einer Menge M die Eigenschaft p, welche festlegt, ob ein x zu M gehört oder nicht, möglichst präzise anzugeben.

Lernziele

Nach Durcharbeiten dieses Kapitels sollten Sie

- den syntaktischen Aufbau der Prädikatenlogik erster Stufe kennen,
- wissen, wie die Semantik prädikatenlogischer Formeln berechnet werden kann.

1.7.1 Alphabet der Prädikatenlogik

Das Alphabet der Prädikatenlogik besteht aus

- Symbolen für *Individuenvariablen*, dafür verwenden wir in der Regel kleine Buchstaben vom Ende des deutschen Alphabetes: $x, y, z, x_1, x_2, \ldots$
- Symbolen für *Individuenkonstanten*, dafür verwenden wir in der Regel kleine Buchstaben vom Anfang des deutschen Alphabetes: $a, b, c, a_1, a_2, \ldots$.
- *k-stellige Funktionssymbole*, die wir in der Regel mit $f^k, g^k, h^k, f_1^k, f_2^k, \ldots$ notieren. Dabei ist $k \in \mathbb{N}_0$.
- *k-stellige Prädikatensymbole*, die wir in der Regel mit $P^k, Q^k, R^k, P_1^k, P_2^k, \ldots$ notieren. Dabei ist $k \in \mathbb{N}_0$.
- den Symbolen $\neg, \wedge, \vee$ für *logische Junktoren*.
- den *Quantorsymbolen* $\forall$ (*Allquantor*, gesprochen „für alle“) und $\exists$ (*Existenzquantor*, gesprochen „es existiert“).
- den Klammersymbolen (und).

1.7.2 Syntax prädikatenlogischer Formeln

Prädikatenlogische Terme

Definition 1.18 Die Menge der *prädikatenlogischen Terme* ist gegeben durch:

(1) Jede Individuenvariable und jede Individuenkonstante ist ein Term.

(2) Sind $t_1, \ldots, t_n$ prädikatenlogische Terme und ist f^n ein n-stelliges Funktionssymbol, dann ist $f^n(t_1, \ldots, t_n)$ ein prädikatenlogischer Term.

(3) Genau die mit den Regeln (1) und (2) bildbaren Zeichenketten sind prädikatenlogische Terme. □

Beispiel 1.32 Die Individuenvariable x und die Individuenkonstante b sind Terme ebenso wie $f^2(x, b)$, $f^2(x, f^2(b, x))$ und $g^3(x, f^2(b, b), h^4(x, y, a, z))$. □

Atomare Formeln

Definition 1.19 Die Menge der *atomaren Formeln* ist gegeben durch:

(1) Sind $t_1, \ldots, t_n$ prädikatenlogische Terme und ist P^n ein n-stelliges Prädikatensymbol, dann ist $P^n(t_1, \ldots, t_n)$ eine atomare Formel.

(2) Genau die Zeichenketten, die mit der Regel (1) gebildet werden können, sind atomare Formeln. □

Beispiel 1.33 Die Zeichenketten $P^2(a, b)$, $P^3(a, g^4(x, y, z, x), f^1(z))$, $R^4(x, y, g^2(f^2(x, a), z))$ und $Q^2(f^2(x, y), f^2(y, x))$ sind atomare Formeln. □

Prädikatenlogische Formeln

Definition 1.20 Die Menge der *prädikatenlogischen Formeln* ist gegeben durch:

(1) Jede atomare Formel ist eine prädikatenlogische Formel.

(2) Sind α und β prädikatenlogische Formeln, dann auch $\neg\alpha$, $(\alpha \wedge \beta)$ sowie $(\alpha \vee \beta)$.

(3) Ist α eine prädikatenlogische Formel, dann auch $(\forall x\,\alpha)$ sowie $(\exists x\,\alpha)$.

(4) Genau die mit den Regeln (1) - (3) bildbaren Zeichenketten sind prädikatenlogische Formeln. □

Beispiel 1.34 Die Zeichenketten

$$(\forall x\,(\neg P^1(x)))$$
$$(\forall x\,(P^2(a, f^2(a, b)) \wedge Q^3(x, a, c))$$
$$(\forall x(\exists y\,(P^3(x, y, z))))$$
$$(\forall x(\forall y\,Q^2(f^2(x, y), f^2(y, x))))$$

sind prädikatenlogische Formeln. □

Variablen, die sich im Wirkungsbereich eines Quantors befinden, heißen *gebunden*, nicht gebundene Variablen heißen *frei*. So sind in der Formel

$$(\forall x(\exists y\,(P^3(x, y, z))))$$

die Variablen x und y gebunden, z ist frei.

Gebundene, freie Variable

Eine Formel heißt *geschlossen*, falls sie keine freie Variable enthält. Die Formel $(\forall x(\forall y\,Q^2(f^2(x, y), f^2(y, x))))$ ist ein Beispiel für eine geschlossene Formel. Geschlossene Formeln sind Aussagen, die wahr oder falsch sein können.

Geschlossene Formel

Gebundene Variablen können beliebig umbenannt werden, solange die Umbennung nicht zu einer freien Variablen führt. So kann in der Formel

$$(\exists x\,(P^2(f^2(x, y), z)))$$

die Variable x in q umbenannt werden: $(\exists q\,(P^2(f^2(q, y), z)))$. Eine Umbennung in z ist nicht erlaubt, denn die Formel bekommt dadurch eine andere Bedeutung: $(\exists z\,(P^2(f^2(z, y), z)))$.

Wie in der Aussagenlogik führen wir zwei weitere logische Verknüpfungen ein: Seien α und β zwei Formeln, dann schreiben wir $(\alpha \to \beta)$ für $(\neg\alpha \vee \beta)$ sowie $(\alpha \leftrightarrow \beta)$ für $((\alpha \to \beta) \wedge (\beta \to \alpha))$.

Sofern Bindungen von Quantoren und Junktoren eindeutig sind, wollen wir entsprechende Klammern weglassen. Wenn man zudem festlegt, dass die Quantoren die höchste Priorität besitzen, dass $\neg$ höhere Priorität als $\wedge$, $\wedge$ höhere Priorität als $\vee$ und $\vee$ höhere Priorität als $\to$ und $\leftrightarrow$ hat, können weitere entsprechende Klammern weggelassen werden. Mit diesen Vereinbarungen kann z.B. anstelle der

$$\text{Formel } (\forall x\,((\neg P^1(x)) \vee Q^1(x))) \text{ die Formel } \forall x\,(\neg P^1(x) \vee Q^1(x))$$

und anstelle von

$$(\forall x(\forall y\,(\neg P^2(x, y)) \vee (Q^2(y, z) \wedge (\neg R^2(x, z)))))$$

kann

$$\forall x(\forall y\, \neg P^2(x,y) \vee Q^2(y,z) \wedge \neg R^2(x,z))$$

geschrieben werden. □

Die Aussagenlogik kann als „Spezialfall“ der Prädikatenlogik aufgefasst werden: Wenn man in der Prädikatenlogik keine Individuenvariablen zulässt, wodurch die Quantoren überflüssig werden, und außerdem keine Individuenkonstanten, keine Funktionssymbole und nur 0-stellige Prädikate zulässt, erhält man genau die Aussagenlogik. Die 0-stelligen Prädikatensymbole übernehmen dabei die Rolle der aussagenlogischen Variablen.

1.7.3 Semantik prädikatenlogischer Formeln

Um die Bedeutung einer prädikatenlogischen Formel zu bestimmen, müssen zunächst Belegungen vorgenommen werden:

Grundmenge
Universum

- Es muss eine *Grundmenge* (auch *Universum* genannt) ausgewählt werden. Mit ihren Elementen müssen Individuenkonstanten und freie Individuenvariablen belegt werden.
- Jedem k-stelligen Funktionssymbol muss eine k-stellige Funktion über der Grundmenge zugeordnet werden, und
- jedem k-stelligen Prädikatensymbol muss eine k-stellige Relation über der Grundmenge zugeordnet werden.

Beispiel 1.35 **a)** Wählen wir als Grundmenge $\mathbb{N}_0$ und ordnen wir dem Funktionssymbol f^2 die Addition $+$ sowie dem Prädikatensymbol Q^2 die Gleichheitsrelation $=$ zu, dann interpretieren wir die geschlossene Formel

$$\forall x \forall y\, (Q^2(f^2(x,y), f^2(y,x))) \text{ durch } \forall x \forall y\, (=(+(x,y), +(y,x)))$$

Schreiben wir Gleichheits- und Additionssymbol wie üblich infix, d.h. zwischen die Operanden, dann erhalten wir die Formel

$$\forall x \forall y\, (x + y = y + x)$$

die das Kommutativgesetz der Addition in der Menge der natürlichen Zahlen ausdrückt.

b) Betrachten wir die Formel $\exists x \forall y\, R^2(f^2(a,x), y))$ und ordnen dem Prädikatensymbol R^2 die Relation $<$ zu, dem Funktionssymbol f^2 die Multiplikation und der Individuenkonstante die Zahl 5, dann erhalten wir die Formel $\exists x \forall y\, (5x < y)$.

Wählen wir als Grundmenge $\mathbb{N}$, dann ist diese Formel nicht wahr, denn z.B. zu $y = 1$ gibt es keine natürliche Zahl x, so dass $5x < 1$ ist. Wählen wir als Grundmenge $\mathbb{Z}$, dann ist diese Formel wahr. □

Wenn die Grundmenge U ausgewählt wurde und Belegungen $\mathcal{I}$ der Individuenkonstanten, der freien Individuenvariablen, der Funktionssymbole sowie der

Prädikatensymbole vorgenommen sind, geschieht die Berechnung $\mathcal{I}^*$ des Wahrheitswertes analog zur Interpretation aussagenlogischer Formeln:

(i) Für einen prädikatenlogischen Term $f^k(t_1, \ldots, t_k)$ gilt

$$\mathcal{I}^*(f^k(t_1, \ldots, t_k)) = \mathcal{I}(f^k)(\mathcal{I}^*(t_1), \ldots, \mathcal{I}^*(t_k))$$

Die Belegung $\mathcal{I}(f^k)$ des Funktionssymbols f^k wird auf das Ergebnis der Interpretationen der Terme $t_1, \ldots, t_k$ angewendet.

(ii) Für eine atomare Formel $P^m(t_1, \ldots, t_m)$ gilt

$$\mathcal{I}^*(P^m(t_1, \ldots, t_m)) = \begin{cases} 1, & ((\mathcal{I}^*(t_1), \ldots, \mathcal{I}^*(t_m)) \in \mathcal{I}(P^m) \\ 0, & \text{sonst} \end{cases}$$

Falls die Interpretationen der Terme in der Relation stehen, die sich durch die Belegung $\mathcal{I}(P^m)$ des Prädikatensymbols ergibt, liefert die Interpretation der atomaren Formel den Wahrheitswert 1, sonst den Wahrheitswert 0.

(iii) Nachdem die Interpretation der atomaren Bestandteile von prädikatenlogischen Formeln festgelegt ist, erfolgt nun die Berechnung zusammengesetzter Formeln. Seien α und β prädikatenlogische Formeln, dann gilt:

(1) $\mathcal{I}^*(\neg\alpha) = 1 - \mathcal{I}^*(\alpha)$.

(2) $\mathcal{I}^*(\alpha \wedge \beta) = \mathit{min}\{\mathcal{I}^*(\alpha), \mathcal{I}^*(\beta)\}$.

(3) $\mathcal{I}^*(\alpha \vee \beta) = \mathit{max}\{\mathcal{I}^*(\alpha), \mathcal{I}^*(\beta)\}$.

(4)

$$\mathcal{I}^*(\exists x\, \alpha) = \begin{cases} 1, & \text{falls ein } a \in U \text{ existiert mit } \mathcal{I}^*_{x,a}(\alpha) = 1 \\ 0, & \text{sonst} \end{cases}$$

(5)

$$\mathcal{I}^*(\forall x\, \alpha) = \begin{cases} 1, & \text{falls für alle } a \in U \text{ gilt } \mathcal{I}^*_{x,a}(\alpha) = 1 \\ 0, & \text{sonst} \end{cases}$$

Dabei gilt

$$\mathcal{I}^*_{x,a}(y) = \begin{cases} \mathcal{I}(y), & y \neq x \\ a, & \text{sonst} \end{cases}$$

$\mathcal{I}^*_{x,a}(\alpha)$ für $a \in U$ bedeutet also, dass jedes Vorkommen von x in α mit dem Wert a aus der Grundmenge U belegt wird.

Wir werden in den folgenden Kapiteln prädikatenlogische Formeln benutzen, um mathematische Sachverhalte zu beschreiben. Dabei werden wir die Formeln „pragmatisch“ verwenden, d.h. wir werden in der Regel nicht mit Individuenkonstanten, Funktions- und Prädikatensymbolen arbeiten, sondern in den Formeln direkt konkrete Belegungen, d.h. Werte und Funktionen bzw. Relationen notieren. Ebenso liegt in der Regel durch den gegbenen Kontext fest, welche

Grundmenge jeweils vorliegt. Oft werden wir die Grundmenge auch in der Formel selbst angeben. Im Übrigen werden die Formulierungen in Definitionen, Sätzen usw. selten rein formal sein, sondern aus formalen und informalen Teilen bestehen.

Als Beispiel für unsere Sprech- und Schreibweise sei der Satz über die Division mit Rest ganzer Zahlen hier aufgeführt:

> $\forall a \in \mathbb{Z}$ und $\forall b \in \mathbb{N}\ \exists q \in \mathbb{Z}$ und $\exists r \in \mathbb{N}_0$, so dass gilt: $a = bq + r$ mit $0 \leq r < b$.
>
> Ausschließlich in Worten: Zu jeder ganzen Zahl a und zu jeder natürlichen Zahl b ungleich Null existieren zwei ganze Zahlen q und r, so dass a bei Division durch b den ganzzahligen Quotienten q ergibt mit einem positiven Rest r, der kleiner als der Divisor b ist.

Anstelle von $\forall x \in M$ für eine Menge M schreiben wir im Folgenden auch $x \in M$.

Insbesondere können wir nun darstellende Beschreibungen von Mengen präziser als bisher angeben. Im Abschnitt 1.1.2 haben wir die allgemeine Form der beschreibenden Darstellung angegeben: $M = \{\, x \mid p(x) \,\}$. x ist ein Platzhalter – eine Variable – für die Elemente von M, und p ist ein Prädikat. Alle Elemente einer Grundmenge, die das Prädikat wahr machen, gehören zu M.

Beispiel 1.36 Wir geben für die in Beispiel 1.3 auf Seite 7 „halbformal“ dargestellten Mengen formale Beschreibungen an:

$$\begin{aligned} A &= \{\, x \mid x \in \mathbb{P} \wedge x \leq 11 \,\} \\ G &= \{\, x \mid x \in \mathbb{N}_0 \wedge 2x = 10 \,\} \\ H &= \{\, (x, y) \mid x \in \mathbb{N}_0 \wedge y \in \mathbb{N}_0 \wedge x + y = 6 \,\} \\ T_{64} &= \{\, y \mid (y \geq 0) \wedge (\exists q \in \mathbb{N}(y \cdot q = 64)) \,\} \end{aligned}$$

Da die formalen Beschreibungen zumeist aus konjunktiv verknüpften Teilformeln bestehen, lässt man das Symbol $\wedge$ weg und schreibt an dessen Stelle ein Komma. So stellt man H wie folgt dar

$$H = \{\, (x, y) \mid x \in \mathbb{N}_0,\ y \in \mathbb{N}_0,\ x + y = 6 \,\}$$

oder noch kürzer

$$H = \{\, (x, y) \mid x, y \in \mathbb{N}_0,\ x + y = 6 \,\}$$

In der „alltäglichen Praxis“ werden beschreibende Mengendarstellungen in der Regel in einer Mischung von informalen und formalen Beschreibungen angegeben, wie z.B.

$$A = \{\, x \mid x \text{ prim},\ x \leq 11 \,\}$$

Maßstab für die Art und Weise der Beschreibung einer Menge M ist, dass aus der Beschreibung – möglicherweise zudem mithilfe des Kontextes – klar wird, welche Elemente zu M gehören. □

Man kann die Begriffe Erfüllbarkeit, Modell, Tautologie, Kontradiktion, Implikation (syntaktische und semantische Folgerungsbegriffe) und Äquivalenz auch für prädikatenlogische Formeln analog zu aussagenlogischen Formeln einführen.

Viele Ergebnisse, die wir für die Aussagenlogik in Kapitel 1.2 betrachtet haben, gelten analog auch für die Prädikatenlogik, insbesondere auch die Äquivalenzen aus Satz 1.5 (Seite 31). Darüber hinaus gelten in der Prädikatenlogik noch folgende Äquivalenzen:

$$\begin{aligned}
\forall x \forall y\, \alpha &\equiv \forall y \forall x\, \alpha \\
\exists x \exists y\, \alpha &\equiv \exists y \exists x\, \alpha \\
\forall x\, \alpha \wedge \forall x\, \beta &\equiv \forall x\, (\alpha \wedge \beta) \\
\exists x\, \alpha \wedge \exists x\, \beta &\equiv \exists x\, (\alpha \wedge \beta) \\
\forall x\, \alpha &\equiv \neg \exists\, \neg \alpha \\
\exists x\, \alpha &\equiv \neg \forall\, \neg \alpha
\end{aligned}$$

Im Gegensatz zur Aussagenlogik ist die Erfüllbarkeit prädikatenlogischer Formeln nicht entscheidbar, sofern sie die Arithmetik natürlicher Zahlen umfasst. Mithilfe einer Wahrheitstafel oder mithilfe des Resolutionskalküls kann z.B. festgestellt werden, ob eine aussagenlogische Formel erfüllbar ist oder nicht. Man kann hingegen beweisen, dass es für die Prädikatenlogik keinen Algorithmus geben kann, der für jede beliebige prädikatenlogische Formel feststellt, ob diese erfüllbar ist oder nicht.

1.7.4 Weitere Logiken

Neben der Aussagenlogik und der Prädikatenlogik 1. Stufe gibt es weitere Logiken, die nicht nur von theoretischem sondern auch von praktischem Interesse sind. In der Prädikatenlogik 2. Stufe ist es unter anderem auch erlaubt, über Mengen, Funktionen und Prädikaten zu quantifizieren. In der Prädikatenlogik 1. Stufe darf nur über Individuenvariablen quantifiziert werden.

Bei der Informations- und Wissensverarbeitung, der Konzipierung und Implementierung verteilter Prozesse oder bei der Verifikation von Systemen finden Logiken wie modale, nichtmonotone, temporale, mehrwertige oder Fuzzy-Logiken Anwendung. In modalen Logiken gibt es neben Quantoren und „klassischen“ Junktoren Operatoren wie $[\,]$ und $\Diamond$: Für eine Formel α bedeutet $[\alpha]$, dass α notwendigerweise wahr ist, und $\Diamond\alpha$ bedeutet, dass α möglicherweise gilt. Mit diesen Operatoren kann man z.B. zeitliches Planen beschreiben: $[\alpha]$ modelliert, dass α immer gilt oder dass α sicheres Wissen beschreibt. $\Diamond\alpha$ modelliert, dass α manchmal gilt oder dass α Meinungen beschreibt.

„Klassische“ Logiken wie die Aussagenlogik und die Prädikatenlogiken sind monoton. Das bedeutet, wenn für Klauselmengen A, B und C gilt, dass alle

Klauseln von A auch zu B gehören und $A \vdash C$ gilt, dass dann auch $B \vdash C$ gilt. Konklusionen gelten also weiter, wenn man zu den Prämissen Klauseln hinzufügt. Bei realen Problemen gibt es aber oft Ausnahmen, welche diese Monotonie zerstören. So trifft die Aussage „alle Vögel fliegen“, die etwa durch die Formel $\forall x\,(\mathit{Vogel}(x) \rightarrow \mathit{Fliegen}(x))$, beschrieben werden kann, zwar auf fast alle Vögel zu, aber es gibt Vogelarten wie Strauße und Pinguine, die nicht fliegen können. Um solche Anwendungen adäquat modellieren zu können, müssen nicht monotone Schlussfolgerungen möglich sein.

Mehrwertige Logiken lassen mehr als zwei Wahrheitswerte und Verknüpfungen dafür zu, und in der Fuzzy-Logik werden Wahrheitswerte durch (stetige) Funktionen beschrieben, die den Zugehörigkeitsgrad eines Elementes zu einer Menge festlegen.

1.7.5 Zusammenfassung

Aussagenlogische Ausdrücke bestehen aus Konstanten und Variablen, die mit aussagenlogischen Operationen miteinander verknüpft werden können. Damit können aber nur sehr einfache Prädikate formuliert werden. Um auch Aussagen über Mengen und Beziehungen zwischen Elementen von Mengen auszudrücken, benötigt man weitere sprachliche Mittel wie weitere Konstanten sowie Quantoren, Funktionen und Relationen. Prädikatenlogische Ausdrücke werden – zusammen mit natürlichsprachlichen Formulierungen – verwendet, um beschreibende Darstellungen von Mengen anzugeben; so z.B. auch für die Definition der Teilmengenbeziehung und der Mengenverknüpfungen. Eigenschaften dieser Begriffe ergeben sich dann aus den entsprechenden Eigenschaften der definierenden Prädikate.

1.8 Beweismethoden

Wie bereits erwähnt benutzen wir die Prädikatenlogik, um Begriffe mathematisch zu definieren, Eigenschaften dafür zu formulieren und diese zu beweisen, d.h. deren Gültigkeit herzuleiten. In diesem Kapitel stellen wir gängige Beweisverfahren vor.

Mathematischer Satz
Theorem

Die meisten mathematischen Sätze (Theoreme; Hilfssätze, Lemmata; Folgerungen, Korrollare) haben die Form:

$$\alpha \Rightarrow \beta$$

Voraussetzung
Behauptung

Dabei sind α und β Formeln. α heißt *Voraussetzung* (Vorbedingung, Hypothese) und β *Behauptung* (Nachbedingung, Folgerung) des Satzes. Sätze, die Äquiva-

lenzen behaupten ($\alpha \Leftrightarrow \beta$) sind äquivalent zu Folgerungen in beide Richtungen ($\alpha \Rightarrow \beta \wedge \beta \Rightarrow \alpha$), so dass wir uns auf Folgerungen beschränken können.

Zu zeigen, dass $\alpha \Rightarrow \beta$ gilt, bedeutet zu zeigen, dass $\alpha \rightarrow \beta$ eine Tautologie ist, d.h. dass $\alpha \rightarrow \beta$ immer wahr ist. Wir geben im Folgenden vier Verfahren an, die dazu verwendet werden können:

1. direkter Beweis,
2. indirekter Beweis,
3. Widerspruchsbeweis,
4. Beweis durch Ringschluss.

Eine weitere sehr wichtige Beweismethode, das Prinzip der vollständigen Induktion, mit dem gezeigt werden kann, dass ein Prädikat $P(n)$ wahr wird für alle natürlichen Zahlen $n \in \mathbb{N}_0$, wird in Abschnitt 3.2 behandelt.

Lernziele

Nach Durcharbeiten dieses Kapitels sollten Sie

- die oben aufgelisteten Beweismethoden erklären und anwenden können.

1.8.1 Direkter Beweis

Ein direkter Beweis eines Theorems $\alpha \Rightarrow \beta$ ist eine Folge von Aussagen

$$\gamma_1, \gamma_2, \ldots, \gamma_n = \beta$$

wobei für jedes i mit $1 \leq i \leq n$ gilt: $\gamma_i = \alpha$ oder γ_i ist eine (bereits bewiesene) bekannte Aussage oder $\gamma_{j_1} \wedge \gamma_{j_2} \wedge \ldots \wedge \gamma_{j_r} \Rightarrow \gamma_i$ mit $j_1, j_2, \ldots, j_r < i$. Bei den Zwischenschritten können also Kombinationen von vorher – im Beweis selbst oder im Rahmen anderer Beweise – etablierten Aussagen verwendet werden.

Beispiel 1.37 Wir wollen den Satz „Ist eine natürliche Zahl durch 2 und durch 3 teilbar, dann ist sie auch durch 6 teilbar“ direkt beweisen. Zunächst formalisieren wir diesen umgangssprachlich formulierten Satz:

$$\frac{x}{2} \in \mathbb{N}_0 \wedge \frac{x}{3} \in \mathbb{N}_0 \Rightarrow \frac{x}{6} \in \mathbb{N}_0$$

Dabei ist

$$\alpha = \left(\frac{x}{2} \in \mathbb{N}_0 \wedge \frac{x}{3} \in \mathbb{N}_0\right)$$

die Voraussetzung und

$$\beta = \left(\frac{x}{6} \in \mathbb{N}_0\right)$$

die Behauptung. Wir betrachten nun die folgenden Aussagen:

$$\gamma_1 = (\exists y \in \mathbb{N}_0\,(x = 2y) \wedge \exists z \in \mathbb{N}_0\,(x = 3z))$$
$$\gamma_2 = (\exists y, z \in \mathbb{N}_0\,(2y = 3z))$$
$$\gamma_3 = (\exists k \in \mathbb{N}_0\,(z = 2k))$$
$$\gamma_4 = \exists z, k \in \mathbb{N}_0\,(x = 3z \wedge z = 2k)$$
$$\gamma_5 = \exists k \in \mathbb{N}_0\,(x = 2 \cdot 3 \cdot k))$$
$$\gamma_6 = \exists k \in \mathbb{N}_0\,(x = 6k))$$

Die beiden Folgerungen: $\alpha \Rightarrow \gamma_1$ und $\gamma_1 \Rightarrow \gamma_2$ gelten offensichtlich. Da $2y = 3z$, muss $3z$ und damit z eine gerade Zahl sein, also gilt: $\gamma_2 \Rightarrow \gamma_3$. Die Folgerungen $\gamma_3 \Rightarrow \gamma_4$, $\gamma_4 \Rightarrow \gamma_5$ und $\gamma_5 \Rightarrow \gamma_6$ sind offensichtlich. Insgesamt erhalten wir:

$$\alpha \Rightarrow \gamma_1 \Rightarrow \gamma_2 \Rightarrow \gamma_3 \Rightarrow \gamma_4 \Rightarrow \gamma_5 \Rightarrow \gamma_6 \Rightarrow \beta$$

und damit $\alpha \Rightarrow \beta$, was zu beweisen war (w.z.b.w.).[13] □

Übungsaufgaben

1.21 Beweisen Sie, dass für $a, b \in \mathbb{R}_+$ gilt

$$\frac{a+b}{2} \geq \sqrt{a \cdot b}$$

(das arithmetische Mittel von zwei positiven Zahlen ist größer gleich deren geometrischem Mittel)! □

1.8.2 Indirekter Beweis

Dem indirekten Beweis liegt die folgende Äquivalenz zugrunde:

$$(\alpha \Rightarrow \beta) \Leftrightarrow (\neg\beta \Rightarrow \neg\alpha)$$

Manchmal ist es tatsächlich einfacher, bequemer oder schneller, die rechte anstelle der linken Folgerung zu beweisen.

Beispiel 1.38 Als Beispiel wollen wir eine weitere Teilbarkeitsregel beweisen: Wenn die letzten beiden Ziffern einer natürlichen Zahl z als Zahl betrachtet durch 4 teilbar sind, dann ist auch die Zahl z durch 4 teilbar. Wir formalisieren:

$$\alpha = (x \in \mathbb{N}_{0,99}) \wedge \left(\frac{x}{4} \in \mathbb{N}_0\right) \wedge (y \in \mathbb{N}_0)$$
$$\beta = (x \in \mathbb{N}_{0,99}) \wedge (y \in \mathbb{N}_0) \wedge \left(\frac{100y + x}{4} \in \mathbb{N}_0\right)$$

13 Auch: *qed* für *quod erat demonstrandum*.

Anstelle $\alpha \Rightarrow \beta$ direkt zu beweisen, beweisen wir $\neg\beta \Rightarrow \neg\alpha$. Es ist:

$$\neg\beta = (x \notin \mathbb{N}_{0,99}) \vee (y \notin \mathbb{N}_0) \vee \left(\frac{100y + x}{4} \notin \mathbb{N}_0\right)$$
$$\neg\alpha = (x \notin \mathbb{N}_{0,99}) \vee \left(\frac{x}{4} \notin \mathbb{N}_0\right) \vee (y \notin \mathbb{N}_0)$$

Da sowohl $x \notin \mathbb{N}_{0,99}$ als auch $y \notin \mathbb{N}_0$ mit *falsch* zu bewerten sind, wird in beiden Disjunktionen der Wahrheitswert durch den jeweils verbleibenden Teilausdruck bestimmt (siehe Satz 1.5, Seite 31, Unerfüllbarkeitsregeln), d.h. wir müssen nur noch

$$\frac{100y + x}{4} \notin \mathbb{N}_0 \Rightarrow \frac{x}{4} \notin \mathbb{N}_0$$

beweisen. Dies tun wir direkt:

$$\begin{aligned}\frac{100y + x}{4} \notin \mathbb{N}_0 &\Rightarrow \frac{100y}{4} + \frac{x}{4} \notin \mathbb{N}_0 \\ &\Rightarrow 25y + \frac{x}{4} \notin \mathbb{N}_0 \\ &\Rightarrow \frac{x}{4} \notin \mathbb{N}_0\end{aligned}$$

Da jede Zahl $z \in \mathbb{N}_0$ sich darstellen lässt als $z = 100y + x$ mit $y \in \mathbb{N}_0$ und $x \in \mathbb{N}_{0,99}$, ist die Behauptung bewiesen. □

1.8.3 Beweis durch Widerspruch

Dem Widerspruchsbeweis liegt die Äquivalenz

$$(\alpha \rightarrow \beta) \Leftrightarrow ((\alpha \wedge \neg\beta) \rightarrow \neg\alpha) \tag{1.18}$$

zugrunde. Um diesen Beweis zu führen, nehmen wir also sowohl die Voraussetzung α als auch die Negation der Folgerung β, also $\neg\beta$, als wahr an und versuchen, daraus einen Widerspruch zu α zu folgern.

Beispiel 1.39 Ein „klassisches“ Beispiel für einen Widerspruchsbeweis ist zu zeigen, dass $\sqrt{2}$ keine rationale Zahl ist. Genauer lautet diese Aussage: Wenn p und q teilerfremde natürliche Zahlen sind, dann ist $\sqrt{2} \neq \frac{p}{q}$. Es seien also:

$$\alpha = \left(\forall k \in \mathbb{N}_2 \left(\frac{p}{k} \notin \mathbb{N}_0 \vee \frac{q}{k} \notin \mathbb{N}_0\right)\right) \quad (p \text{ und } q \text{ sind teilerfremd}) \tag{1.19}$$
$$\beta = \left(\sqrt{2} \neq \frac{p}{q}\right) \quad (\sqrt{2} \text{ ist nicht rational})$$

Wir nehmen nun an, dass es teilerfremde Zahlen p und q gibt mit $\sqrt{2} = \frac{p}{q}$, d.h. wir nehmen an, dass $\alpha \wedge \neg\beta$ gilt.

Aus $\sqrt{2} = \frac{p}{q}$ folgt, dass $2 = \frac{p^2}{q^2}$ ist. Daraus folgt, dass $p^2 = 2q^2$ gilt, und daraus, dass 2 ein Teiler von p^2 ist, und daraus, dass 2 ein Teiler von p ist. Hieraus folgt, dass $2 \cdot 2$ ein Teiler von $p \cdot p = p^2$ und damit 4 ein Teiler von $2q^2$ ist. Hieraus folgt, dass 2 ein Teiler von q^2 und damit ein Teiler von q ist. Aus diesen Schlussfolgerungen folgt, dass 2 ein Teiler von p und von q ist, womit p und q nicht teilerfremd sind, d.h.:

$$\exists k \in \mathbb{N}_2 \left(\frac{p}{k} \in \mathbb{N}_0 \wedge \frac{q}{k} \in \mathbb{N}_0 \right) \tag{1.20}$$

Diese Aussage ist gleich $\neg\alpha$, also ein Widerspruch zu α, vergleiche (1.19).

Damit haben wir $(\alpha \wedge \neg\beta) \Rightarrow \neg\alpha$ gezeigt, und damit ist wegen der Äquivalenz (1.18) die Folgerung $\alpha \Rightarrow \beta$ bewiesen. □

Varianten von Widerspruchsbeweisen basieren auf den Äquivalenzen:

$$(\alpha \to \beta) \Leftrightarrow ((\alpha \wedge \neg\beta) \to \beta)$$
$$(\alpha \to \beta) \Leftrightarrow ((\alpha \wedge \neg\beta) \to (\gamma \wedge \neg\gamma))$$
$$(\alpha \to \beta) \Leftrightarrow ((\alpha \wedge \neg\beta) \to 0)$$

Übungsaufgaben

1.22 (1) Beweisen Sie die Behauptung aus Übung 1.21 durch Widerspruchsbeweis!

(2) Beweisen Sie durch direkten sowie durch Widerspruchsbeweis, dass das Produkt zweier ungerader Zahlen wieder ungerade ist: $x, y \in \mathbb{U}_+ \Rightarrow xy \in \mathbb{U}_+$! □

1.8.4 Ringschluss

Um die Äquivalenz der Formeln $\alpha_1, \ldots, \alpha_k$, d.h.

$$\alpha_1 \Leftrightarrow \alpha_2 \Leftrightarrow \ldots \Leftrightarrow \alpha_k,\ k \geq 2$$

zu beweisen, kann man anstelle der $2(k-1)$ Folgerungen $\alpha_i \Rightarrow \alpha_{i+1}$ und $\alpha_{i+1} \Rightarrow \alpha_i$, $1 \leq i \leq k-1$, die $k-1$ Folgerungen $\alpha_i \Rightarrow \alpha_{i+1}$, $1 \leq i \leq k-1$, und die Folgerung $\alpha_k \Rightarrow \alpha_1$ zeigen. Insgesamt ist zum Beweis der Äquivalenz der Formeln α_i, $1 \leq i \leq k$, also die Folge der k Implikationen

$$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_k \Rightarrow \alpha_1$$

zu zeigen.

Satz 1.18 Sind $\alpha_1, \ldots, \alpha_k$, $k \geq 2$, prädikatenlogische Formeln, dann ist

$$\alpha_1 \Leftrightarrow \alpha_2 \Leftrightarrow \ldots \Leftrightarrow \alpha_k \tag{1.21}$$

äquivalent zu

$$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_k \Rightarrow \alpha_1 \tag{1.22}$$

Beweis Aus Beispiel 1.10 c) auf Seite 25 folgt:

$$(\alpha \Rightarrow \beta \Rightarrow \gamma) \Rightarrow (\alpha \Rightarrow \gamma) \tag{1.23}$$

Wir zeigen zunächst, dass (1.22) aus (1.21) folgt. Aus (1.21) folgt, dass

$$\alpha_k \Rightarrow \alpha_{k-1} \Rightarrow \ldots \Rightarrow \alpha_2 \Rightarrow \alpha_1 \tag{1.24}$$

und dass

$$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_{k-1} \Rightarrow \alpha_k \tag{1.25}$$

gilt. Aus (1.23) und (1.24) folgt, dass $\alpha_k \Rightarrow \alpha_1$ gilt. Hieraus folgt mit (1.25), dass (1.22) gilt. Damit ist gezeigt, dass (1.22) aus (1.21) folgt.

Jetzt zeigen wir, dass (1.21) aus (1.22) folgt. Aus (1.22) und (1.23) folgt $\alpha_1 \Rightarrow \alpha_i$, $1 \leq i \leq k$, und $\alpha_j \Rightarrow \alpha_1$, $1 \leq j \leq k$, sowie $\alpha_i \Rightarrow \alpha_j$, $1 \leq i, j \leq k-1$, $i \leq j$. Es folgt, dass $\alpha_j \Rightarrow \alpha_1 \Rightarrow \alpha_i$ für $1 \leq i, j \leq k$ mit $i \leq j$ gilt. Es folgt, dass

$$\alpha_k \Rightarrow \alpha_{k-1} \Rightarrow \ldots \Rightarrow \alpha_2 \Rightarrow \alpha_1$$

gilt. Mit der Voraussetzung (1.22) folgt hieraus die Behauptung (1.21). □

1.8.5 Zusammenfassung

Eine mathematische Behauptung $\alpha \Rightarrow \beta$ mit der Voraussetzung α und der Behauptung β ist eine logische Schlussfolgerung, die bewiesen, d.h. deren Gültigkeit gezeigt werden muss. Aufgrund von logischen Äquivalenzen kann der direkte Beweis von $\alpha \Rightarrow \beta$ auch mit anderen Methoden gezeigt werden, etwa durch indirekten Beweis oder durch einen Widerspruchsbeweis. Eine Äquivalenz von mehreren Formeln kann durch einen Ringschluss gezeigt werden.

1.9 Operationen auf Mengen

Wir kehren nun zu Mengen zurück und benutzen prädikatenlogische Formeln, um Teilmengenbeziehungen zwischen Mengen sowie Verknüpfungen von Mengen zu definieren. Dabei benutzen wir folgende formale Schreibweise bei der Definition von neuen Begriffen:

$$\alpha :\Leftrightarrow \beta$$

Dabei ist β eine Formel, in der bereits definierte Begriffe und Symbole verwendet werden, um den neuen Begriff oder das neue Symbol oder die neue Schreibweise α präzise festzulegen.

In vielen Anwendungsbereichen müssen Datenmengen miteinander verknüpft werden. Hat man z.B. eine Datenbasis, in der die Kunden pro Tag festgehalten werden, und möchte man daraus eine Auflistung aller Kunden einer Woche haben, dann müssen die Tagesmengen vereinigt werden. Oder möchte man wissen, welche Kunden an allen Tagen Bestellungen aufgegeben haben, dann müssen die gemeinsamen Elemente der Tagesmengen bestimmt werden.

In diesem Kapitel definieren wir mithilfe logischer Ausdrücke elementare Mengenoperationen und betrachten ihre grundlegenden Eigenschaften.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- den Teilmengenbegriff und dessen grundlegende Eigenschaften kennen,
- wissen, wie die Potenzmenge einer Menge gebildet wird,
- für einfache Beispiele Element- bzw. Teilmengenbeziehungen entscheiden können,
- die Definitionen für Vereinigung, Durchschnitt, Differenz und Komplement von Mengen kennen,
- die grundlegenden Eigenschaften dieser Operationen kennen und beweisen können,
- diese Operationen anwenden können.

1.9.1 Teilmengen

Teilmenge

Definition 1.21 **a)** Eine Menge A ist *Teilmenge* einer Menge B, falls jedes Element von A auch Element von B ist. Wir schreiben: $A \subseteq B$. Formal lautet die Definition:

$$A \subseteq B :\Leftrightarrow x \in A \Rightarrow x \in B \tag{1.26}$$

Untermenge
Obermenge

Ein synonymer Begriff für Teilmenge ist *Untermenge*. Entsprechend nennt man, wenn $A \subseteq B$ gilt, B *Obermenge* von A.

Gleichheit von Mengen

b) Zwei Mengen A und B sind *gleich*, wenn jede Teilmenge der anderen ist:

$$A = B :\Leftrightarrow A \subseteq B \wedge B \subseteq A \tag{1.27}$$

Sind die Mengen A und B nicht gleich, gilt also $\neg(A = B)$, dann schreiben wir $A \neq B$.

Echte Teilmenge

c) Eine Menge A ist eine *echte Teilmenge* einer Menge B, falls A Teilmenge von B ist, aber nicht gleich B ist:

$$A \subset B :\Leftrightarrow A \subseteq B \wedge A \neq B \tag{1.28}$$

□

Beispiel 1.40 Es gilt:

a) $\{2,3,4,7\} \subseteq \{1,2,3,4,7,13\}$.

b) $\{1,2,3\} = \{3,2,1\}$ sowie $\{1,2,3\} \subseteq \{3,2,1\}$, aber $\{1,2,3,4\} \not\subseteq \{3,2,4,1\}$.

c) $\{2,3,4,7\} \subset \{1,2,3,4,7,13\}$. □

Folgerung 1.13 **a)** Für jede Menge A gilt $\emptyset \subseteq A$, d.h. die leere Menge ist Teilmenge jeder Menge.

b) Für jede Menge A gilt $A \subseteq A$, d.h. jede Menge ist Teilmenge von sich selbst.

c) Seien A, B und C Mengen, dann gilt: $A \subseteq B \wedge B \subseteq C \Rightarrow A \subseteq C$.

Beweis **a)** Nach Definition 1.21 ist zu zeigen: $x \in \emptyset \Rightarrow x \in A$, d.h. wir müssen zeigen, dass die Subjunktion $x \in \emptyset \rightarrow x \in A$ immer wahr ist.

Das Prädikat $x \in \emptyset$ ist immer falsch, und das Prädikat $x \in A$ kann wahr oder falsch sein. Folgende Wahrheitstafel zeigt die beiden möglichen Belegungen für die Subjunktion:

$x \in \emptyset$	$x \in A$	$x \in \emptyset \rightarrow x \in A$
0	1	1
0	0	1

$x \in \emptyset \rightarrow x \in A$ ist also immer wahr (es handelt sich um eine Tautologie der Art $0 \rightarrow \alpha$), d.h. es gilt $x \in \emptyset \Rightarrow x \in A$ und damit die Behauptung $\emptyset \subseteq A$.

Der Beweis für **b)** folgt analog (siehe folgende Übung).

c) Wir setzen $\alpha = (x \in A)$, $\beta = (x \in B)$ sowie $\gamma = (x \in C)$. Aus Beispiel 1.10 c) (Seite 25) wissen wir, dass $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \gamma) \Rightarrow (\alpha \Rightarrow \gamma)$ gilt (Kettenschluss). Damit können wir folgern

$$\begin{aligned} A \subseteq B \wedge B \subseteq C &\Rightarrow ((x \in A) \Rightarrow (x \in B)) \wedge ((x \in B) \Rightarrow (x \in C)) \\ &\Rightarrow (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \gamma) \\ &\Rightarrow (\alpha \Rightarrow \gamma) \\ &\Rightarrow ((x \in A) \Rightarrow (x \in C)) \\ &\Rightarrow A \subseteq C \end{aligned}$$

womit die Behauptung gezeigt ist. □

Übungsaufgaben

1.23 (1) Beweisen Sie Folgerung 1.13 b)!

(2) Seien K und M irgendwelche Mengen sowie $A = \{3,4,\{5,6\}\}$ und $B = \{5,6\}$.

(a) Setzen Sie eines oder mehrere der Symbole $\in, \notin, \subseteq, \not\subseteq$ richtig ein: (i) $M \ldots M$, (ii) $\emptyset \ldots \{\,0\,\}$, (iii) $K \ldots \{\,\{\,1,2\,\},\ K\,\}$, (iv) $\{\,1\,\} \ldots \{\,\{\,1,2\,\}, K\,\}$.

(b) Welche der folgenden Aussagen sind wahr: (i) $\{\,3,4\,\} \subseteq A$, (ii) $\emptyset \in A$, (iii) $\emptyset \subseteq A$, (iv) $B \subseteq A$, (v) $B \in A$, (vi) $6 \in A$, (g) $\{\,6\,\} \subseteq A$? □

1.9.2 Potenzmengen

Potenzmenge

Definition 1.22 Sei M eine Menge. Dann heißt $\mathcal{P}(M) = \{\,A \mid A \subseteq M\,\}$, die Menge aller Teilmengen von M, die *Potenzmenge* von M. Es gilt also: $A \in \mathcal{P}(M) :\Leftrightarrow A \subseteq M$. Anstelle von $\mathcal{P}(M)$ schreiben wir auch 2^M. □

Beispiel 1.41 Sei $M = \{\,a, b, c\,\}$, dann gilt

$$\mathcal{P}(M) = \{\,\emptyset, \{a\}, \{b\}, \{c\}, \{\,a, b\,\}, \{\,a, c\,\}, \{\,b, c\,\}, \{\,a, b, c\,\}\,\}$$

Aus Folgerung 1.13 a) und b) folgt unmittelbar

Folgerung 1.14 Für jede Menge M gilt $\emptyset \in \mathcal{P}(M)$ und $M \in \mathcal{P}(M)$. □

Satz 1.19 Sei M eine Menge mit m Elementen, $|M| = m$, dann hat $\mathcal{P}(M)$ 2^m Elemente: $|\mathcal{P}(M)| = 2^{|M|}$.

Beweis Es sei $M = \{x_1, \ldots, x_m\}$. Wir repräsentieren jede Teilmenge $A \subseteq M$ durch eine Folge $b^A = \langle\, b_1^A, \ldots, b_m^A \,\rangle$ von m Bits $b_i^A \in \{0, 1\}$, $1 \leq i \leq m$, mit $b_i^A = 1$, falls $x_i \in A$ und $b_i^A = 0$, falls $x_i \notin A$. Offensichtlich wird jede Teilmenge A durch genau eine Folge b^A repräsentiert, und jeder Bitfolge $b = \langle\, b_1, \ldots, b_m \,\rangle$ entspricht genau eine Teilmenge von A. Die Anzahl der Teilmengen von $A \subseteq M$ mit $|M| = m$ und die Anzahl der Bitfolgen der Länge m sind also gleich. Da es 2^m Bitfolgen der Länge m gibt, gibt es also 2^m Teilmengen, womit die Behauptung gezeigt ist. □

Mit $\mathcal{F}(M)$ bezeichnen wir die Menge der endlichen Teilmengen von M. Für endliche Mengen M gilt $\mathcal{F}(M) = \mathcal{P}(M)$, für unendliche Mengen M gilt $\mathcal{F}(M) \subset \mathcal{P}(M)$. Es gilt z.B. $\mathbb{G}_+ \in \mathcal{P}(\mathbb{Z})$, aber $\mathbb{G}_+ \notin \mathcal{F}(\mathbb{Z})$.

Übungsaufgaben

1.24 (1) Bestimmen Sie $\mathcal{P}(\emptyset)$ sowie $\mathcal{P}(\mathcal{P}(\emptyset))$!

(2) Bestimmen Sie die einzige Menge M, für die $M \subseteq \mathcal{P}(M)$ gilt! □

1.9.3 Verknüpfung von Mengen

Definition 1.23 Es seien A und B zwei Mengen.

a) Die Menge $A \cup B = \{ x \mid x \in A \vee x \in B \}$, welche alle Elemente von A und B enthält, heißt *Vereinigung* von A und B. **Vereinigung**

b) Die Menge $A \cap B = \{ x \mid x \in A \wedge x \in B \}$, welche alle gemeinsamen Elemente von A und B enthält, heißt *Durchschnitt* (auch *Schnittmenge*) von A und B. **Durchschnitt, Schnittmenge**

c) Gilt $A \cap B = \emptyset$, dann heißen A und B *disjunkt* (auch *elementfremd*). **Disjunktheit**

d) Die Menge $A - B = \{ x \mid x \in A \wedge x \notin B \}$, welche alle Elemente von A enthält, die nicht Element von B sind, heißt *Differenz* von A und B. **Differenz**

e) Die Menge $A \ominus B = (A - B) \cup (B - A)$, welche alle Elemente von A enthält, die nicht Element von B sind, und alle Elemente von B enthält, die nicht Element von A sind, heißt *symmetrische Differenz* von A und B. **Symmetrische Differenz**

f) Falls $A \subseteq B$ ist, dann heißt $\mathcal{C}_B A = B - A$ das *Komplement* von A bezüglich B. Falls die Menge B aus dem Zusammenhang heraus klar ist, schreibt man anstelle von $\mathcal{C}_B A$ auch $\overline{A}$. □ **Komplement**

Beispiel 1.42 Es sei $A = \{ 1, 2, 3, 4 \}$ und $B = \{ 3, 4, 5 \}$. Dann gilt:

a) $A \cup B = \{ 1, 2, 3, 4, 5 \}$

b) $A \cap B = \{ 3, 4 \}$

c) $A - B = \{ 1, 2 \}$

d) $A \ominus B = \{ 1, 2, 5 \}$

e) $\mathcal{C}_{\mathbb{N}_0} \mathbb{G} = \mathbb{U}$ □

Für die Vereinigung bzw. für den Durchschnitt von n Mengen A_1, A_2, ..., A_n, $n \geq 0$, führen wir noch folgende Schreibweisen ein:

$$A_1 \cup A_2 \cup \ldots \cup A_n = \bigcup_{i=1}^{n} A_i$$

$$A_1 \cap A_2 \cap \ldots \cap A_n = \bigcap_{i=1}^{n} A_i$$

Diese Schreibweise kann man noch verallgemeinern für den Fall, dass die Indizes nicht die Zahlen 1, 2, ..., n sind, sondern Elemente einer – möglicherweise unendlichen – Indexmenge I:

$$\bigcup_{i \in I} A_i \text{ bzw. } \bigcap_{i \in I} A_i$$

Betrachten wir hierzu ein Beispiel: Sei

$$tag = \{ mo, di, mi, do, fr, sa \}$$

eine Indexmenge, und sei K_t die Menge der Kunden, die am Tag t kaufen, dann bezeichnet

$$\bigcup_{t \in tag} K_t$$

die Menge der Kunden an allen Tagen, und

$$\bigcap_{t \in tag} K_t$$

bezeichnet die Menge der Kunden, die jeden Tag gekauft haben.

1.9.4 Elementare Eigenschaften

Im Folgenden werden die elementaren Eigenschaften der oben eingeführten Mengenoperationen aufgelistet. Eine Reihe dieser Eigenschaften gelten, weil entsprechende Eigenschaften für die die Operationen definierenden logischen Veknüpfungen gelten (vergleiche Satz 1.5, Seite 31).

Satz 1.20 Für alle Mengen A, B und C gelten die folgenden Gesetze:

(1) Die Operationen Vereinigung, Durchschnitt und symmetrische Differenz sind kommutativ:

$$A \cup B = B \cup A$$
$$A \cap B = B \cap A$$
$$A \ominus B = B \ominus A$$

(2) Falls $A \subseteq B$ ist, dann gilt:

$$A \cup B = B$$
$$A \cap B = A$$
$$A - B = \emptyset$$
$$A \ominus B = B - A$$

(3) Vereinigung und Durchschnitt sind idempotente Verknüpfungen:

$$A \cup A = A$$
$$A \cap A = A$$

(4) Aus den Eigenschaften (1) - (3) folgt:

$$A \cup \emptyset = A$$
$$A \cap \emptyset = \emptyset$$
$$A - A = \emptyset$$
$$\emptyset - A = \emptyset$$
$$A - \emptyset = A$$

(5) Für Vereinigung, Durchschnitt und Mengendifferenz gilt:

$$\begin{aligned} A &\subseteq A \cup B \\ B &\subseteq A \cup B \\ A \cap B &\subseteq A \\ A \cap B &\subseteq B \\ A - B &\subseteq A \end{aligned}$$

(6) Für die symmetrische Differenz gilt:

$$A \ominus B = (A \cup B) - (A \cap B)$$

(7) Vereinigung und Durchschnitt sind assoziative Operationen:

$$\begin{aligned} A \cup (B \cup C) &= (A \cup B) \cup C \\ A \cap (B \cap C) &= (A \cap B) \cap C \end{aligned}$$

(8) Vereinigung und Durchschnitt sind distributiv:

$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \\ (A \cap B) \cup C &= (A \cup C) \cap (B \cup C) \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \\ (A \cap B) \cup C &= (A \cup C) \cap (B \cup C) \end{aligned}$$

(9) Es gelten die De Morganschen Regeln:[14]

$$\begin{aligned} \overline{A \cup B} &= \overline{A} \cap \overline{B} \\ \overline{A \cap B} &= \overline{A} \cup \overline{B} \end{aligned}$$

(10) Es gelten die Absorptionsgesetze:

$$\begin{aligned} A \cup (B \cap A) &= A \\ A \cap (B \cup A) &= A \end{aligned}$$

(11) Doppelte Komplementbildung: $\overline{\overline{A}} = A$

Beweis Bei allen Eigenschaften – bis auf (5) – sind Gleichheiten von Mengen zu zeigen. Gemäß Definition 1.21 b) auf Seite 72 sind zwei Mengen gleich, wenn jede Teilmenge der anderen ist. Mit dieser Methode zeigen wir die erste Gleichheit von (9), alle anderen Beweise sind Gegenstand von Übung 1.25. Fast alle Gleichheiten lassen sich auf die Äquivalenz der Prädikate zurückführen, mithilfe derer die Mengenoperationen definiert sind (siehe Definition 1.23 auf Seite 75).

14 Benannt nach Augustus De Morgan (1806 - 1871), britischer Mathematiker und Logiker. De Morgan leistete Beiträge zur Algebra und zur mathematischen Logik und begründete neben George Boole eine Algebra der Logik.

Um $\overline{A \cup B} = \overline{A} \cap \overline{B}$ zu zeigen, müssen wir also

$$\overline{A \cup B} \subseteq \overline{A} \cap \overline{B} \text{ und } \overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$$

zeigen.

$\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$ gilt, falls die Implikation $x \in \overline{A \cup B} \Rightarrow x \in \overline{A} \cap \overline{B}$ gültig ist, was wir im Folgenden zeigen:

$x \in \overline{A \cup B}$	$\Rightarrow x \notin A \cup B$	(gemäß Definiton des Komplements)
	$\Rightarrow \neg(x \in A \cup B)$	(gemäß Definition von $\notin$)
	$\Rightarrow \neg(x \in A \vee x \in B)$	(gemäß Definition der Vereinigung)
	$\Rightarrow x \notin A \wedge x \notin B$	(gemäß De Morganscher Regeln für Konjunktion und Disjunktion, Satz 1.5, Seite 31)
	$\Rightarrow x \in \overline{A} \wedge x \in \overline{B}$	(gemäß Definiton des Komplements)
	$\Rightarrow x \in \overline{A} \cap \overline{B}$	(gemäß Definition des Durchschnitts)

Der Beweis von $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$ geschieht durch Umkehrung dieser Implikationen. □

Übungsaufgaben

1.25 (1) Beweisen Sie Satz 1.20!

(2) Zeigen Sie, dass die Mengendifferenz im Allgemeinen keine kommutative Verknüpfung ist!

(3) Zeigen Sie, dass die symmetrische Differenz eine kommutative Verknüpfung ist! □

Folgerung 1.15 **a)** Für zwei endliche Mengen A und B gilt:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

b) Sind A und B endlich und disjunkt, dann gilt $|A \cup B| = |A| + |B|$. □

Partition
Überdeckung
Zerlegung

Definition 1.24 Sei A eine nicht leere Menge, I eine Indexmenge und $A_i \subseteq A$, $i \in I$, eine Familie von nicht leeren Teilmengen von A. Dann heißt $\{A_i\}_{i \in I}$ eine *Partition* von A genau dann, wenn gilt:

(1) $A_i \cap A_j = \emptyset$ für $i, j \in I$ mit $i \neq j$,

(2) $\bigcup_{i \in I} A_i = A$.

Eine Partition *zerlegt* also eine Menge vollständig in disjunkte Teilmengen. Man sagt auch, dass die Menge A von $\{A_i\}_{i\in I}$ (vollständig und disjunkt) *überdeckt* wird.

Vergröberung, Verfeinerung einer Partition

Ist $J \subseteq I$ eine weitere Indexmenge und $\{B\}_{j\in J}$ eine weitere Partition von A. Dann heißt die Partition $\{A\}_{i\in I}$ *feiner* als die Partition $\{B\}_{j\in J}$ bzw. die Partition $\{B\}_{j\in J}$ heißt *gröber* als die Partition $\{A\}_{i\in I}$, falls zu jedem $i \in I$ ein $j \in J$ existiert mit $A_i \subseteq B_j$. □

Beispiel 1.43 **a)** Die Mengen $\mathbb{G}_+$ und $\mathbb{U}_+$ bilden eine Partition von $\mathbb{N}_0$.

b) Sei $\Sigma = \{a, b, \ldots, z\}$ die Menge der Kleinbuchstaben des deutschen Alphabets. Seien $W_a, W_b, \ldots, W_z$ die Menge der Wörter der deutschen Sprache, die mit a, mit b usw. oder mit z beginnen. Dann bilden diese Mengen eine Partition der Menge D aller deutschen klein geschriebenen Wörter, denn es gilt: $\bigcup_{\alpha\in\Sigma} W_\alpha = D$ und $W_\alpha \cap W_\beta = \emptyset$ für $\alpha \neq \beta$, $\alpha,\beta \in \Sigma$.

c) Die Mengen

$$
\begin{aligned}
[0]_3 &= \{0, 3, 6, \ldots\} = \{z | z = 3k,\ k \in \mathbb{N}_0\} \\
[1]_3 &= \{1, 4, 7, \ldots\} = \{z | z = 3k + 1,\ k \in \mathbb{N}_0\} \\
[2]_3 &= \{0, 5, 7, \ldots\} = \{z | z = 3k + 2,\ k \in \mathbb{N}_0\}
\end{aligned}
$$

bilden eine Partition von $\mathbb{N}_0$; wir bezeichnen diese mit $\mathbb{N}_0/3$, d.h. es ist

$$\mathbb{N}_0/3 = \{[0]_3, [1]_3, [2]_3\}$$

Entsprechend bilden die Mengen

$$[i]_6 = \{z | z = 6k + i,\ k \in \mathbb{N}_0\}$$

für $0 \leq i \leq 5$ ebenfalls eine Partition von $\mathbb{N}_0$:

$$\mathbb{N}_0/6 = \{[0]_6, [1]_6, [2]_6, [3]_6, [4]_6, [5]_6\}$$

Die Partition $\mathbb{N}_0/6$ ist feiner als die Partition $\mathbb{N}_0/3$, denn es gilt:

$$
\begin{aligned}
[0]_6 &\subseteq [0]_3 \\
[1]_6 &\subseteq [1]_3 \\
[2]_6 &\subseteq [2]_3 \\
[3]_6 &\subseteq [0]_3 \\
[4]_6 &\subseteq [1]_3 \\
[5]_6 &\subseteq [2]_3
\end{aligned}
$$

(siehe auch Beispiel 2.11 auf Seite 100). □

1.9.5 Zusammenfassung

Grundlegende Verknüpfungen von Mengen, die sowohl für theoretische Untersuchungen als auch für praktische Anwendungen von Bedeutung sind, umfassen die Teilmengen- und Potenzmengenbildung, die Vereinigung, den Durchschnitt, die Differenz und die symmetrische Differenz sowie die Komplementbildung. Diese Beziehungen und Verknüpfungen werden mithilfe von logischen Operatoren definiert. Deshalb besitzen die Mengenverknüpfungen dieselben Eigenschaften wie die entsprechenden logischen Verknüpfungen.

1.10 Boolesche Algebra

Wenn wir Satz 1.5 (Seite 31) und Satz 1.20 (Seite 76) vergleichen, bemerken wir, dass sowohl für Verknüpfungen der Logik als auch für Mengenverknüpfungen gleiche Eigenschaften gelten. Wir wollen von den konkreten Operanden, aussagen- oder prädikatenlogische Formeln bzw. Mengen abstrahieren und Verknüpfungsstrukturen betrachten, die Elemente einer Menge mit drei Operatoren verknüpfen und dabei bestimmten Rechenregeln genügen. Wir lernen damit eine erste abstrakte Rechenstruktur kennen, von der wir bereits zwei Beispiele, nämlich aussagenlogische Formeln und Mengen, betrachtet haben.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- den Begriff der Booleschen Algebra und dessen grundlegende Eigenschaften kennen,
- den Begriff der Isomorphie Boolescher Algebren erklären können,
- wissen, dass die Boolesche Algebra der Wahrheitswerte ein Repräsentant für alle minimalen Booleschen Algebren ist,
- wissen, dass die Booleschen Algebren der Potenzmengen von endlichen Mengen Repräsentanten für endliche Boolesche Algebren sind.

1.10.1 Definitionen und grundlegende Eigenschaften

Boolesche Algebra
Nullelement
Einselement

Definition 1.25 Eine Boolesche Algebra[15] $\mathcal{B} = (B, 0, 1, \oplus, \otimes, ')$ ist gegeben durch eine Menge B mit zwei ausgezeichneten Elementen 0 (sogenanntes *Nullelement*) und 1 (sogenanntes *Einselement*) aus B sowie den zweistelligen Opera-

15 Der britische Mathematiker und Logiker George Boole (1815 - 1864) gilt als Begründer der mathematischen Logik. Durch Formalisierung des mathematischen Denkens („An investigation of the laws of thought“ ist eine berühmte Schrift von Boole zu diesem Thema) entwickelte er eine Algebra der Logik, d.h. eine Logik, mit der man „rechnen“ kann (siehe Kapitel 1.2 über Aussagenlogik).

toren $\oplus$ und $\otimes$, die angewendet auf zwei Elemente aus B wieder ein Element aus B ergeben, und den einstelligen Operator $'$, der angewendet auf ein Element aus B wieder ein Element aus B liefert. Die Operatoren und beliebigen Elemente $x, y, z \in B$ müssen dabei folgenden Gesetzen genügen:

(i) Kommutativität für $\oplus$ und $\otimes$:

$$x \oplus y = y \oplus x$$
$$x \otimes y = y \otimes x$$

(ii) Assoziativität für $\oplus$ und $\otimes$:

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z$$
$$x \otimes (y \otimes z) = (x \otimes y) \otimes z$$

(iii) Absorption:

$$x \oplus (y \otimes x) = x$$
$$x \otimes (y \oplus x) = x$$

(iv) Distributivität:

$$x \oplus (y \otimes z) = (x \oplus y) \otimes (x \oplus z)$$
$$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$$

(v) Eigenschaften von 0 und 1:

$$x \oplus 0 = x \qquad x \otimes 0 = 0$$
$$x \oplus 1 = 1 \qquad x \otimes 1 = x$$

(vi) Eigenschaften von $'$:

$$x \oplus x' = 1 \qquad x \otimes x' = 0$$

□

Die nächste Folgerung gibt zwei Boolesche Algebren an, die wir schon kennen.

Folgerung 1.16 **a)** Sei M irgendeine Menge, dann bildet

$$\mathcal{P}_M = (\mathcal{P}(M), \emptyset, M, \cup, \cap, \mathcal{C}_M)$$

eine Boolesche Algebra.

b) $\mathbb{B} = (\{\, 0, 1 \,\}, 0, 1, \vee, \wedge, \neg)$ mit

$$1 \vee 1 = 1 \quad 1 \vee 0 = 1 \quad 0 \vee 1 = 1 \quad 0 \vee 0 = 0 \quad \neg 1 = 0$$
$$1 \wedge 1 = 1 \quad 1 \wedge 0 = 0 \quad 0 \wedge 1 = 0 \quad 0 \wedge 0 = 0 \quad \neg 0 = 1$$

bildet ebenfalls eine Boolesche Algebra. □

$\mathcal{P}_M$ ist die Boolesche Algebra aller Teilmengen einer gegebenen Menge M. Die Operation $\oplus$ ist die Vereinigung, die Operation $\otimes$ ist der Durchschnitt und die Operation $'$ ist das Komplement bezüglich M. Das Nullelement ist die leere Menge und das Einselement ist die Menge M selbst. Die Mengenverknüpfungen genügen den Bedingungen (i) - (vi) der Definition 1.25 für eine Boolesche Algebra (siehe Satz 1.20, Seite 76).

$\mathbb{B}$ ist die Boolesche Algebra der Wahrheitswerte in der Aussagenlogik. Die Operation $\oplus$ ist die Disjunktion, die Operation $\otimes$ ist die Konjunktion und die Operation $'$ ist die Negation. Das Nullelement ist der Wahrheitswert 0 („falsch“) und das Einselement ist der Wahrheitswert 1 („wahr“). Die Junktoren genügen den Bedingungen (i) - (vi) der Definition 1.25 für eine Boolesche Algebra (siehe Satz 1.5, Seite 31).

Für Boolesche Algebren allgemein gelten natürlich alle Eigenschaften, die wir für die speziellen Booleschen Algebren $\mathbb{B}$ und $\mathcal{P}_M$ für eine Menge M bereits gezeigt haben (siehe Sätze 1.5 und 1.20). Ebenso gilt für jede Boolesche Algebra das Dualitätsprinzip, das wir in Satz 1.11 (Seite 41) für die Aussagenlogik bereits formuliert haben.

Dualitätsprinzip Boolescher Algebren

Satz 1.21 Ist eine Eigenschaft einer Booleschen Algebra $\mathcal{B}$ gültig, dann ist auch ihre duale Eigenschaft in $\mathcal{B}$ gültig. Die duale Eigenschaft erhält man, indem man jedes Vorkommen von $\oplus$ durch $\otimes$, jedes $\otimes$ durch $\oplus$, jedes Vorkommen von 0 durch 1 und jede 1 durch 0 ersetzt. □

Über die im Satz 1.5 bzw. im Satz 1.20 formulierten Eigenschaften hinaus gelten für alle Booleschen Algebren noch die im folgenden Satz aufgeführten fundamentalen Eigenschaften.

Satz 1.22 Sei $\mathcal{B} = (B, 0, 1, \oplus, \otimes, ')$ eine Boolesche Algebra. Dann sind durch die definierenden Eigenschaften in Definition 1.25

a) das Null- und das Einselement sowie

b) a' für jedes $a \in B$ eindeutig bestimmt.

Beweis **a)** Wir nehmen an, es gäbe außer 0 noch das davon verschiedene Nullelement z. z muss die Eigenschaft (v) aus Definition 1.25 erfüllen. Es gilt also $a \otimes z = z$ für alle $a \in B$, d.h. diese Gleichung gilt auch für $a = 0$:

$$0 \otimes z = z \tag{1.29}$$

Andererseits gilt $a \otimes 0 = 0$ für alle $a \in B$, also auch für $a = z$, d.h.

$$z \otimes 0 = 0 \tag{1.30}$$

Aus den Gleichungen (1.29) und (1.30) und mit der Kommutativität von $\otimes$ folgt

$$z = 0 \otimes z = z \otimes 0 = 0$$

und damit $z = 0$.

Der Beweis der Eindeutigkeit des Einselements erfolgt analog durch Dualisierung des obigen Beweises der Eindeutigkeit des Nullelementes.

b) Wir nehmen an, es gäbe zu $x \in B$ außer x' noch ein Element c_x mit den Eigenschaften (vi) aus Definition 1.25. Es gilt also:

$$x \oplus c_x = 1 \tag{1.31}$$
$$x \otimes c_x = 0 \tag{1.32}$$

Es folgt:

$$\begin{aligned}
x' &= x' \oplus 0 && \text{wegen Eigenschaft (v) von Definition 1.25}\\
&= x' \oplus (x \otimes c_x) && \text{wegen (1.32)}\\
&= (x' \oplus x) \otimes (x' \oplus c_x) && \text{wegen Eigenschaft (iv) von Definition 1.25}\\
&= 1 \otimes (x' \oplus c_x) && \text{wegen Eigenschaft (vi) von Definition 1.25}\\
&= x' \oplus c_x && \text{wegen Eigenschaft (v) von Definition 1.25}
\end{aligned}$$

Es gilt also

$$x' = x' \oplus c_x \tag{1.33}$$

Dual zur Herleitung dieser Gleichung lässt sich mit

$$\begin{aligned}
x' &= x' \otimes 1 && \text{wegen Eigenschaft (v) von Definition 1.25}\\
&= x' \otimes (x \oplus c_x) && \text{wegen (1.31)}\\
&= (x' \otimes x) \oplus (x' \otimes c_x) && \text{wegen Eigenschaft (iv) von Definition 1.25}\\
&= 0 \oplus (x' \otimes c_x) && \text{wegen Eigenschaft (vi) von Definition 1.25}\\
&= x' \otimes c_x && \text{wegen Eigenschaft (v) von Definition 1.25}
\end{aligned}$$

die Gleichung

$$x' = x' \otimes c_x \tag{1.34}$$

herleiten. Mit diesen Ergebnissen folgt nun:

$$\begin{aligned}
x' &= x' \otimes c_x && \text{wegen (1.34)}\\
&= (x' \oplus c_x) \otimes c_x && \text{wegen (1.33)}\\
&= c_x \otimes (x' \oplus c_x) && \text{wegen Eigenschaft (i) von Definition 1.25}\\
&= c_x && \text{wegen Eigenschaft (iii) von Definition 1.25}
\end{aligned}$$

Damit haben wir ausgerechnet, dass $x' = c_x$ gilt. □

1.10.2 Isomorphie Boolescher Algebren

$\mathbb{B}$ ist im Übrigen im folgenden Sinn die kleinste Boolesche Algebra: Jede Boolesche Algebra mit zwei Elementen ist strukturgleich zu $\mathbb{B}$. Das bedeutet, dass es

zum einen keine Boolesche Algebra mit nur einem Element geben kann, und zum anderen gibt es für jede andere Boolesche Algebra $\mathcal{B}' = (\{a, b\}, a, b, \oplus, \otimes, ')$ mit zwei Elementen eine eineindeutige Zuordnung $\varphi : \{ a, b \} \to \{ 0, 1 \}$, so dass für alle $x, y \in \{ a, b \}$ gilt

$$\begin{aligned} \varphi(x \oplus y) &= \varphi(x) \vee \varphi(y) \\ \varphi(x \otimes y) &= \varphi(x) \wedge \varphi(y) \\ \varphi(x') &= \neg\varphi(x) \end{aligned}$$

Isomorphismus

Bis auf die Umbenennung φ der Elemente von $\mathcal{B}'$ und der entsprechenden Umbenennung der Operatoren sind also alle zweielementigen Booleschen Algebren identisch zu $\mathbb{B}$. Diese Art der Strukturgleichheit nennt man *Isomorphismus*.

So wie die Boolesche Algebra der Wahrheitswerte $\mathbb{B}$ ein „Prototyp“ für alle minimalen Booleschen Algebren ist, ist die Boolesche Algebra

$$\mathcal{P}_n = (\mathcal{P}(\mathbb{N}_{1,n}), \cup, \cap, \mathcal{C}_{\mathbb{N}_{1,n}})$$

der Teilmengen der Menge $\mathbb{N}_{1,n} = \{ 1, \ldots, n \}$, $n \geq 1$, ein Protoyp für alle endlichen Booleschen Algebren. Dies besagt der folgenden Satz, den wir ohne Beweis angeben.

Satz 1.23 Zu jeder endlichen Booleschen Algebra $\mathcal{X} = (X, 0, 1, \oplus, \otimes, ')$ gibt es eine Zahl $n \in \mathbb{N}$, so dass $\mathcal{X}$ und $\mathcal{P}_n$ isomorph zueinander sind, d.h. es gibt eine eineindeutige Zuordnung („Umbennung“) $\varphi : \mathcal{P}(\mathbb{N}_{1,n}) \to X$, so dass für alle $A, B \in \mathcal{P}(\mathbb{N}_{1,n})$ gilt

$$\begin{aligned} \varphi(A \cup B) &= \varphi(A) \oplus \varphi(B) \\ \varphi(A \cap B) &= \varphi(A) \otimes \varphi(B) \\ \varphi(\mathcal{C}_{\mathbb{N}_{1,n}} A) &= (\varphi(A))' \end{aligned}$$

□

Folgerung 1.17 **a)** Die Anzahl der Elemente einer endlichen Booleschen Algebra ist immer 2^n für ein $n \geq 1$.

b) Je zwei endliche, isomorphe Boolesche Algebren haben in jedem Fall dieselbe Anzahl von Elementen. □

Übungsaufgaben

1.26 Geben Sie Beispiele für zwei „kleinste“ Boolesche Algebren an! □

1.10.3 Zusammenfassung

Da die Verknüpfungen von Mengen mithilfe von logischen Prädikaten definiert werden, ist es nicht verwunderlich, dass logische Verknüpfungen von Aussagen und Verknüpfungen von Teilmengen einer Menge analoge Eigenschaften erfüllen, wie z.B. Kommutativität, Assoziativität und Distributivität sowie die Existenz von ausgezeichneten Elementen wie die Menge selbst und die leere Menge bzw. die logische 1 und die logische 0. Es stellt sich heraus, dass die Rechenstruktur der Potenzmenge einer Menge mit den Mengenverknüpfungen Vereinigung, Durchschnitt und Komplement quasi dieselbe ist wie aussagenlogischen Formeln, welche mit der Booleschen Basis gebildet werden. Eine Abstraktion für solche Rechenstrukturen stellt die Boolesche Algebra dar. Die Potenzmenge einer n-elementigen Menge ist z.B. ein Prototyp für eine endliche Boolesche Algebra. Daraus folgt, dass endliche Boolesche Algebren immer genau 2^n Elemente besitzen. Die kleinste Algebra besitzt somit zwei Elemente, und ein Repräsentant dafür ist die Aussagenlogik.

2 Relationen und Funktionen

Relationen und Funktionen als spezielle Relationen spielen in der Informatik eine sehr wichtige Rolle. Relationen setzen Werte aus verschiedenen Mengen in Beziehung. Betrachten wir zur Einführung eine kleine relationale Datenbank. Diese soll Daten über Studierende und über Professorinnen und Professoren enthalten. Für die Datenbank sollen bei den Studierenden die Merkmale Matrikelnummer, Name, Wohnort und Studienfach, und für die Professorinnen und Professoren sollen die Merkmale Name, Fachbereich, Raum und Telefonnummer gespeichert werden. Außerdem soll festgehalten werden, welche Studierende bei welchen Professorinnen oder Professoren in welchem Semester welche Vorlesung mit welchen Prüfungsergebnissen gehört haben. Diese Anwendung können wir durch drei Beziehungen modellieren:

1. Die Relation *Studis* setzt für jeden Studierenden die entsprechenden Werte der oben genannten Merkmale in Beziehung. Wir können diese wie folgt als Tabelle darstellen:

Studis	Matr	Name	Wohnort	Studienfach
	123456	Schmitz	Bonn	Informatik
	790123	Müller	Berlin	Mathematik
	456789	Meier	Dresden	Mathematik
	⋮	⋮	⋮	⋮

2. Die Relation *Profs* setzt für jede Professorin und jeden Professor die entsprechenden Werte ihrer oben genannten Merkmale in Beziehung. Folgende Tabelle zeigt mögliche Daten:

Profs	Name	Fachbereich	Raum	Telefon
	Einstein	Physik	C 123	4567
	Noether	Mathematik	A 890	1234
	Codd	Informatik	B 567	8901
	⋮	⋮	⋮	⋮

3. Die Relation *hört bei* setzt Studierende und Professorinnen oder Professoren mit entsprechenden Werten der oben dafür genannten Merkmale in Beziehung. Folgende Tabelle zeigt mögliche Daten:

hört bei	Studi	Prof	Semester	Vorlesung	Ergebnis
	123456	Codd	WS 00/01	Datenbanken	2,0
	790123	Noether	SS 00	Verbandstheorie	1,0
	456789	Einstein	WS00/01	Math. Physik I	3,0
	⋮	⋮	⋮	⋮	⋮

Den Merkmalen sind Wertebereiche zugeordnet, und eine Relation setzt Kombinationen von Werten den Merkmalen entsprechender Wertebereiche in Beziehung. Im Beispiel ist dem Merkmal *Matr* als Wertebereich die Menge sechsstelliger Ziffernfolgen zugeordnet, *Name*, *Wohnort* und *Studienfach* sind jeweils Zeichenketten als Werte zugeordnet. Die Relation *Studis* setzt Kombinationen von

K.-U. Witt, *Mathematische Grundlagen für die Informatik*,
DOI 10.1007/978-3-658-03079-7_2, © Springer Fachmedien Wiesbaden 2013

Werten dieser Wertebereiche in Beziehung, jede Zeile der obigen Tabelle *Studis* ist eine solche Kombination. Tabellen sind eine Möglichkeit zur Darstellung von Relationen. Für die beiden anderen Tabellen bzw. Relationen gilt Entsprechendes. Zur Tabelle *hört bei* sei bemerkt, dass Werte des Merkmals *Studi* Werte sind, die als Werte des Merkmals *Matr* in der Tabelle *Studis* vorkommen und dass Werte des Merkmals *Prof* Werte sind, die als Werte des Merkmals *Name* in der Tabelle *Profs* vorkommen. Bedingung dafür ist, dass *Matr* ein identifizierendes Merkmal für *Studis* und *Name* ein identifizierendes Merkmal für *Profs* ist. *Matr* und *Name* heißen auch Schlüsselmerkmale für *Studis* bzw. für *Profs*. *Studi* und *Prof* heißen Fremdschlüssel in *hört bei*.

Der Entwurf, die Implementierung und die Anwendung relationaler Datenbanksysteme basieren auf soliden mathematischen Grundlagen. Relationale Datenbanksystem-Technologien sind heutzutage die weltweit am verbreitesten eingesetzten Technologien zur Speicherung und Verwaltung von Daten.[16]

Funktionen sind eindeutige Relationen in dem Sinne, dass ein Element einer Menge mit höchstens einem Element einer anderen Menge in Beziehung stehen darf. Funktionale Beziehungen treten bei vielen Phänomenen in Natur- und Ingenieurwissenschaften auf, sie sind ein wichtiges Hilfsmittel zu deren Modellierung und Analyse.

2.1 Relationen

Relationen sind ein wichtiges Hilfsmittel um Beziehungen zwischen Elementen von Mengen auszudrücken. So kann man, wie oben in der Einleitung angedeutet, den Entwurf und die Implementierung von Datenbanken mithilfe von Relationen realisieren. Weitere Beispiele für die Verwendung des Relationsbegriffs in der Informatik sind: das Ableiten von Formelmengen mithilfe von Kalkülen (siehe Kapitel 1.3.3 und 1.5), die Ableitung von Wörtern mithilfe von Grammatiken, die Konfigurationsübergänge von Automaten beim Abarbeiten von Wörten, die Ausführung von Programmen auf (abstrakten) Maschinen. In diesem Kapitel werden grundlegende Begriffe und Eigenschaften für Relationen vorgestellt.

Lernziele

Nach dem Durcharbeiten des Kapitels sollten Sie

- die Begriffe kartesisches Produkt und Relation kennen sowie wie Eigenschaften von Relationen, insbesondere Reflexivität, Symmetrie, Antisymmetrie und Transitivität erklären können,
- den Begriff der Ordnung kennen und nachweisen können, ob eine Relation eine Ordnung ist,

16 In realen Anwendungen enthalten relationale Datenbanken durchaus Hunderte oder gar Tausende von Relationen (Tabellen).

- erklären können, was totale und was dichte Ordnung bedeutet,
- den Begriff der Äquivalenzrelation kennen und nachweisen können, ob eine Relation eine Äquivalenzrelation ist, und gegebenenfalls die Äquivalenzklassen der Relation bestimmen können,
- wissen, was der Begriff Partitionierung bedeutet und dessen Zusammenhang zu Äquivalenzrelationen verstehen,
- die Begriffe Umkehrrelation und Komposition von Relationen kennen,
- wissen, was die reflexiv-transitive bzw. die transitive Hülle einer Relation ist.

2.1.1 Kartesisches Produkt

Zunächst definieren wir als weitere Mengenverknüpfung das kartesische Produkt.

Definition 2.1 Für n Mengen $A_1, \ldots, A_n$, $n \geq 0$, heißt die Menge

$$A_1 \times \ldots \times A_n = \{\, (x_1, \ldots, x_n) \mid x_i \in A_i,\ 1 \leq i \leq n \,\}$$

n-stelliges kartesisches Produkt von $A_1, \ldots, A_n$. Anstelle von $A_1 \times \ldots \times A_n$ schreiben wir auch $\times_{i=1}^{n} A_i$. Falls $A_i = \emptyset$ für mindestens ein i, $1 \leq i \leq n$, ist, dann ist $\times_{i=1}^{n} A_i = \emptyset$.

Kartesisches Produkt

$(x_1, \ldots, x_n)$ heißt n-*Tupel*, für $n = 2$ sprechen wir von *Paaren*, für $n = 3$ von *Tripeln* und für $n = 4$ oder $n = 5$ auch von *Quadrupeln* bzw. von *Quintupeln*.

x_i, $1 \leq i \leq n$, heißt die *i-te Komponente* von $(x_1, \ldots, x_n)$. □

n-Tupel
Paar, Tripel
Quadrupel
Quintupel

Tupel entsprechen den Zeilen in den Tabellen in der obigen Einleitung, welche die Relationen unserer „Hochschuldatenbank“ darstellen.

i-te Komponente

Beispiel 2.1 Für die Mengen $A = \{\, 1, 2 \,\}$, $B = \{\, a, b, c \,\}$ und $C = \{\, 2, 3 \,\}$ ist

$$\begin{aligned} A \times B &= \{\, (1,a), (1,b), (1,c), (2,a), (2,b), (2,c) \,\} \\ A \times B \times C &= \{\, (1,a,2), (1,a,3), (1,b,2), (1,b,3), (1,c,2), (1,c,3), \\ &\qquad (2,a,2), (2,a,3), (2,b,2), (2,b,3), (2,c,2), (2,c,3) \,\} \end{aligned}$$

Folgerung 2.1 **a)** Ist $|A_i| < \infty$, $1 \leq i \leq n$, $n \geq 0$, dann gilt

$$|A_1 \times \ldots \times A_n| = |A_1| \cdot \ldots \cdot |A_n|$$

b) Sind alle A_i identisch, d.h. $A_i = A_{i+1}$, $1 \leq i \leq n-1$, $n \geq 1$, dann heißt

$$A_1 \times \ldots \times A_n = \underbrace{A \times \ldots \times A}_{n-\text{mal}}$$

n-faches kartesisches Produkt

n-faches kartesisches Produkt von A. Abkürzend benutzen wir dafür auch die Potenzschreibweise: A^n. Es gilt dann $A^1 = A$. Das n-fache kartesische Produkt von A können wir auch wie folgt rekursiv definieren:

$$A^1 = A$$
$$A^{n+1} = A^n \times A,\ n \geq 1$$

Ist $|A| < \infty$, dann ist $|A^n| = |A|^n$.[17] □

2.1.2 Relationen: Definitionen und Eigenschaften

Ein n-stelliges kartesisches Produkt setzt alle Elemente der zugrunde liegenden Mengen miteinander in Beziehung. Sollen nur bestimmte Elemente der zugrunde liegenden Mengen in Beziehung gesetzt werden, wie etwa bei der „Hochschuldatenbank“ in der Einleitung, dann sprechen wir von Relationen.

n-stellige Relation

Homogene, heterogene Relation

Grundmenge

Definition 2.2 Jede Teilmenge $R \subseteq A_1 \times \ldots \times A_n$ heißt *n-stellige Relation* über $A_1, \ldots, A_n$. Sind alle Mengen A_i identisch, dann heißt R *homogen*, sonst *heterogen*. Bei einer n-stelligen homogenen Relation $R \subseteq A \times \ldots \times A$ heißt A auch die *Grundmenge* von R. □

Beispiel 2.2 Es sei $A = \{ -3, -2, -1, 0, 1, 2, 3 \}$.

a) Für die Relation $R_1 \subseteq A \times A$ definiert durch

$$R_1 = \{ (x, y) \in A \times A \mid x \cdot y > 2 \}$$

gilt

$$\begin{aligned} R_1 = \{ &(-3,-3), (-3,-2), (-3,-1), \\ &(-2,-3), (-2,-2), \\ &(-1,-3), \\ &(1,3), \\ &(2,2), (2,3), \\ &(3,1), (3,2), (3,3) \} \end{aligned}$$

b) Für die Relation $R_2 \subseteq A \times A$ definiert durch

$$R_2 = \{ (x, y, z) \in A^3 \mid x + y = z \}$$

17 $\mathbb{N}_u^k$ bedeutet gemäß den Vereinbarungen von Kapitel 1.1.3 über die Notation von Zahlenmemgen die Menge der natürlichen Zahlen von u bis k. Nach der hier getroffenen Vereinbarung kann damit auch das k-fache kartesiche Produkt der Menge $\mathbb{N}_u$ der natürlichen Zahlen größer gleich u gemeint sein. Im Folgenden wird jeweils aus dem Zusammenhang klar, welche dieser beiden Bedeutungen für $\mathbb{N}_u^k$ gemeint ist.

gilt

$$\begin{aligned} R_2 = \{\, &(-3,0,-3),(-3,1,-2),(-3,2,-1),(-3.3,0),\\ &(-2,-1,-3),\ldots,(-2,3,1),\\ &(-1,-2,-3),\ldots,(-1,3,2),\\ &(0,-3,-3),\ldots,(0,3,3),\\ &(1,-3,-2),\ldots,(1,2,3),\\ &(2,-3,-1),\ldots,(2,1,3),\\ &(3,-3,0),\ldots,(3,0,3)\,\} \end{aligned}$$

R_1 und R_2 sind homogene Relationen. □

Endliche zweistellige Relationen $R \subseteq A \times B$ lassen sich auch als Boolesche Matrizen darstellen: Die Zeilen werden mit den Elementen aus A gekennzeichnet, die Spalten mit den Elementen aus B. Ist $A = \{a_1, \ldots, a_m\}$ und $B = \{b_1, \ldots, b_n\}$, dann tragen wir am Kreuzungspunkt der Zeile i und der Spalte j genau dann eine 1 ein, falls $(a_i, b_j) \in R$ ist, ansonsten tragen wir dort 0 ein.

Die Relation R_1 aus dem obigen Beispiel kann also durch folgende Matrix dargestellt werden:

$x \backslash y$	-3	-2	-1	0	1	2	3
-3	1	1	1	0	0	0	0
-2	1	1	0	0	0	0	0
-1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1
2	0	0	0	0	0	1	1
3	0	0	0	0	1	1	1

Übungsaufgaben

2.1 Sei $A = B = \{\, x \in \mathbb{Z} \mid -3 \leq x \leq 3 \,\}$. Geben Sie die folgenden Relationen über $A \times B$ in aufzählender Form an:

(1) $R_1 = \{\, (x,y) \mid y < 2x + 2 \,\}$

(2) $R_2 = \{\, (x,y) \mid y < x + 1 \,\}$

(3) $R_3 = \{\, (x,y) \mid 2x + y > 1 \,\} \cup \{\, (x,y) \mid x = y \,\}$

(4) $R_4 = \{\, (x,y) \mid 2x + y > 1 \,\} \cap \{\, (x,y) \mid x = y \,\}$

(5) $R_5 = \{\, (x,y) \mid 3 - x^2 \leq y \,\} \cap \{\, (x,y) \mid x^2 \geq y + 3 \,\}$ □

Ausgangsmenge **Zielmenge** **Definitionsbereich** **Wertebereich**

Bei einer zweistelligen Relation $R \subseteq A \times B$ nennen wir A die *Ausgangsmenge* und B die *Zielmenge* von R. Wir nennen die Menge $Def(R) = \{x \in A \mid \exists y \in B,\ (x,y) \in R\}$ der Elemente der Ausgangsmenge von R, die als erste Komponente in Elementen von R vorkommen, den *Definitionsbereich* von R. Analog nennen wir die Menge $W(R) = \{y \in B \mid \exists x \in A,\ (x,y) \in R\}$ der Elemente der Zielmenge von R, die als zweite Komponente von Elementen in R vorkommen, den *Wertebereich* von R.

Beispiel 2.3 Für die Relation R_1 aus Beispiel 2.2 gilt $Def(R_1) = W(R_1) = \{-3,-2,-1,1,2,3\} = A - \{0\}$. □

Wir wollen die Tatsache, dass ein Paar $(x,y) \in A \times B$ zu R gehört („in der Beziehung R steht“) ausdrücken, indem wir die normale *Elementschreibweise* $(x,y) \in R$, die *Präfixschreibweise* $R(x,y)$ oder die *Infixschreibweise* xRy verwenden.

Nullrelation **Vollständige Relation** **Identische Relation**

Ist $R = \emptyset$, dann heißt R *Nullrelation*. Ist $R = A \times B$, dann heißt R *vollständig*. Ist $R \subseteq A \times A$ mit $R = \{\,(x,x) \mid x \in A\,\}$, dann ist R die *identische Relation* über A, diese wird in der Regel mit id_A bezeichnet.

Im Folgenden betrachten wir Eigenschaften zweistelliger homogener Relationen über einer Grundmenge A.

Definition 2.3 Sei $R \subseteq A \times A$ eine zweistellige homogene Relation über der Grundmenge A. Dann heißt R

Reflexivität **a)** *reflexiv* genau dann, wenn xRx für alle $x \in A$ gilt. Bei einer reflexiven Relation muss also jedes Element der Grundmenge mit sich selber in Relation stehen.

Irreflexivität **b)** *irreflexiv* genau dann, wenn $(x,x) \notin R$ für alle $x \in A$ ist.

Symmetrie **c)** *symmetrisch* genau dann, wenn gilt: $xRy \Rightarrow yRx$. Eine Relation ist symmetrisch, wenn gilt: Steht x mit y in Relation, dann muss auch y mit x in der Relation stehen.

Asymmetrie **d)** *asymmetrisch* genau dann, wenn gilt: Ist xRy, dann $\neg yRx$.

Antisymmetrie **e)** *antisymmetrisch* genau dann, wenn gilt: $xRy \wedge yRx \Rightarrow x = y$. Eine Relation ist antisymmetrisch, wenn gilt: Steht x mit y in Relation und y mit x, dann muss notwendigerweise $x = y$ sein.

Transitivität **f)** *transitiv* genau dann, wenn gilt: $xRy \wedge yRz \Rightarrow xRz$. Eine Relation ist transitiv, wenn gilt: Stehen x mit y und y mit z in Relation, dann steht auch x mit z in Relation.

Injektivität **g)** *linkseindeutig* oder *injektiv* genau dann, wenn gilt: Ist x_1Ry_1, x_2Ry_2 und $x_1 \neq x_2$, dann muss $y_1 \neq y_2$ gelten.

Rechtseindeutige Relation **Funktion** **h)** *rechtseindeutig* genau dann, wenn gilt: Ist x_1Ry_1, x_2Ry_2 und $y_1 \neq y_2$, dann muss $x_1 \neq x_2$ gelten. Rechtseindeutige Relationen werden in der Regel *Funktionen* genannt. Funktionen werden im Allgemeinen mit den Buchstaben f, g und h bezeichnet, und anstelle $f \subseteq A \times A$ wird $f : A \to A$ geschrieben sowie anstelle von xfy wird $f(x) = y$ notiert.

i) *linkstotal* oder *total* genau dann, wenn gilt: Für alle $x \in A$ existiert ein $y \in A$ mit xRy.

Totale Relation

k) *rechtstotal* oder *surjektiv* genau dann, wenn gilt: Für alle $y \in A$ existiert ein $x \in A$ mit xRy.

Surjektivität

l) *bijektiv* genau dann, wenn R total, injektiv und surjektiv ist. Eine Bijektion ist eine eineindeutige Zuordnung zwischen den Elementen von Ausgangs- und Zielmenge.

Bijektivität

Die Definitionen g) – l) gelten analog auch für Relationen $R \subseteq A \times B$, d.h. für Relationen, bei denen Ausgangs- und Zielmenge verschieden sein können. □

Beispiel 2.4 Wir definieren die Relation $\leq \subseteq \mathbb{N}_0 \times \mathbb{N}_0$ durch

$$x \leq y \text{ genau dann, wenn es ein } c \in \mathbb{N}_0 \text{ gibt, so dass } x + c = y$$

Es gilt z.B. $3 \leq 5$, denn es gibt ein $c = 2$, so dass $3 + 2 = 5$.

$\leq$ ist eine reflexive Relation, denn für jedes $x \in \mathbb{N}_0$ gibt es $c = 0$, so dass $x + 0 = x$ ist, d.h. für alle $x \in \mathbb{N}_0$ gilt $x \leq x$.

$\leq$ ist nicht symmetrisch, denn es gilt z.B. $3 \leq 5$, aber nicht $5 \leq 3$.

$\leq$ ist antisymmetrisch: Sei $x \leq y$ und $y \leq x$, d.h. es gibt ein $c_x \in \mathbb{N}_0$ mit $x + c_x = y$, und es gibt ein $c_y \in \mathbb{N}_0$ mit $y + c_y = x$. Aus den beiden Gleichungen folgt $x + c_x + c_y = x$ und daraus $c_x + c_y = 0$. Da $c_x, c_y \in \mathbb{N}_0$, kann $c_x + c_y = 0$ nur gelten, wenn $c_x = c_y = 0$ gilt. Das bedeutet aber, dass $x = y$ ist.

Da die Relation reflexiv ist, ist sie nicht asymmetrisch.

$\leq$ ist transitiv: Sei $x \leq y$ und $y \leq z$. Wir müssen zeigen, dass dann auch $x \leq z$ gilt. Da $x \leq y$ ist, gibt es ein $c_x \in \mathbb{N}_0$ mit $x + c_x = y$, und da $y \leq z$ ist, gibt es ein $c_y \in \mathbb{N}_0$ mit $y + c_y = z$. Hieraus folgt unmittelbar, dass es ein $c_z = c_x + c_y$ gibt mit $x + c_z = z$, d.h. es gilt $x \leq z$.

Die Relation ist nicht injektiv, denn es gilt z.B. $3 \leq 5$ und $4 \leq 5$. Damit ist die Relation auch nicht bijektiv.

Die Relation ist nicht rechtseindeutig und damit keine Funktion, denn es gilt z.B. $3 \leq 4$ und $3 \leq 5$.

Da die Relation reflexiv ist, ist sie auch total und surjektiv. □

Übungsaufgaben

2.2 Beweisen Sie folgende Behauptungen:

(1) Eine homogene Relation R ist antisymmetrisch genau dann, wenn gilt: Ist xRy und $x \neq y$, dann gilt $\neg yRx$.

(2) Eine Relation R ist injektiv genau dann, wenn gilt: Ist $x_1 R y$ und $x_2 R y$, dann ist $x_1 = x_2$, d.h. verschiedene Elemente der Ausgangsmenge können nicht mit demselben Element der Zielmenge in Relation stehen.

(3) Eine Relation R ist rechtseindeutig genau dann, wenn gilt: Ist $x R y_1$ und $x R y_2$, dann ist $y_1 = y_2$. Ein Element der Ausgangsmenge kann höchstens mit einem Element der Zielmenge in Relation stehen.

(4) Eine Relation $R \subseteq A \times B$ ist genau dann total, wenn $Def(R) = A$ gilt.

(5) Eine Relation $R \subseteq A \times B$ ist genau dann surjektiv, wenn $W(R) = B$ gilt.

2.3 Sei D die Menge der Wörter der deutschen Sprache. Die Relation $\alpha \subseteq D \times D$ sei definiert durch: $x \,\alpha\, y$ genau dann, wenn x und y denselben Anfangsbuchstaben haben. Überlegen Sie, dass α eine reflexive, symmetrische und transitive Relation ist! □

2.1.3 Ordnungen

Partielle Ordnung

Definition 2.4 Eine Relation $R \subseteq A \times A$ heißt *partielle Ordnung* über A genau dann, wenn R reflexiv, antisymmetrisch und transitiv ist. Partielle Ordnungen werden auch einfach nur Ordnungen genannt. □

Beispiel 2.5 **a)** Die Relation $\leq$ aus dem Beispiel 2.4 ist eine Ordnung über $\mathbb{N}_0$.

b) Sei $M = \{ a, b, c \}$ und $S \subseteq \mathcal{P}(M) \times \mathcal{P}(M)$ definiert durch

$$xSy \text{ genau dann, wenn } x \subseteq y$$

Die Teilmenge $x \subseteq M$ steht also in Relation S zur Teilmenge $y \subseteq M$, falls x eine Teilmenge von y ist. S ist eine Ordnung über $\mathcal{P}(M)$, denn S ist reflexiv, antisymmetrisch und transitiv:

S ist reflexiv, denn jede Menge ist Teilmenge von sich selbst (siehe Folgerung 1.13 b) auf Seite 73), und damit gilt $x \subseteq x$ für jedes $x \in \mathcal{P}(M)$ und damit xSx für jedes $x \in \mathcal{P}(M)$.

S ist antisymmetrisch: Gilt xSy und ySx, d.h. $x \subseteq y$ und $y \subseteq x$, dann gilt (siehe Definition 1.21 b) auf Seite 72) $x = y$.

S ist transitiv: Sei xSy und ySz, dann muss auch xSz gelten. xSy bedeutet $x \subseteq y$ und ySz bedeutet $y \subseteq z$. Hieraus folgt (siehe Folgerung 1.13 c) auf Seite 73), dass $x \subseteq z$ und damit xSz. □

Geordnete Menge

Ist R eine partielle Ordnung über A, dann schreibt man dafür auch (A, R) und nennt A eine *geordnete Menge*. Für unsere bisherigen Beispiele 2.4 und 2.5 b) können wir also schreiben: $(\mathbb{N}_0, \leq)$ bzw. $(\mathcal{P}(\{ a, b, c \}), S)$ oder für Letzteres auch $(\mathcal{P}(\{ a, b, c \}), \subseteq)$

Definition 2.5 Sei (A, R) eine Ordnung und $x, y \in A$.

a) Gilt xRy oder yRx, dann heißen x und y *vergleichbar*. Gilt $\neg xRy$ und $\neg yRx$, dann heißen x und y *unvergleichbar*.

b) Sei $B \subseteq A$, $B \neq \emptyset$. $x \in B$ heißt *minimales Element* von B, falls xRy für alle $y \in B$ gilt. $x \in B$ heißt *maximales Element* von B, falls yRx für alle $y \in B$ gilt.

Minimales Element
Maximales Element

c) $K \subseteq A$, $K \neq \emptyset$, heißt *Kette* genau dann, wenn für alle $x, y \in K$ gilt, dass x und y vergleichbar sind.

Kette

d) (A, R) heißt *totale Ordnung* oder auch *lineare Ordnung* (A, R) genau dann, wenn A eine Kette bildet.

Totale Ordnung

e) Eine totale Ordnung (A, R) heißt *Wohlordnung* genau dann, wenn jede Teilmenge $K \in A$, $K \neq \emptyset$, ein minimales Element besitzt. □

Wohlordnung

Beispiel 2.6 **a)** Die Ordnung $(\mathbb{N}_0, \leq)$ ist total, denn für zwei natürliche Zahlen x und y gilt $x \leq y$ oder $y \leq x$. Diese Ordnung ist zudem eine Wohlordnung.

b) Wenn wir die Ordnung aus Beispiel 2.4 auf ganze Zahlen erweitern, dann bildet $(\mathbb{Z}, \leq)$ eine totale Ordnung, aber keine Wohlordnung, denn die Teilmenge der geraden Zahlen besitzt kein minimales Element. Wenn wir die ganzen Zahlen aber anders ordnen, z.B. in der Reihenfolge

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

dann liegt eine Wohlordnung vor. Um diese formal zu beschreiben, verwenden wird die Bijektion $\varphi : \mathbb{Z} \to \mathbb{N}_0$ definiert durch

$$\varphi(x) = \begin{cases} 2x, & \text{falls } x \geq 0 \\ -(2x+1), & \text{falls } x < 0 \end{cases} \tag{2.1}$$

und definierten die Relation $\leq_\varphi \subseteq \mathbb{Z} \times \mathbb{Z}$ damit wie folgt:

$$x \leq_\varphi y \text{ genau dann, wenn } \varphi(x) \leq \varphi(y)$$

Die Bijektion φ nimmt also eine eineindeutige Umbennenung der ganzen Zahlen durch natürliche Zahlen vor und überträgt quasi die Relation $\leq$ von den natürlichen Zahlen auf die ganzen Zahlen. Damit bildet $(\mathbb{Z}, \leq_\varphi)$ eine Wohlordnung.

c) Die Ordnung $(\mathcal{P}(\{a, b, c\}), \subseteq)$ ist nicht total, denn es gibt Teilmengen von $\{a, b, c\}$, die nicht in Teilmengen-Beziehung zueinander stehen, d.h. die bezüglich der Relation $\subseteq$ unvergleichbar sind. Es gilt z.B. weder $\{a, b\} \subseteq \{b, c\}$ noch $\{b, c\} \subseteq \{a, b\}$.

d) Die Ordnung $(\mathcal{P}(\{a, b, c\}), \subseteq)$ enthält unter anderen Ketten die Kette $\{\emptyset, \{a\}, \{a, b\}, \{a, b, c\}\}$, denn es gilt

$$\emptyset \subseteq \{a\} \subseteq \{a, b\} \subseteq \{a, b, c\}$$

Bestimmen Sie weitere Ketten in dieser Ordnung! □

Übungsaufgaben

2.4 (1) Überlegen Sie, für welche Mengen M die Ordnung $(\mathcal{P}(M), \subseteq)$ total ist!

(2) Beweisen Sie, dass die in (2.1) definierte Relation tatsächlich eine Bijektion ist! □

Da die $\leq$-Relation auf allen Zahlenmengen eine totale Ordnung festlegt, gilt sie als Prototyp für totale Ordnungen. Deshalb benutzt man das Symbol $\leq$ auch allgemein als Symbol für totale Ordnungen. Wird also $(A, \leq)$ für irgendeine Menge A notiert, soll dies bedeuten, dass eine total geordnete Menge A vorliegt. Wenn wir im Folgenden von Ordnungen oder geordneten Mengen sprechen, sind totale Ordnungen bzw. total geordnete Mengen gemeint.

Dichte Menge

Definition 2.6 Sei $(A, \leq)$ eine Ordnung. A heißt *dicht* bezüglich $\leq$ genau dann, wenn für alle $x, y \in A$ mit $x \neq y$ und $x \leq y$ ein $z \in A$ existiert mit $z \neq x$, $z \neq y$ und $x \leq z \leq y$. □

Eine geordnete Menge ist also dicht, falls zwischen zwei Elementen dieser Menge immer noch ein drittes liegt.

Beispiel 2.7 **a)** $(\mathbb{N}_0, \leq)$ und $(\mathbb{Z}, \leq)$ sind nicht dicht, denn zwischen zwei benachbarten natürlichen (ganzen) Zahlen x und $y = x + 1$ liegt keine weitere natürliche (ganze) Zahl.

b) Die Menge der rationalen Zahlen (Brüche) $(\mathbb{Q}, \leq)$ ist dicht. Betrachten wir z.B. $a, b \in \mathbb{Q}$ mit $a \leq b$ und $a \neq b$, dann ist auch

1. $\frac{a+b}{2} \in \mathbb{Q}$,
2. $a \neq \frac{a+b}{2}$ und $b \neq \frac{a+b}{2}$ und
3. $a = \frac{a+a}{2} \leq \frac{a+b}{2} \leq \frac{b+b}{2} = b$.

Zwischen zwei rationalen Zahlen a und b existiert also immer eine weitere rationale Zahl, z.B. die Zahl $\frac{a+b}{2}$. □

Übungsaufgaben

2.5 Überlegen Sie, dass in einer dichten Menge A zwischen zwei verschiedenen Elementen $a, b \in A$ unendlich viele von a und b verschiedene Elemente liegen! □

2.1.4 Äquivalenzrelationen

Äquivalenzrelation

Definition 2.7 Eine Relation $R \subseteq A \times A$ heißt *Äquivalenzrelation* über A genau dann, wenn R reflexiv, symmetrisch und transitiv ist. □

Beispiel 2.8 **a)** Die Relation α in Übung 2.3 ist eine Äquivalenzrelation.

b) Die Relation $\equiv_3 \subseteq \mathbb{Z} \times \mathbb{Z}$ sei definiert durch

$$x \equiv_3 y \text{ genau dann, wenn } \frac{x-y}{3} \in \mathbb{Z}$$

Die zwei ganzen Zahlen x und y stehen in der Relation $\equiv_3$ genau dann, wenn die Differenz $x - y$ durch 3 teilbar ist. Diese Relation ist eine Äquivalenzrelation, denn sie ist reflexiv, symmetrisch und transitiv:

$\equiv_3$ ist reflexiv: Für alle $x \in \mathbb{Z}$ gilt $\frac{x-x}{3} = 0 \in \mathbb{Z}$, also gilt $x \equiv_3 x$ für alle $x \in \mathbb{Z}$ und damit ist $\equiv_3$ reflexiv.

$\equiv_3$ ist symmetrisch, denn es gilt:

$$x \equiv_3 y \Rightarrow \frac{x-y}{3} \in \mathbb{Z} \Rightarrow (-1) \cdot \frac{x-y}{3} \in \mathbb{Z} \Rightarrow \frac{y-x}{3} \in \mathbb{Z} \Rightarrow y \equiv_3 x$$

$\equiv_3$ ist transitiv, denn es gilt:

$$\begin{aligned} x \equiv_3 y \wedge y \equiv_3 z \quad &\Rightarrow \quad \frac{x-y}{3} \in \mathbb{Z} \wedge \frac{y-z}{3} \in \mathbb{Z} \\ &\Rightarrow \quad \frac{x-y}{3} + \frac{y-z}{3} \in \mathbb{Z} \\ &\Rightarrow \quad \frac{x-z}{3} \in \mathbb{Z} \\ &\Rightarrow \quad x \equiv_3 z \end{aligned}$$

□

Äquivalenzklasse
Repräsentant
Index

Definition 2.8 Sei $R \subseteq A \times A$ eine Äquivalenzrelation und $x \in A$. Dann heißt die Menge

$$[x]_R = \{\, y \in A \mid xRy \,\}$$

aller Elemente von A, mit denen x in der Beziehung R steht, *Äquivalenzklasse* von R. x heißt *Repräsentant* der Äquivalenzklasse $[x]_R$. Die Anzahl der Äquivalenzklassen von R heißt der *Index* von R. □

Beispiel 2.9 **a)** Für die Äquivalenzrelation α aus Übung 2.3 gilt z.B.

$$[Boot]_\alpha = \{\, Boot, Buch, Ball, Badesalz, boxen, \ldots \}$$

$[Boot]_\alpha$ enthält Wörter, die mit B oder b beginnen: Alle diese Wörter haben denselben Anfangsbuchstaben (abgesehen von Groß- und Kleinschreibung). Offensichtlich kann jedes Element von $[Boot]_\alpha$ als Repräsentant dieser Äquivalenzklasse gewählt werden, denn es gilt z.B. $[Boot]_\alpha = [boxen]_\alpha$.

Die Äquivalenzklassen von α werden durch die Anfangsbuchstaben bestimmt. Es gibt somit 26 Äquivalenzklassen, der Index von α ist also 26.

b) Für die Äquivalenzrelation $\equiv_3$ aus Beispiel 2.8 b) gilt:

$$
\begin{aligned}
[0]_{\equiv_3} &= \{\, 0, 3, -3, 6, -6, \dots \} &&= \{\, x \mid x = 3y,\ y \in \mathbb{Z} \,\} \\
[1]_{\equiv_3} &= \{\, 1, -2, 4, -5, 7, -8, \dots \} &&= \{\, x \mid x = 3y + 1,\ y \in \mathbb{Z} \,\} \\
[2]_{\equiv_3} &= \{\, 2, -1, 5, -4, 8, -7, \dots \} &&= \{\, x \mid x = 3y + 2,\ y \in \mathbb{Z} \,\}
\end{aligned}
$$

$[0]_{\equiv_3}$ enthält die durch 3 teilbaren ganzen Zahlen, $[1]_{\equiv_3}$ enthält die ganzen Zahlen, die bei Division durch 3 den Rest 1 lassen, und $[2]_{\equiv_3}$ enthält die ganzen Zahlen, die bei Division durch 3 den Rest 2 lassen. Weitere Äquivalenzklassen gibt es nicht, denn bei Division durch 3 können nur drei Reste auftreten. Der Index von $\equiv_3$ ist also 3. Auch in diesem Beispiel sieht man sofort, dass jedes Element einer Äquivalenzklasse als ihr Repräsentant gewählt werden kann. Die

Restklasse

Äquivalenzklassen von $\equiv_3$ heißen auch *Restklassen modulo* 3 von $\mathbb{Z}$. □

Die Elemente einer Äquivalenzklasse sind äquivalent zueinander, d.h. sie sind durch die „Brille“ (den „Filter“) der Äquivalenzrelation betrachtet ununterscheidbar. Die Relation α projiziert alle Wörter der deutschen Sprachen nur auf den Anfangsbuchstaben. So gesehen, d.h. durch den ersten Buchstben bestimmt, gibt es 26 Arten von Wörtern. Die Relation $\equiv_3$ betrachtet für alle Zahlen nur die Reste, die bei Division durch 3 bleiben, alle Zahlen mit demselben Rest werden zusammengefasst. So betrachtet gibt es nur 3 (Arten von) Zahlen: $[0]_{\equiv_3}$, $[1]_{\equiv_3}$ und $[2]_{\equiv_3}$, die man jetzt der einfacheren Notation wegen auch wieder mit 0, 1 bzw. 2 bezeichnen könnte.[18]

Satz 2.1 Sei $R \subseteq A \times A$ mit $A \neq \emptyset$ eine Äquivalenzrelation. Dann gilt:

a) Für alle $x \in A$ ist $[x]_R \neq \emptyset$, d.h. Äquivalenzklassen sind niemals leer.

b) Für alle $y \in [x]_R$ gilt $[x]_R = [y]_R$, d.h. jedes Element einer Äquivalenzklasse kann als ihr Repräsentant gewählt werden. Äquivalenzklassen sind also unabhängig von ihrem Repräsentanten.

c) Falls $(x, y) \notin R$ ist, dann ist $[x]_R \cap [y]_R = \emptyset$, d.h. die Äquivalenzklassen nicht in Relation stehender Repräsentanten sind disjunkt.

d) Für $x, y \in A$ gilt entweder $[x]_R = [y]_R$ oder $[x]_R \cap [y]_R = \emptyset$, d.h. zwei Elemente aus der Grundmenge einer Äquivalenzrelation repräsentieren entweder dieselbe oder zwei disjunkte Äquivalenzklassen.

e) $A = \bigcup_{x \in A} [x]_R$, d.h. die Äquivalenzklassen bilden eine Überdeckung von A.

18 Mit diesen Zahlen kann man wie folgt rechnen:

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Addition und Multiplikation werden *modulo* 3 durchgeführt.

Beweis: **a)** Wegen der Reflexivität von R gilt xRx für alle $x \in A$ und damit $x \in [x]_R$ für alle $x \in A$.

b) Sei $z \in [x]_R$, d.h. es ist xRz. Nach Voraussetzung ist $y \in [x]_R$, also xRy, und damit, da R als Äquivalenzrelation symmetrisch ist, gilt auch yRx. Da R transitiv ist, folgt, da yRx und xRz gilt, dass auch yRz gilt, d.h. es gilt $z \in [y]_R$. Wir haben gezeigt, dass gilt: $z \in [x]_R \Rightarrow z \in [y]_R$ und damit $[x]_R \subseteq [y]_R$.

Sei $z \in [y]_R$, d.h. es ist yRz. Nach Voraussetzung ist $y \in [x]_R$, also xRy. Da R transitiv ist, folgt xRz, d.h. es gilt $z \in [x]_R$. Wir haben gezeigt, dass gilt: $z \in [y]_R \Rightarrow z \in [x]_R$ und damit $[y]_R \subseteq [x]_R$.

Insgesamt folgt die Behauptung $[x]_R = [y]_R$.

c) Wir nehmen an, dass $[x]_R \cap [y]_R \neq \emptyset$. Es gibt also mindestens ein $z \in [x]_R \cap [y]_R$, d.h. es ist $z \in [x]_R$ und $z \in [y]_R$. Mit b) folgt dann, dass $[x]_R = [z]_R$ bzw. $[z]_R = [y]_R$ ist. Hieraus folgt, dass $[x]_R = [y]_R$ ist, und damit gilt xRy. Dies ist ein Widerspruch zur Voraussetzung $(x, y) \notin R$, womit unsere Annahme falsch ist.

d) folgt unmittelbar aus b) und c).

e) Sei $z \in A$, dann ist $z \in [z]_R$ und damit $z \in \bigcup_{x \in A} [x]_R$. Also ist $A \subseteq \bigcup_{x \in A} [x]_R$.
Sei $z \in \bigcup_{x \in A} [x]_R$. Dann gibt es ein $x \in A$ mit $z \in [x]_R$. Da $[x]_R \subseteq A$ für jedes $x \in A$ ist, folgt, dass $z \in A$ ist. Also gilt $\bigcup_{x \in A} [x]_R \subseteq A$

Insgesamt haben wir gezeigt: $A = \bigcup_{x \in A} [x]_R$. □

Die Aussagen c) und e) des Satzes besagen, dass eine Äquivalenzrelation R über der Grundmenge A eine Partition (siehe Definition 1.24 auf Seite 78) dieser Menge in die Äquivalenzklassen von R induziert. Die nächste Folgerung besagt in ihren Aussagen a) und b), dass Äquivalenzrelationen und Partitionen äquivalente Zerlegungskonzepte sind.

Folgerung 2.2 **a)** Jede Äquivalenzrelation $R \subseteq A \times A$ legt eine Partition von A fest.

b) Jede Partition von A definiert eine Äquivalenzrelation auf A.

Feinste Partiton

c) Die *identische Relation* id_A legt die feinste Partition (siehe Definition 1.24) von A fest, denn jedes Element bildet genau eine Äquivalenzklasse: Es ist $[x]_R = \{ x \}$ für alle $x \in A$.

Gröbste Partiton

d) Die vollständige Relation $R = A \times A$ legt die gröbste Partition (siehe Definition 1.24) auf A fest, denn es gibt genau eine Äquivalenzklasse: Es ist $[x]_R = A$ für alle $x \in A$. □

Übungsaufgaben

2.6 Beweisen Sie Folgerung 2.2! □

Beispiel 2.10 **a)** Die Relation α aus Beispiel 1.43 b) auf Seite 79 partitioniert die Menge D der Wörter der deutschen Sprache in 26 disjunkte Wörtermengen nach dem ersten Buchstaben, wie es in der Regel in Wörterbüchern vorzufinden ist (siehe auch Beispiel 2.9 a) auf Seite 97).

b) Die Äquivalenzklassen $[0]_{\equiv_3}$, $[1]_{\equiv_3}$ und $[2]_{\equiv_3}$ der Relation $\equiv_3$ (siehe auch Beispiele 2.8 b) bzw. 2.9 b) auf Seite 97) sind disjunkt, und sie überdecken die Grundmenge $\mathbb{Z}$:

$$\mathbb{Z} = [0]_{\equiv_3} \cup [1]_{\equiv_3} \cup [2]_{\equiv_3}$$

□

Definition 2.9 Es seien R und S Äquivalenzrelationen über der Grundmenge A. Dann heißt R eine *Verfeinerung* von S und S heißt eine *Vergöberung* von R genau dann, wenn die Äquivalenzklassen von R eine Verfeinerung der Äquivalenzklassen von S sind. Wir notieren dies formal durch $R \leq S$. □

Verfeinerung
Vergröberung

Beispiel 2.11 Die Relation $\equiv_6 \subseteq \mathbb{Z} \times \mathbb{Z}$ sei definiert durch (siehe Beispiel 2.9 b)

$$x \equiv_6 y \text{ genau dann, wenn } \frac{x-y}{6} \in \mathbb{Z}$$

ist eine Äquivalenzrelation, die eine Verfeinerung der Relation $\equiv_3$ darstellt, d.h. es ist $\equiv_6 \leq \equiv_3$, denn es gelten die Beziehungen $[0]_6, [3]_6 \subseteq [0]_3$, $[1]_6, [4]_6 \subseteq [1]_3$ und $[2]_6, [5]_6 \subseteq [2]_3$ (siehe auch Beispiel 1.43 c) auf Seite 79). □

Übungsaufgaben

2.7 Für $m \in \mathbb{N}$ sei allgemein die Relation $\equiv_m \subseteq \mathbb{Z} \times \mathbb{Z}$ definiert durch

$$x \equiv_m y \text{ genau dann, wenn } \frac{x-y}{m} \in \mathbb{Z}$$

(1) Zeigen Sie, dass $\equiv_m$ eine Äquivalenzrelation ist, und geben Sie die Äquivalenzklassen und den Index von $\equiv_m$ an!

(2) Überlegen Sie, dass folgende Behauptung gilt: Es sei $m, n \in \mathbb{N}$, und m sei ein Teiler von n, dann gilt $\equiv_n \leq \equiv_m$.

2.8 Sei die Relation $suc \subseteq \mathbb{N}_0 \times \mathbb{N}_0$ definiert durch: $(x, y) \in suc$ genau dann, wenn $y = x + 1$. Ist suc eine Ordnung? Ist suc eine Äquivalenzrelation? Beweisen Sie Ihre Anworten!

2.9 Sei A eine endliche Menge.

(1) Sei $R = A \times A$ die vollständige Relation über A. Ist R eine Ordnung? Ist R eine Äquivalenzrelation? Beweisen Sie Ihre Aussagen! Falls R eine Äquivalenzrelation ist, dann geben Sie ihre Äquivalenzklassen an!

(2) Ist die identische Relation id_A eine Ordnung? Ist id_A eine Äquivalenzrelation? Beweisen Sie Ihre Aussagen! Falls id_A eine Äquivalenzrelation ist, dann geben Sie ihre Äquivalenzklassen an! □

2.1.5 Umkehrrelationen

Umkehrrelation

Definition 2.10 Seien A und B zwei Mengen. Für die Relation $R \subseteq A \times B$ heißt die Relation $R^{-1} \subseteq B \times A$ definiert durch

$$yR^{-1}x \text{ genau dann, wenn } xRy$$

die *Umkehrrelation* zu R. □

Folgerung 2.3 **a)** Die Umkehrrelation R^{-1} zu einer Relation $R \subseteq A \times B$ enthält genau die „umgedrehten" Paare von R:

$$R^{-1} = \{(y, x) \in B \times A \,|\, (x, y) \in R\}$$

b) $R \subseteq A \times B$ ist linkseindeutig genau dann, wenn R^{-1} rechtseindeutig ist.

c) $R \subseteq A \times B$ ist bijektiv genau dann, wenn R^{-1} bijektiv ist.

d) Ist R eine Äquivalenzrelation über der Grundmenge A, dann ist auch R^{-1} eine Äquivalenzrelation über A.

e) Ist $R \subseteq A \times A$ eine Äquivalenzrelation, dann gilt $R = R^{-1}$. □

Übungsaufgaben

2.10 Beweisen Sie diese Folgerungen! □

2.1.6 Komposition von Relationen

Komposition
Produkt

Definition 2.11 Seien A, B und C Mengen sowie $R \subseteq A \times B$ und $S \subseteq B \times C$ Relationen. Dann heißt die Relation $R \circ S \subseteq A \times C$ definiert durch

$$R \circ S = \{\, (x, z) \in A \times C \mid \exists y \in B : xRy \wedge ySz \,\}$$

die *Komposition* oder auch das *Produkt* von R und S. □

Beispiel 2.12 Es seien die Relationen $R_1, R_2 \subseteq \mathbb{N} \times \mathbb{N}$ definiert durch xR_1y genau dann, wenn $y = 2x$, sowie xR_2y genau dann, wenn $y = 3x$, gegeben. Es gilt also

$$R_1 = \{\, (1, 2), (2, 4), (3, 6), \ldots \}$$
$$R_2 = \{\, (1, 3), (2, 6), (3, 9), \ldots \}$$

Die Komposition von R_1 und R_2 ergibt

$$R_1 \circ R_2 = \{\,(1,6),(2,12),(3,18),\dots\,\}$$

d.h. $xR_1 \circ R_2 y$ genau dann, wenn $y = 6x$. Es gilt nämlich $aR_1 \circ R_2 b$ genau dann, wenn es ein c gibt mit $aR_1 c$ und $cR_2 b$, d.h. wenn es ein c gibt mit $c = 2a$ bzw. mit $b = 3c$, und hieraus folgt, dass $b = 6a$ ist. □

Der folgende Satz nennt wesentliche Eigenschaften der identischen Relation und begründet die Berechtigung des Begriffes „Umkehrrelation".

Satz 2.2 Seien A und B Mengen und $R \subseteq A \times B$. Dann gilt: **a)** $id_A \circ R = R$, **b)** $R \circ id_B = R$, **c)** $id_A \circ R \circ id_B = R$, **d)** ist R total, dann ist $id_A \subseteq R \circ R^{-1}$ und **e)** ist R surjektiv, dann ist $id_B \subseteq R^{-1} \circ R$. □

Der folgende Satz besagt, wie die Umkehrung einer Kompositon von Relationen berechnet wird.

Satz 2.3 Seien A, B und C Mengen sowie $R \subseteq A \times B$ und $S \subseteq B \times C$. Dann gilt:

$$(R \circ S)^{-1} = S^{-1} \circ R^{-1}$$

Die Komposition von Relationen ist eine assoziative Verknüpfung, und sie verhält sich distributiv zur Vereinigung und zum Durchschnitt.

Satz 2.4 Seien A, B, C und D Mengen.

a) Für die Relationen $R \subseteq A \times B$, $S \subseteq B \times C$ und $T \subseteq C \times D$ gilt:

$$R \circ (S \circ T) = (R \circ S) \circ T$$

b) Für die Relationen $R \subseteq A \times B$ und $S, T \subseteq B \times C$ gilt:

$$R \circ (S \cup T) = (R \circ S) \cup (R \circ T) \qquad R \circ (S \cap T) = (R \circ S) \cap (R \circ T)$$

Analog gelten die Rechtsdistributivitäten. □

Die Komposition ist verträglich mit der Teilmengenbeziehung.

Satz 2.5 Für die Relationen $R_1, R_2, S_1, S_2 \subseteq A \times B$ mit $R_1 \subseteq S_1$ und $R_2 \subseteq S_2$ gilt $R_1 \circ S_1 \subseteq R_2 \circ S_2$. □

Übungsaufgaben

2.11 (1) Beweisen Sie die Sätze 2.2, 2.3, 2.4 und 2.5!

(2) Zeigen Sie, dass im Allgemeinen die Komposition von Relationen $R, S \subseteq A \times A$ nicht kommutativ ist, also im Allgemeinen $R \circ S \neq S \circ R$ gilt! □

Der folgende Satz zeigt, dass die in Definition 2.3 eingeführten Begriffe auch mithilfe der identischen Relation sowie mit Kompositionen von Relationen und Umkehrrelationen beschrieben werden können.

Satz 2.6 Sei $R \subseteq A \times A$, dann gilt

a) R ist reflexiv genau dann, wenn $id_A \subseteq R$,

b) R ist irreflexiv genau dann, wenn $id_A \cap R = \emptyset$,

c) R ist symmetrisch genau dann, wenn $R = R^{-1}$,

d) R ist asymmetrisch genau dann, wenn $R \cap R^{-1} = \emptyset$,

e) R ist antisymmetrisch genau dann, wenn $R \cap R^{-1} \subseteq id_A$,

f) R ist transitiv genau dann, wenn $R^2 \subseteq R$,

g) R ist injektiv genau dann, wenn $R \circ R^{-1} \subseteq id_A$,

h) R ist rechtseindeutig genau dann, wenn $R^{-1} \circ R \subseteq id_A$,

i) R ist total genau dann, wenn $id_A \subseteq R \circ R^{-1}$,

k) R ist surjektiv genau dann, wenn $id_A \subseteq R^{-1} \circ R$,

l) R ist bijektiv genau dann, wenn $R \circ R^{-1} \subseteq R^{-1} \circ R$. □

2.1.7 Reflexiv-transitive Hüllen

Wir betrachten jetzt die fortgesetzte Kompositionen einer Relation $R \subseteq A \times A$ mit sich selbst.

Definition 2.12 Sei A eine Menge und $R \subseteq A \times A$ eine zweistellige Relation über A. Für R setzen wir fest:

(i) $R^0 = id_A = \{(x, x) \mid x \in A\}$

(ii) $R^{n+1} = R^n \circ R$, $n \geq 0$

(iii) $R^+ = R^1 \cup R^2 \cup \ldots = \bigcup_{i \geq 1} R^i$

(iv) $R^* = R^+ \cup R^0 = \bigcup_{i \geq 0} R^i$

R^+ heißt die *transitive Hülle* von R, und R^* ist die *reflexiv-transitive Hülle* von R. □

Transitive Hülle

Reflexiv-transitive Hülle

Beispiel 2.13 **a)** Es sei M die Menge der Menschen, die bisher auf der Erde gelebt haben, und die Relation $K \subseteq M \times M$ sei definiert durch: xKy gilt genau dann, wenn x ein Kind von y ist. Dann enthält K^2 alle Enkel-Beziehungen, K^3 alle Urenkel-Beziehungen usw. K^+ enthält alle Nachkommen-Beziehungen über alle Generationen hinweg.

b) Sei $R \subseteq \mathbb{N} \times \mathbb{N}$ definiert durch: xRy genau dann, wenn $y = 2x$. Es gilt

(i) $R^0 = id_{\mathbb{N}} = \{\,(1,1),(2,2),(3,3),\ldots\,\} = \{\,(x,x) \mid x \in \mathbb{N}\,\}$

(ii) $R^1 = R^0 \circ R = id_{\mathbb{N}} \circ R = \{\,(1,2),(2,4),(3,6),\ldots\,\} = \{\,(x,y) \mid y = 2x\,\}$

(iii) Fortgesetzte Komposition von R:

$$\begin{array}{lllllll} R^2 & = & R \circ R & = & \{\,(1,4),(2,8),(3,12),\dots\,\} & = & \{\,(x,y) \mid y = 4x\,\} \\ R^3 & = & R^2 \circ R & = & \{\,(1,8),(2,16),(3,24),\dots\,\} & = & \{\,(x,y) \mid y = 8x\,\} \\ R^4 & = & R^3 \circ R & = & \{\,(1,16),(2,32),(3,48),\dots\,\} & = & \{\,(x,y) \mid y = 16x\,\} \\ & \vdots & & \vdots & & \vdots & \end{array}$$

Für $n \in \mathbb{N}_0$ ergibt sich:

$$R^n = R^{n-1} \circ R = \{\,(1, 2^n \cdot 1), (2, 2^n \cdot 2), (3, 2^n \cdot 3), \dots\,\} = \{\,(x,y) \mid y = 2^n x\,\}$$

(iv) $R^* = \{\,(x,y) \mid y = 2^n x,\ n \in \mathbb{N}_0\,\}$

(v) $R^+ = \{\,(x,y) \mid y = 2^n x,\ n \in \mathbb{N}\,\}$ □

Übungsaufgaben

2.12 Sei $R \subseteq A \times A$ eine Relation. Zeigen Sie, dass $R^+ = R^* \circ R = R \circ R^*$ gilt! □

2.1.8 Zusammenfassung

Relationen setzen Elemente von Mengen in Beziehung. Wichtige Eigenschaften von Relationen sind Refexivität, Symmetrie, Antisymmetrie und Transitivität. Mithilfe von reflexiven, antisymmetrischen und transitiven Relationen werden Mengen geordnet. Geordnete Mengen M werden im Allgemeinen mit $(M, \leq)$ bezeichnet. Gilt für $a, b \in M$ $a \leq b$ oder $b \leq a$, dann heißt diese Ordnung total. Gibt es zu $a, b \in M$ mit $a \leq b$ und $a \neq b$ ein $c \in M$ mit $a \neq c$, $b \neq c$ und $a \leq c \leq b$, dann heißt die Ordnung dicht.

Relationen, die reflexiv, symmetrisch und transitiv sind, heißen Äquivalenzrelationen. Eine Äquivalenzrelation partitioniert die Grundmenge vollständig in disjunkte Äquivalenzklassen. Die Anzahl der Klassen heißt Index der Relation. Die Identität auf einer Grundmenge A induziert die feinste Partition von A, das kartesische Produkt $A \times A$ legt die gröbste Partition von A fest. Durch die „Brille“ einer Äquivalenzrelation betrachtet sind die Elemente einer Äquivalenzklasse ununterscheidbar – jedes Element einer Klasse kann als ihr Repräsentant gewählt werden.

Werden alle durch eine Relation R (reflexiv und) transitiv in Beziehung stehenden Elemente zu einer Relation zusammengefasst, entsteht die so genannte (reflexiv-) transitive Hülle (R^*) R^+ von R.

2.2 Funktionen

In diesem Kapitel betrachten wir Funktionen, also rechtseindeutige Relationen $R \subseteq A \times B$, bei denen einem Element aus der Ausgangsmenge A höchstens ein Element aus der Zielmenge B zugeordnet sein darf (siehe Definition 2.3 h). Mithilfe von injektiven und bijektiven Abbildungen werden wir Begriffe für den Größenvergleich von Mengen einführen und ein paar Eigenschaften dieser Vergleichsrelation betrachten.

Lernziele

Nach dem Durcharbeiten des Kapitels sollten Sie

- feststellen können, ob eine Relation eine Funktion ist,
- Definitions- und Wertebereiche von Funktionen bestimmen können,
- feststellen können, ob eine Funktion total, injektiv, surjektiv, bijektiv ist,
- den Begriff Prädikat erklären können,
- den Begriff der Mächtigkeit von Mengen erklären können,
- erläutern können, warum es eine unendliche Hierarchie von Mengenmächtigkeiten gibt.

2.2.1 Begriffe und Eigenschaften

Beispiel 2.14 **a)** Die Relation $R \subseteq \mathbb{N} \times \mathbb{N}$ definiert durch xRy genau dann, wenn $\frac{y}{x} \in \mathbb{N}$, ist keine Funktion. Es gilt z.B. $2R4$ und $2R6$, R ist also nicht rechtseindeutig.

b) Die Relation $sqr \subseteq \mathbb{N} \times \mathbb{N}$ definiert durch $x\ sqr\ y$ genau dann, wenn $y = x^2$, ist rechtseindeutig. Die Umkehrrelation sqr^{-1} ist ebenfalls rechtseindeutig.

c) Die Relation $sqr \subseteq \mathbb{Z} \times \mathbb{Z}$ definiert durch $x\ sqr\ y$ genau dann, wenn $y = x^2$, ist rechtseindeutig. Die Umkehrrelation sqr^{-1} ist in diesem Fall nicht rechtseindeutig, denn es gilt z.B. $4\ sqr^{-1}\ 2$ und $4\ sqr^{-1}\ -2$, aber $2 \neq -2$. □

An diesen Beispielen können wir sehen, dass die Rechtseindeutigkeit einer Relation nicht alleine von der Vorschrift abhängt, die festlegt, welche Objekte in Beziehung stehen, sondern auch von der Wahl der Ausgangs- oder der Zielmenge. Aus diesem Grund führen wir den Begriff der Abbildung ein, der außer der Funktionsvorschrift die Wahl der Mengen mit einbezieht.

Abbildung

Definition 2.13 Seien A und B Mengen. Eine *Abbildung* $\phi = (A, B, f)$ besteht aus den beiden Mengen A und B sowie aus der Funktion $f : A \to B$. □

Beispiel 2.15 Durch diese Festlegung ist nun geklärt, dass $\phi_1 = (\mathbb{N}, \mathbb{N}, sqrt)$ mit $y = sqrt(x) = \sqrt{x}$ eine Abbildung ist, da $\sqrt{}$ auf $\mathbb{N}$ rechtseindeutig ist, während $\phi_2 = (\mathbb{Z}, \mathbb{Z}, sqrt)$ mit $x\ sqrt\ y$ genau dann, wenn $y^2 = x$, keine Abbildung ist. □

Nachdem wir den Unterschied zwischen den Begriffen *Abbildung* und *Funktion* geklärt haben, werden wir im Folgenden beide Begriffe – wie durchaus in der Literatur üblich – synonym verwenden.

Übungsaufgaben

2.13 Welche der Relationen in Übung 2.1 sind Funktionen? □

Definition 2.14 Sei $f : A \to B$ eine Funktion sowie $x \in Def(f)$, $y \in W(f)$, $C \subseteq A$ und $D \subseteq B$, dann heißt

Argumente, Bilder, Werte einer Funktion

a) $y = f(x)$ das *Bild* von x unter f; die Elemente $x \in Def(f)$ heißen auch *Argumente* von f, und die Elemente $y \in W(f)$ heißen auch *Werte* oder *Bilder* von f.

b) $f(C) = \{f(x) \mid x \in C\} = \bigcup_{x \in C} \{f(x)\}$ das *Bild* von C unter f.

Urbildmenge

c) $f^{-1}(y) = \{x \in A \mid f(x) = y\}$ die *Urbildmenge* von y unter f.

d) $f^{-1}(D) = \bigcup_{y \in D} f^{-1}(y)$ die *Urbildmenge* von D unter f.

e) Ist $z \in A - Def(f)$, dann schreiben wir auch $f(z) = \perp$ und sagen, dass f *undefiniert* auf z ist. Ist $E \subseteq A - Def(f)$, dann schreiben wir $f(E) = \emptyset$, denn die Bildmenge der Menge der Elemente, für die f undefiniert ist, ist leer.

f) Mit $B^A = \{f : A \to B \mid f \text{ total}\}$ bezeichnen wir die Menge aller totalen Funktionen von A nach B. □

Folgerung 2.4 Sei $f : A \to B$ eine Funktion. Dann gilt:

a) $f^{-1}(y) = \emptyset$ für alle $y \in B - W(f)$.

Injektive Funktion

b) f ist injektiv genau dann, wenn $|f^{-1}(y)| \leq 1$ für alle $y \in B$ ist, d.h. wenn jedes Element der Zielmenge höchstens ein Urbild hat.

c) f^{-1} ist eine Funktion, d.h. eine rechtseindeutige Relation, genau dann, wenn f injektiv, d.h. linkseindeutig, ist.

Surjektive Funktion

d) f ist surjektiv genau dann, wenn $|f^{-1}(y)| \geq 1$ für alle $y \in B$ ist, d.h. wenn jedes Element der Zielmenge mindestens ein Urbild hat.

Bijektive Funktion

e) f ist bijektiv genau dann, wenn f total ist und $|f^{-1}(y)| = 1$ für alle $y \in B$ ist, d.h. wenn es eine eineindeutige Zuordnung zwischen allen Elementen von A und allen Elementen von B gibt.

f) f ist genau dann bijektiv, wenn f^{-1} bijektiv ist.

g) Sind A und B endliche Mengen, dann gilt $|A| = |B|$ genau dann, wenn eine Bijektion zwischen A und B existiert. □

Übungsaufgaben

2.14 Beweisen Sie die Folgerung 2.4!

2.15 Aus den drei Eigenschaften surjektiv, injektiv und total kann man sechs Kombinationen bilden. Geben Sie Funktionen an, die jeweils genau eine solche Kombination von Eigenschaften erfüllt!

2.16 Geben Sie für die Funktion

$$f : \mathbb{R} \to \mathbb{R} \text{ definiert durch } f(x) = \frac{1}{x}$$

$D(f)$ und $W(f)$ an! Ist f total, surjektiv, injektiv? Geben Sie f^{-1} an! Ist f^{-1} eine Funktion?

2.17 Geben Sie für die Funktion

$$f : \mathbb{R} \to \mathbb{R} \text{ definiert durch } f(x) = \frac{1}{x^2}$$

$D(f)$ und $W(f)$ an! Ist f total, surjektiv, injektiv? Geben Sie f^{-1} an! Ist f^{-1} eine Funktion? □

Komposition von Funktionen

Wie Relationen können Funktionen komponiert werden. Allerdings ist es üblich, nicht wie bei Relationen die zu komponierenden Relationen von links nach rechts aufzuführen, sondern die zu komponierenden Funktion werden von rechts nach links aufgeführt. Sind also $f : A \to B$ und $g : B \to C$ Funktionen, dann schreiben wir $g \circ f : A \to C$ und nicht $f \circ g$. Der Grund dafür ist, dass $(g \circ f)(x) = g(f(x))$ ist; eine Auswertung der Komposition von f und g also „von innen nach außen“ stattfindet: Zuerst wird f auf x angewendet und dann g auf das Ergebnis $f(x)$.

Beispiel 2.16 Sei die Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch $f(x) = 2x + 1$ und die Funktion $g : \mathbb{N}_0 \to \mathbb{N}_0$ durch $g(x) = x^2$, dann gilt

$$(g \circ f)(x) = g(f(x)) = g(2x + 1) = (2x + 1)^2 = 4x^2 + 4x + 1$$

bzw.

$$(f \circ g)(x) = f(g(x)) = f(x^2) = 2x^2 + 1$$

Im Allgemeinen ist die Komposition von Funktionen also keine kommutative Verknüpfung (was wir in Übung 2.11 (2) schon für Relationen gezeigt haben). □

Für die n-fache Komposition einer Funktion f mit sich selbst schreiben wir wie bei Relationen f^n, und es gilt auch hier $f^0 = id$.

2.2.2 Operationen und Prädikate

n-stellige Operation

n-stellige Funktionen, sind Funktionen, bei denen die Ausgangsmenge ein n-stelliges kartesisches Produkt ist. Funktionen dieser Art heißen auch *n-stellige Operationen*, *Operatoren* oder *Verknüpfungen*. Sie haben die Gestalt

$$f : A_1 \times \ldots \times A_n \to B,\ n \geq 0$$

Operand

In $f(x_1, \ldots, x_n) = y$ heißen die $x_i \in A_i$, $1 \leq i \leq n$, *Operanden* von f. Alle bisher definierten Begriffe sowie Eigenschaften von Funktionen werden analog auf n-stellige Operatoren übertragen.

Konstante Funktion

0-stellige Funktionen entsprechen Konstanten. Betrachten wir z.B. die 0-stellige Funktion $f :\to \mathbb{N}$ definiert durch $f() = 17$, dann können wir diese Funktion mit der Zahl 17 identifizieren, d.h. wir hätten f auch 17 nennen können: $17 :\to \mathbb{N}$ definiert durch $17() = 17$.

Bei Rechenstrukturen, wie z.B. bei Booleschen Algebren (siehe Definition 1.25 auf Seite 80) treten überwiegend ein- oder zweistellige Operationen auf, bei denen zudem alle Ausgangsmengen und die Zielmenge identisch sind, d.h. es handelt sich um Funktionen der Art $f : A \to A$ bzw. $f : A \times A \to A$. Beispiele sind arithmetische Operationen wie $+$, $-$, $\cdot$ usw., die Zahlen miteinander verknüpfen und deren Ergebnis wieder eine Zahl ist, oder aussagenlogische Junktoren wie $\neg$, $\vee$ und $\wedge$, die Wahrheitswerte zu Wahrheitswerten verknüpfen, oder die Mengenverknüpfungen $\cup$, $\cap$ und $\mathcal{C}$, die Teilmengen einer Menge zu Teilmengen dieser Menge verknüpfen.

Bei den Notationen ein- oder zweistelliger Verknüpfungen der Art $f : A \to A$ bzw. der Art $f : A \times A \to A$ unterscheidet man generell drei Schreibweisen:

Infixschreibweise

1. *Infixschreibweise:* Der Operator steht zwischen den Operanden: $xfy = z$, z.B. $x + y = z$ bei arithmetischen Verknüpfungen von Zahlen.

Präfixschreibweise

2. *Präfixschreibweise:* Der Operator steht vor dem Argument: $fx = y$, z.B. beim Logarithmus $\log x = y$.

Postfixschreibweise

3. *Postfixschreibweise:* Der Operator steht hinter dem Argument: $xf = y$, z.B. $x! = 1 \cdot 2 \cdot \ldots \cdot x$ (sprich: „x Fakultät“), für $x \in \mathbb{N}$.

Darüber hinaus gibt es noch Schreibweisen, die nicht unmittelbar einer dieser drei aufgelisteten zugeordnet werden können, wie z.B. $f(x, y) = \frac{x}{y}$, $f(x, y) = x^y$ oder $f(x) = \sqrt{x}$.

Prädikat

n-stellige Funktionen

$$f : A_1 \times \ldots A_n \to \mathbb{B},\ n \geq 0$$

mit der Zielmenge $\mathbb{B} = \{ 0, 1 \}$ heißen auch *Prädikate*.

Beispiel 2.17 **a)** Die Funktion $d : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{B}$ definiert durch

$$d(x, y, z) = \begin{cases} 1, & x + y = z \\ 0, & \text{sonst} \end{cases}$$

ist ein Prädikat, welches testet, ob die Summe von x und y gleich z ist.

b) Für das dreistellige Prädikat $\Delta : \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{B}$ definiert durch

$$\Delta(x,y,z) = \begin{cases} 1, & \text{falls } x,y,z \text{ Seitenlängen eines Dreiecks sein können} \\ 0, & \text{sonst} \end{cases}$$

gilt z.B. $\Delta(3,4,5) = 1$ und $\Delta(3,1,1) = 0$. □

Es sind prinzipiell zwei 0-stellige Prädikate möglich:

$$\textbf{true} \ : \ \to \ \{\,0,1\,\} \quad \text{mit} \quad \textbf{true}() \ = \ 1$$

sowie

$$\textbf{false} \ : \ \to \ \{\,0,1\,\} \quad \text{mit} \quad \textbf{false}() \ = \ 0$$

welche als Wahrheitswerte interpretiert werden können. Die Festlegung der Funktionen **true** und **false** könnte natürlich auch umgekehrt erfolgen, was aber unüblich ist. Im Kapitel 1.2.3 haben wir **true** mit $\underline{1}$ und **false** mit $\underline{0}$ benannt.

Jedes total definierte n-stellige Prädikat $f : A_1 \times \ldots A_n \to \mathbb{B}$, $n \geq 1$, legt eine n-stellige Relation $R_f \subseteq A_1 \times \ldots A_n$ wie folgt fest:

$$R_f = \{\,(x_1,\ldots,x_n) \in A_1 \times \ldots A_n \mid f(x_1,\ldots,x_n) = 1\,\}$$

Auf diese Weise erfolgt in Kapitel 1.7.3 die Interpretation von Prädikatsymbolen. Einem n-stelligen Prädikatensymbol P^n wird als Bedeutung eine n-stellige Relation R_{P^n} zugeordnet, $\mathcal{I}^*(P^n) = R_{P^n}$, und es gilt

$$\mathcal{I}^*(P^n)(x_1\ldots,x_n) = \begin{cases} 1, & \text{falls } (\mathcal{I}^*(x_1),\ldots,\mathcal{I}^*(x_n)) \in R_{P^n} \\ 0, & \text{sonst} \end{cases}$$

Lösungsmenge eines Prädikats

$R_f = \{\,(x_1,\ldots,x_n) \in A_1 \times \ldots A_n \mid f(x_1,\ldots,x_n) = 1\,\}$ heißt auch die *Lösungsmenge* von f. Dabei schreiben wir anstelle von $f(x_1,\ldots,x_n) = 1$ nur $f(x_1,\ldots,x_n)$ um auszudrücken, dass das Prädikat f auf das Argument $(x_1,\ldots,x_n)$ zutrifft. Bei der beschreibenden Darstellung von Mengen haben wir das bisher auch so praktiziert: $M = \{\,(x_1,\ldots,x_n) \mid P(x_1,\ldots,x_n)\,\}$ (siehe Festlegung (1.1) auf Seite 7 in Kapitel 1.1.2). Es gilt $M = R_P$.

2.2.3 Zusammenfassung

Bei rechtseindeutigen Relationen steht ein Element der Ausgangsmenge mit höchstens einem Element der Zielmenge in Beziehung. Solche Beziehungen werden funktional genannt.

Ist die Funktion linkseindeutig, d.h. hat jeder Wert der Funktion genau ein Argument, dann heißt sie injektiv. Die Umkehrrelation einer linkseindeutigen, also injektiven Funktion ist rechtseindeutig, also wieder eine Funktion.

Funktionen, die total, surjektiv und injektiv sind, heißen eineindeutig oder bijektiv.

Prädikate sind Funktionen, welche den Argumenten boolesche Werte zuweisen. Sie werden z.B. für die beschreibende Darstellung von Mengen verwendet.

2.3 Mächtigkeit von Mengen

In Folgerung 2.4 g) haben wir festgestellt, dass zwei endliche Mengen genau dann gleichmächtig sind, also die selbe Anzahl von Elementen enthalten, wenn es eine Bijektion zwischen diesen Mengen gibt, denn dann können die Elemente der beiden Mengen vollständig in beide Richtungen eins zu eins zugeordnet werden. Wir verallgemeinern diesen Begriff nun auf beliebige Mengen.

2.3.1 Definitionen und Beispiele

Gleichmächtige Mengen

Definition 2.15 Zwei Mengen A und B heißen *gleichmächtig* genau dann, wenn eine bijektive Funktion $f : A \to B$ existiert. Sind die Mengen A und B gleichmächtig, dann schreiben wir dafür auch $|A| = |B|$. Wir notieren $|A| \leq |B|$ und nennen A *höchstens gleichmächtig* zu B genau dann, wenn es eine totale, injektive Abbildung $f : A \to B$ gibt, und wir notieren $|A| < |B|$ und nennen B *mächtiger* als A genau dann, wenn $|A| \leq |B|$ und $|A| \neq |B|$ gelten. □

Unmittelbar aus diesen Definitionen folgt

Folgerung 2.5 **a)** A, B und C seien Mengen, und es sei $\sim \in \{=, \leq, <\}$, dann gilt: Ist $|A| \sim |B|$ und $|B| \sim |C|$, dann gilt auch $|A| \sim |C|$.

b) Ist $A \subseteq B$, dann ist $|A| \leq |B|$, denn die Abbildung $f : A \to B$ definiert durch $f(x) = x$ ist offensichtlich total und injektiv. □

Die folgenden Beispiele zeigen, dass die in der obigen Definition eingeführten Begriffe – zumindest beim ersten Eindruck – nicht unbedingt mit unseren intuitiven Vorstellungen über die Größenordnungen von Mengen übereinstimmen. So können Teilmengen einer Menge nicht nur höchstens gleichmächtig zu ihren Obermengen sein, sondern sogar gleichmächtig.

Beispiel 2.18 **a)** Es gilt nicht nur $|\mathbb{N}| \leq |\mathbb{N}_0|$, sondern $|\mathbb{N}| = |\mathbb{N}_0|$, obwohl intuitiv $\mathbb{N}_0$ ein Element, die 0, mehr enthält als $\mathbb{N}$. Wir können nämlich etwa die Bijektion $f : \mathbb{N}_0 \to \mathbb{N}$ durch $f(n) = n + 1$ definieren.

b) Für die Menge der positiven geraden Zahlen $\mathbb{G}_+$ gilt $|\mathbb{G}_+| = |\mathbb{N}_0|$, denn die Funktion $f : \mathbb{N}_0 \to \mathbb{G}_+$ definiert durch $f(k) = 2k$ ist bijektiv.

c) Auch die Menge $\mathbb{Z}$ der ganzen Zahlen ist gleichmächtig zur Menge der natürlichen Zahlen („obwohl es intuitiv gesehen doppelt so viele ganze wie natürliche

	1	2	3	4	...	q	...
1	1	2	4	7			
	↙	↙	↙				
2	3	5	8				
	↙	↙					
3	6	9					
	↙						
4	10						
$\vdots$							
p							
$\vdots$							

Abb. 2: Bijektion zwischen $\mathbb{N} \times \mathbb{N}$ und $\mathbb{N}$

Zahlen gibt“). Die Funktion $f : \mathbb{Z} \to \mathbb{N}_0$ definiert durch

$$f(z) = \begin{cases} 0, & \text{falls } z = 0 \\ 2z, & \text{falls } z > 0 \\ -(2z+1), & \text{falls } z < 0 \end{cases}$$

ist eine bijektive Abbildung, die den positiven ganzen Zahlen die geraden Zahlen und den negativen ganzen Zahlen die ungeraden Zahlen zuordnet. Mit b) und Folgerung 2.5 a) folgt zudem $|\mathbb{G}_+| = |\mathbb{Z}|$.

d) Es gilt $|\mathbb{N}\times\mathbb{N}| = |\mathbb{N}|$. Die Matrix in Abbildung 2 stellt eine mögliche Bijektion $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ dar.
Wir wollen nun die Bijektion angeben, die diese Matrix darstellt. In der ersten Spalte stehen die Nummern der Paare $(p, 1)$, $p \geq 1$. Diese ergeben sich durch (siehe Beispiel 3.3 a) auf Seite 125)

$$f(p,1) = \sum_{i=1}^{p} i = \frac{p(p+1)}{2} \tag{2.2}$$

Für $q \geq 2$ gilt, dass sich die Nummer an der Stelle (p, q) in der Matrix ergibt, indem man die Nummer an der Stelle $(p + 1, q - 1)$ (eine Zeile weiter und eine Spalte vorher) um 1 vermindert. Für f gilt also

$$f(p,q) = f(p+1, q-1) - 1,\ q \geq 2$$

Daraus ergibt sich

$$\begin{aligned} f(p,q) &= f(p+1,q-1) - 1 \\ &= f(p+2,q-2) - 2 \\ &\vdots \\ &= f(p+q-1,1) - (q-1),\ q \geq 2 \end{aligned}$$

Mit (2.2) folgt hieraus für $q \geq 1$

$$f(p,q) = \sum_{i=1}^{p+q-1} i - (q-1) = \frac{(p+q-1)(p+q)}{2} - (q-1)$$

Anschaulich verdeutlicht die Tabelle 2, dass f bijektiv ist; dies kann aber auch formal bewiesen werden, womit die Gleichmächtigkeit von $\mathbb{N} \times \mathbb{N}$ und $\mathbb{N}$ gezeigt ist.

Cantorsche Tupelfunktion

Aus der Funktion f lassen sich die sogenannten *Cantorschen k-Tupelfunktionen*

$$\mathit{cantor}_k : \mathbb{N}_0^k \to \mathbb{N}_0,\ k \geq 2$$

ableiten:

$$\mathit{cantor}_2(i,j) = f(i, j+2) = \frac{(i+j+1)(i+j+2)}{2} - (j+1)$$

$$\mathit{cantor}_{k+1}(i_1, \ldots, i_k, i_{k+1}) = \mathit{cantor}_2(\mathit{cantor}_k(i_1, \ldots, i_k), i_{k+1}),\ k \geq 2$$

cantor_k ist für jedes k bijektiv. Es gilt also $|\mathbb{N}_0^k| = |\mathbb{N}_0|$ für alle $k \geq 1$.

Gebräuchliche alternative Notationen zu $cantor_k(i_1, \ldots, i_k)$ sind $c_k(i_1, \ldots, i_k)$ und $\langle\, i_1, \ldots, i_k \,\rangle_k$.

e) Mithilfe von d) können wir nun zeigen, dass sogar die Menge $\mathbb{Q}$ der rationalen Zahlen gleichmächtig zur Menge der natürlichen Zahlen ist. Das ist intuitiv noch verwunderlicher als die Gleichmächtigkeit von $\mathbb{Z}$ und $\mathbb{N}$, denn die Menge $\mathbb{Q}$ ist dicht (siehe Beispiel 2.7 b). Wir betrachten zunächst die rationalen Zahlen größer 0:

$$\mathbb{Q}_+ = \left\{ \frac{p}{q} \mid p, q \in \mathbb{N} \right\}$$

Den Bruch $\frac{p}{q}$ können wir auch als Paar (p,q) darstellen und sehen damit sofort, dass $|\mathbb{Q}_+| = |\mathbb{N} \times \mathbb{N}|$ ist.

Analog können wir die negativen Brüche $\mathbb{Q}_- = \left\{ -\frac{p}{q} \mid p, q \in \mathbb{N} \right\}$ durch die Paare $(-p,q) \in -\mathbb{N} \times \mathbb{N}$ darstellen, dabei sei $-\mathbb{N} = \{-n \mid n \in \mathbb{N}\}$ die Menge der negativen ganzen Zahlen. Die Abbildung $g : -\mathbb{N} \times \mathbb{N} \to -\mathbb{N}$ definiert durch $g(-p,q) = -f(p,q)$ ist eine Bijektion, da f eine Bijektion ist.

Insgesamt erhalten wir eine bijektive Abbildung $h : \mathbb{Q} \to \mathbb{Z}$ durch

$$h(p,q) = \begin{cases} f(p,q), & p,q \in \mathbb{N} \\ 0, & p = 0 \\ -f(p,q), & p \in -\mathbb{N},\ q \in \mathbb{N} \end{cases}$$

Es gilt also $|\mathbb{Q}| = |\mathbb{Z}|$. Laut obigem Beispiel c) gilt $|\mathbb{Z}| = |\mathbb{N}_0|$ und damit mit Folgerung 2.5 a) $|\mathbb{Q}| = |\mathbb{N}_0|$, d.h. $\mathbb{Q}$ ist gleichmächtig zu $\mathbb{N}_0$.

Zu bemerken ist, dass bei den Bijektionen der Bruch $(p, q) \in \mathbb{Q}$ mit teilerfremden p und q verschieden von allen seinen Erweiterungen (tp, tq), $t \in \mathbb{N}$, betrachtet wird, d.h. jede rationale Zahl (außer der 0) wird unendlich oft verschiedenen natürlichen Zahlen zugeordnet. Man kann auch Bijektionen angeben, bei denen die Erweiterungen nicht berücksichtigt werden, darauf gehen wir aber nicht weiter ein. □

Der folgende *Satz von Cantor* besagt, dass eine Menge immer weniger mächtig als ihre Potenzmenge ist.

Satz von Cantor

Satz 2.7 Für jede Menge M gilt: $|M| < |\mathcal{P}(M)|$.

Beweis: Wir nehmen an, dass $\mathcal{P}(M)$ abzählbar ist. Dann gibt es eine Bijektion $f : M \to \mathcal{P}(M)$. Dann gilt für jedes Element $x \in M$, dass entweder x in seinem Bild $f(x)$ enthalten ist oder nicht, d.h. es gilt für jedes $x \in M$ entweder $x \in f(x)$ oder $x \notin f(x)$. Wir bilden nun die Menge $D = \{x \in M \mid x \notin f(x)\}$. Offensichtlich gilt $D \subseteq M$ und damit $D \in \mathcal{P}(M)$. Da f bijektiv und damit surjektiv ist, muss die Menge D ein Urbild haben, d.h. es muss ein $x_D \in M$ existieren mit $f(x_D) = D$. Für dieses Element gibt es zwei Möglichkeiten: entweder $x_D \in D$ oder $x_D \notin D$. Beide Fälle führen zum Widerspruch

$$x_D \in D \Rightarrow x_D \notin f(x_D) \Rightarrow x_D \notin D$$
$$x_D \notin D \Rightarrow x_D \in f(x_D) \Rightarrow x_D \in D$$

womit unsere Annahme widerlegt und die Behauptung gezeigt ist. □

Aus dem Satz folgt, dass es keine Menge geben kann, die mächtiger als alle anderen Mengen ist. Denn gäbe es eine solche Menge M, dann wäre $\mathcal{P}(M)$ echt mächtiger und damit mächtiger als die mächtigste Menge. Außerdem kann man eine unendliche Hierarchie von Klassen von Mengen mit wachsender Mächtigkeit unterscheiden:

$$|\mathbb{N}| < |\mathcal{P}(\mathbb{N})| < |\mathcal{P}(\mathcal{P}(\mathbb{N}))| < |\mathcal{P}(\mathcal{P}(\mathcal{P}(\mathbb{N})))| < \ldots \tag{2.3}$$

Der folgende Satz, das sogenannte *Cantor-Bernstein-Schröder-Theorem* stellt ein Hilfsmittel zur Verfügung, mit dem die Gleichmächtigkeit von Mengen gezeigt werden kann.

Cantor-Bernstein-Schröder-Theorem

Satz 2.8 Für zwei Mengen A und B gilt $|A| = |B|$ genau dann, wenn $|A| \leq |B|$ und $|B| \leq |A|$ gelten. □

Der Beweis ist in einer Richtung offensichtlich: Aus $|A| = |B|$ folgt, dass es eine Bijektion $f : A \to B$ gibt. Damit gibt es aber auch injektive Abbildungen $f : A \to B$ und $f^{-1} : B \to A$, womit $|A| \leq |B|$ bzw. $|B| \leq |A|$ gezeigt ist. Der Beweis, dass aus $|A| \leq |B|$ und $|B| \leq |A|$ die Gleichmächtigkeit von A und B folgt, ist nicht ganz so einfach und soll hier nicht geführt werden.

In den folgenden Beispielen wenden wir diesen Satz an. Dabei verwenden wir die Tatsache, dass sich jede reelle Zahl $z \in \mathbb{R}$ als eine unendliche Dezimalzahl darstellen lässt. Für irrationale Zahlen, die keine rationale Zahlen sind, wie etwa

$\sqrt{2}$, ist das offensichtlich. Bei den rationalen kann man zeigen, dass sie sich entweder als endliche Dezimalzahlen darstellen lassen, wie z.B. $\frac{1}{4} = 0.25$, oder als periodische Dezimalzahlen, wie z.B. $\frac{1}{3} = 0.\overline{3}$ oder $\frac{1}{7} = 0.\overline{142857}$. Endliche Dezimalzahlen kann man durch eine periodische Dezimalzahl beliebig annähern, wie z.B. 0.5 durch $0.4\overline{9}$; man kann zeigen, dass es keine andere Dezimalzahl gibt, die näher an 0.5 liegt, als $0.4\overline{9}$. So können wir also davon ausgehen, dass sich jede rationale Zahl durch eine periodische Dezimalzahl darstellen lässt. Da zudem, wie schon erwähnt, die irrationalen Zahlen nur als unendliche Dezimalzahlen darstellbar sind, folgt, dass sich alle reellen Zahlen als als unendliche Dezimalzahlen darstellen lassen. Wenn wir vor den ganzzahligen Anteil unendliche viele Nullen hinzufügen, dann sind diese Zahlen sogar in beide Richtungen unendlich. Betrachten wir z.B. die Zahl $372.\overline{142857}$, dann stellen wir diese wie folgt dar:

$$\ldots 0 \ldots 0372.142857142857142857 \ldots$$

Im Allgemeinen bezeichnen wir die Ziffern vor dem Dezimalpunkt mit z_i, $i \in \mathbb{N}_0$, und die Ziffern hinter dem Dezimalpunkt mit z_{-j}, $j \in \mathbb{N}$. So ergibt sich die allgemeine Darstellung einer reellen Zahl z durch

$$z = \ldots z_{m-1} \ldots z_0.z_{-1} \ldots z_n \ldots$$

Der Wert dieser Zahl ergibt sich dann durch

$$wert(z) = \sum_{i=-\infty}^{\infty} z_i \cdot 10^i$$

Beispiel 2.19 **a)** Es sei $[0,1) = \{z \in \mathbb{R} \mid 0 \leq z < 1\}$ die Menge der reellen Zahlen zwischen 0 und 1 einschließlich 0 und ausschließlich 1. Es gilt $|\,[0,1)\,| = |\mathbb{R}|$. Wir zeigen diese Behauptung mithilfe von Satz 2.8. Es gilt einerseits $|\,[0,1)\,| \leq |\mathbb{R}|$, weil die Abbildung $f : [0,1) \to \mathbb{R}$ definiert durch $f(x) = x$ offensichtlich total und injektiv ist. Andererseits gilt auch $|\mathbb{R}| \leq |\,[0,1)\,|$, weil wir jeder reellen Zahl

$$z = v \ldots z_{m-1} \ldots z_0.z_{-1} \ldots z_n \ldots \in \mathbb{R}$$

den Dezimalbruch

$$z' = 0.v'\, z_0 z_{-1} z_1 z_{-2} z_2 \ldots \in [0,1)$$

zuordnen können; dabei sei $v \in \{+,-\}$ das Vorzeichen, welches in z' durch $v' = 0$, falls $v = +$ bzw. durch $v' = 1$, falls $v = -$ ist, kodiert wird. z' hat den Wert

$$wert(z') = v' \cdot 10^{-1} + z_0 \cdot 10^{-2} + \sum_{i=1}^{\infty} \left(z_{-i} \cdot 10^{-2i-1} + z_i \cdot 10^{-2(i+1)} \right)$$

Diese Zuordnung ist total und injektiv.

b) Man kann das Intervall sogar beliebig klein machen: Sei $\epsilon \in \mathbb{R}$ mit $0 < \epsilon \leq 1$ und $[0, \epsilon) = \{z \in \mathbb{R} \mid 0 \leq z < \epsilon\}$, dann gilt $|\,[0, \epsilon)\,| = |\mathbb{R}|$. Wir zeigen $|\,[0, \epsilon)\,| = |\,[0, 1)\,|$, dann folgt mit a) und Folgerung 2.5 a) diese Behauptung. Die Beziehung $|\,[0, \epsilon)\,| \leq |\,[0, 1)\,|$ ist offensichtlich, wir wählen wieder die totale Injektion $f : [0, \epsilon) \to [0, 1)$ definiert durch $f(x) = x$. Die Abbildung $g : [0, 1) \to [0, \epsilon)$ definiert durch $g(x) = \epsilon \cdot x$ ist ebenfalls total und injektiv, woraus $|\,[0, 1)\,| \leq |\,[0, \epsilon)\,|$ folgt.

c) Es gilt $|\mathbb{R}| = |\mathcal{P}(\mathbb{N})|$, die Menge der reellen Zahlen und die Potenzmenge der natürlichen Zahlen sind gleichmächtig. Dazu überlegen wir, dass $|\,[0, 1)\,| = |\mathcal{P}(\mathbb{N})|$ ist, dann folgt die Behauptung mit a) und Folgerung 2.5 a). Wir benutzen die Tatsache, dass jede reelle Zahl $z \in [0, 1)$ eineindeutig als Dualbruch dargestellt werden kann:

$$\hat{z} = 0.z_1 z_2 \ldots$$

mit $z_i \in \{0, 1\}$. Der Wert von $\hat{z}$ ist

$$wert(\hat{z}) = \sum_{i=1}^{\infty} z_i \cdot 2^{-i}$$

Betrachten wir als Beispiel den Dualbruch $\hat{z} = 0.1011$, dann ist

$$wert(\hat{z}) = \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = \frac{11}{16} = 0.6875$$

Es gibt endliche und unendliche Dualbrüche; wenn wir uns die endlichen hinter der letzten Stelle mit unendlich vielen Nullen ergänzt denken, sind alle Dualbrüche unendlich. Wir ordnen nun jeder Zahl $z \in [0, 1)$ ihren unendlichen Dualbruch $\hat{z} = 0.z_1 z_2 \ldots$ zu. Mithilfe dieses Dualbruches bilden wir die Menge $N_z = \{i \mid z_i = 1\}$. Unserem obigen Beispiel $z = 0.6875$ wird also die Menge $N_{0.6875} = \{1, 3, 4\}$ zugeordnet. Die so entstehenden Mengen N_z für $z \in [0, 1)$ sind Teilmengen von $\mathbb{N}_0$, also Elemente von $\mathcal{P}(\mathbb{N}_0)$. Diese Zuordnung ist total und injektiv, somit gilt $|\,[0, 1)\,| \leq |\mathcal{P}(\mathbb{N}_0)|$. Jetzt betrachten wir irgendein Element $N \in \mathcal{P}(\mathbb{N}_0)$. Ist die Menge $N = \{n_1, n_2, \ldots, n_k\}$ endlich, dann ordnen wir ihr den Dualbruch

$$z_N = \sum_{i=1}^{k} 2^{-n_i}$$

zu; ist die Menge $N = \{n_1, n_2, \ldots\}$ unendlich, dann ordnen wir ihr den Dualbruch

$$z_N = \sum_{i=1}^{\infty} 2^{-n_i}$$

zu. Insgesamt ist dadurch eine totale Injektion von $\mathcal{P}(\mathbb{N}_0)$ nach $[0, 1)$ festgelegt ist, womit $|\mathcal{P}(\mathbb{N}_0)| \leq |\,[0, 1)\,|$ gezeigt ist.

d) Aus c) und der Mächtigkeitshierarchie (2.3) folgt $|\mathbb{N}| < |\mathbb{R}|$: Die Menge der reellen Zahlen ist mächtiger als die Menge der natürlichen Zahlen, und, da $\mathbb{N}$ und $\mathbb{Q}$ gleichmächtig sind (siehe Beispiel 2.18 e), gilt auch $|\mathbb{Q}| < |\mathbb{R}|$, d.h. die Menge der reellen Zahlen ist mächtiger als die Menge der rationalen Zahlen. □

2.3.2 Zusammenfassung

Mithilfe total injektiver (höchstens gleichmächtig) und bijektiver (gleichmächtig) Funktionen kann die Mächtigkeit von Mengen verglichen werden. Es stellt sich heraus, dass die Menge der rationalen Zahlen, obwohl dicht geordnet, gleichmächtig zur Menge der natürlichen Zahlen ist, während die Menge der reellen Zahlen, sogar jedes (noch so kleine) nicht leere Intervall von reellen Zahlen, nicht gleichmächtig zur Menge der natürlichen Zahlen ist.

Die Cantorschen k-Tupelfunktionen stellen hilfreiche Bijektion zwischen $\mathbb{N}_0^k$ und $\mathbb{N}_0$ zur Verfügung.

Der Cantorsche Satz besagt, dass die Potenzmenge jeder Menge echt mächtiger als die Menge selbst ist. Durch fortgesetzte Anwendung des Potenzmengenoperators ergibt sich daraus eine unendliche Hierarchie von echt monoton wahchsenden Mächtigkeiten.

Das Cantor-Bernstein-Schröder-Theorem liefert eine oft anwendbare Methode zum Nachweis der Gleichmächtigkeit von Mengen.

3 Zahlenmengen

Wir haben in den vorigen Kapiteln bereits die aus der Schule und dem täglichen Leben bekannten Zahlenmengen in Beispielen verwendet. In diesem Kapitel werden wir zunächst die natürlichen Zahlen axiomatisch einführen und dann aus ihnen mithilfe von Äuivalenzrelationen schrittweise die ganzen Zahlen und die rationalen Zahlen konstruieren.

3.1 Die Menge der natürlichen Zahlen

Wir werden zunächst den Begriff der Zählstruktur durch dafür wesentliche Eigenschaften axiomatisch festlegen und dann die Menge der natürlichen Zahlen als ein Beispiel für eine solche Zählstruktur ansehen. Zählen basiert darauf, zu einer Zahl die nächste zu bestimmen, also zu einer Zahl die Zahl 1 hinzuaddieren zu können. Wir werden auf der Basis dieser elementaren Operation arithmetische Verknüpfungen wie die Addition, die Multiplikation und das Potenzieren mithilfe rekursiver Definitionen einführen und die für diese Rechenarten bekannten Regeln auflisten. Da die natürlichen Zahlen zum Zählen verwendet werden, liegt es nahe, diese Menge als Referenzmenge für die Abzählbarkeit von beliebigen Mengen zu betrachten.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- die axiomatische Einführung der Menge der natürlichen Zahlen verstehen,
- die Rechenregeln für natürliche Zahlen kennen.

3.1.1 Einführung der Menge der natürlichen Zahlen

Man kann Zahlen, und so sind diese vielleicht auch in der Menschheitsgeschichte entstanden, als Abstraktionen von Mengen mit derselben Anzahl von Elementen betrachten. Sieben Äpfel, sieben Birnen, sieben Ziegen bilden Mengen mit 7 Elementen, welches man z.B. durch sieben Kerben in einem Holzstab oder sieben Striche im Sand oder durch die Symbole 7 oder *VII* notieren kann.

Was braucht man nun „wirklich“ zum Zählen? Nun, wir brauchen eine unendliche Menge, deren Elemente zum Zählen verwendet werden können (Strichfolgen, Zahlensymbole, ...). Dann brauchen wir eine Zahl, bei der das Zählen beginnt, eine „kleinste“ Zahl, und wir müssen festlegen, dass verschiedene Zahlensymbole auch verschiedene Anzahlen angeben.

K.-U. Witt, *Mathematische Grundlagen für die Informatik*,
DOI 10.1007/978-3-658-03079-7_3, © Springer Fachmedien Wiesbaden 2013

Diese grundlegenden Eigenschaften des Zählens wollen wir mathematisch mithilfe einer *Zählstruktur*

$$(\mathcal{S}, 0, s)$$

festlegen. Dabei sei $\mathcal{S}$ eine nicht leere Menge, $0 \in \mathcal{S}$ ein ausgezeichnetes Element, welches wir *Null* nennen, sowie $s : \mathcal{S} \to \mathcal{S}$ eine Funktion, die wir *Nachfolgerfunktion* nennen. Die Zählstruktur soll dabei folgende Axiome erfüllen:

Zählstruktur

(P1) Für alle $x \in \mathcal{S}$ gilt: $s(x) \neq 0$, d.h. 0 kann kein Nachfolger einer Zahl sein. Damit wird 0 quasi als kleinste Zahl festgelegt.

(P2) s ist injektiv, d.h. für alle $x, y \in \mathcal{S}$ gilt: Ist $x \neq y$, dann ist auch $s(x) \neq s(y)$, verschiedene Zahlen haben verschiedene Nachfolger. Das bedeutet allgemeiner, dass verschiedene Zahlen auch verschiedene Anzahlen bedeuten.

Induktionsaxiom

(P3) Für jede Teilmenge M von $\mathcal{S}$ gilt: Ist $0 \in M$ und folgt daraus, dass $x \in M$ ist, auch, dass $s(x) \in M$ ist, dann gilt $M = \mathcal{S}$. Dieses ist das sogenannte *Induktionsaxiom*.

Dedekindtripel Peano-Axiome

Diese Axiome gehen auf Dedekind[19] und Peano[20] zurück. Deswegen wird $(\mathcal{S}, 0, s)$ auch *Dedekindtripel* genannt, und die Axiome P1 - 3 gehören zu den *Peano-Axiomen* zur Definition der Menge der natürlichen Zahlen.

Die konkreten Strukturen im folgenden Beispiel erfüllen diese Axiome, sind also Beispiele für Zählstrukturen.

Beispiel 3.1 **a)** $(\mathcal{S}_1, 0_1, s_1)$ mit $\mathcal{S}_1 = \{\varepsilon, |, ||, |||, \dots, \}$, $0_1 = \varepsilon$, die leere Strichfolge, und $s_1(x) = x|$.

b) $(\mathcal{S}_2, 0_2, s_2)$ mit $\mathcal{S}_2 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}, \dots\}$ mit $0_2 = \emptyset$ und $s_2(x) = x \cup \{x\}$. □

Isomorphie

Wir wollen zwei Peano-Strukturen $(\mathcal{S}_1, 0_1, s_1)$ und $(\mathcal{S}_2, 0_2, s_2)$ *strukturgleich* oder *isomorph* nennen, falls es eine bijektive Abbildung

$$\varphi : \mathcal{S}_1 \to \mathcal{S}_2$$

gibt, für die

$$\varphi(s_1(x)) = s_2(\varphi(x)) \text{ für alle } x \in \mathcal{S}_1 \tag{3.1}$$

gilt. Die Abbildung φ ordnet also jedem Element der einen genau ein Element der anderen Struktur zu, und diese Zuordnung ist verträglich mit den Nachfolgerfunktionen in beiden Strukturen. Das bedeutet, dass, wenn $x \in \mathcal{S}_1$ dem Element $y \in \mathcal{S}_2$ zugeordnet wird, dann werden auch ihre Nachfolger $s_1(x) \in \mathcal{S}_1$ und $s_2(y) \in \mathcal{S}_2$ einander zugeordnet. Sind $\mathcal{S}_1$ und $\mathcal{S}_2$ isomorph, so schreiben wir: $(\mathcal{S}_1, 0_1, s_1) \cong (\mathcal{S}_2, 0_2, s_2)$.

19 Julius Wilhelm Richard Dedekind (1831 - 1916), deutscher Mathematiker, lieferte unter anderem zur Defintion und Eigenschaften von Zahlenmengen sowie zur Algebra fundamentale Beiträge.

20 Giuseppe Peano (1858 - 1932), italienischer Mathematiker und Logiker, beschäftigte sich mit axiomatischen Ansätzen zur Beschreibung von Mengen und Strukturen.

Beispiel 3.2 Wir zeigen, dass die beiden Zählstrukturen aus Beispiel 3.1 isomorph sind. Dazu legen wir die Abbildung $\varphi : \mathcal{S}_1 \to \mathcal{S}_2$ wie folgt rekursiv fest:

$$\begin{aligned}\varphi(\varepsilon) &= \emptyset \\ \varphi(x|) &= \varphi(x) \cup \{\varphi(x)\}\end{aligned}$$

Wir berechnen mithilfe dieser rekursiven Definition z.B. $\varphi(|||)$ schrittweise:

$$\varphi(|) = \varphi(\varepsilon) \cup \{\varphi(\varepsilon)\} = \emptyset \cup \{\emptyset\} = \{\emptyset\}$$

$$\varphi(||) = \varphi(|) \cup \{\varphi(|)\} = \{\emptyset\} \cup \{\{\emptyset\}\} = \{\emptyset, \{\emptyset\}\}$$

$$\varphi(|||) = \varphi(||) \cup \{\varphi(||)\} = \{\emptyset, \{\emptyset\}\} \cup \{\{\emptyset\,, \{\emptyset\}\}\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$$

Diese Abbildung φ ist bijektiv und sie erfüllt die Bedingung (3.1):

$$\varphi(s_1(x)) = \varphi(x|) = \varphi(x) \cup \{\varphi(x)\} = s_2(\varphi(x))$$

Es gilt also: $(\mathcal{S}_1, \varepsilon, s_1) \cong (\mathcal{S}_2, \emptyset, s_2)$. □

Durch Verallgemeinerung der Überlegungen in obigem Beispiel kann man beweisen, dass alle Zählstrukturen isomorph zueinander sind: Zu je zwei Strukturen, die die Peano-Axiome erfüllen, lässt sich eine bijektive Abbildung finden, die die Bedingung (3.1) erfüllt. Der folgende Satz besagt, dass diese Abbildung in jedem Fall die beiden Nullen einander zuordnet.

Satz 3.1 Für alle Zählstrukturen $(\mathcal{S}_1, 0_1, s_1)$ und $(\mathcal{S}_2, 0_2, s_2)$ gilt

$$(\mathcal{S}_1, 0_1, s_1) \cong (\mathcal{S}_2, 0_2, s_2)$$

und für alle Isomorphismen φ zwischen diesen Strukturen gilt

$$\varphi(0_1) = 0_2$$

□

Übungsaufgaben

3.1 Beweisen Sie Satz 3.1! □

Gemäß Satz 3.1 sind also alle Zählstrukturen bis auf die Benennung ihrer Elemente identisch. Wir können für den täglichen Gebrauch also irgendeine Zählstruktur auswählen. Da die Notation der Elemente der Zählstrukturen in Beispiel 3.1 in Bezug auf ihr Hinschreiben, ihr Lesen oder in Bezug auf das Rechnen

mit ihnen sehr umständlich ist, stellt sich die Frage nach geeigneten Zahlensymbolen. Im Verlaufe der Geschichte hat es hier unterschiedliche Entwicklungen gegeben, wie z.B. die römischen Zahlen oder die arabischen Zahlen. Die arabischen Zahlen haben sich durchgesetzt, weil sie auf einem Stellenwertsystem basieren, woraus sich eine systematische Notation ergibt sowie einfache Verfahren zur Durchführung von Rechenoperationen.

Die Menge der natürlichen Zahlen

Wenn wir nun von der Struktur $(\mathcal{S}_1, \varepsilon, s_1)$ im Beispiel 3.1 ausgehen und für eine Folge von n Strichen das arabische Zahlensymbol für n und den Nachfolger $s(n)$ von n mit $n+1$ notieren, erhalten wir die Menge der uns bekannten natürlichen Zahlen

$$\mathbb{N}_0 = \{\, 0, 1, 2, 3, \dots \}$$

als weitere Zählstruktur $(\mathbb{N}_0, 0, s)$. Da man gelegentlich Aussagen über alle natürlichen Zahlen ohne die Null machen möchte, führen wir $\mathbb{N}$ als Symbol für diese Menge ein:

$$\mathbb{N} = \mathbb{N}_0 - \{0\} = \{\, 1, 2, 3, \dots \}$$

3.1.2 Rechnen mit natürlichen Zahlen

Addition und Multiplikation natürlicher Zahlen

Mithilfe der Nachfolgerfunktion s lassen sich nun die elementaren arithmetischen Operationen für natürliche Zahlen, *Addition* und *Multiplikation*, rekursiv definieren (siehe auch Kapitel 4).

(1) Wir führen die Addition zweier Zahlen x und y auf die y-fache Addition der 1 zu x zurück: $\mathit{add} : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$\begin{aligned} \mathit{add}(x, 0) &= x \\ \mathit{add}(x, s(y)) &= s(\mathit{add}(x, y)) \end{aligned}$$

Mit dieser Berechnungsvorschrift ergibt sich z.B.

$$\begin{aligned} \mathit{add}(2,3) &= \mathit{add}(2, s(2)) = s(\mathit{add}(2,2)) = s(\mathit{add}(2, s(1))) \\ &= s(s(\mathit{add}(2,1))) = s(s(\mathit{add}(2, s(0)))) \\ &= s(s(s(\mathit{add}(2,0)))) = s(s(s(2))) = s(s(3)) = s(4) \\ &= 5 \end{aligned}$$

(2) Mithilfe der Addition können wir nun die Multiplikation $\mathit{mult} : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definieren:

$$\begin{aligned} \mathit{mult}(x, 0) &= 0 \\ \mathit{mult}(x, s(y)) &= \mathit{add}(\mathit{mult}(x, y), x) \end{aligned}$$

Die Multiplikation von x und y wird hier durch y-fache Addition von x mit sich selbst berechnet.

(3) Analog können wir nun die Potenzfunktion mithilfe der Multiplikation definieren: $\mathit{exp} : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ mit

$$\begin{aligned} \mathit{exp}(x, 0) &= 1 \\ \mathit{exp}(x, s(y)) &= \mathit{mult}(\mathit{exp}(x, y), x) \end{aligned}$$

Die Potenzierung von x mit y wird hier durch y-fache Multiplikation von x mit sich selbst berechnet.

In analoger Art und Weise können weitere arithmetische Operationen definiert werden, die letztendlich immer auf der Addition und damit auf der Addition der 1, d.h. auf der Nachfolgerfunktion s – also auf dem Zählen – basieren. Und zu diesem Zweck, nämlich zum Zählen, haben wir die natürlichen Zahlen ja gerade eingeführt.

Übungsaufgaben

3.2 (1) Berechnen Sie $\mathit{mult}(2, 3)$!

(2) Berechnen Sie $\mathit{exp}(2, 3)$! □

Im Folgenden schreiben wir wie üblich $x + y$ anstelle von $\mathit{add}(x, y)$ und $x \cdot y$ oder xy anstelle von $\mathit{mult}(x, y)$ sowie x^y für $\mathit{exp}(x, y)$.

Abgeschlossenheit von Addition und Multiplikation

Die natürlichen Zahlen sind abgeschlossen bezüglich der Addition und Multiplikation. Das bedeutet, dass die Summe bzw. das Produkt zweier natürlicher Zahlen wieder eine natürliche Zahl ist. Die natürlichen Zahlen sind nicht abgeschlossen gegenüber Subtraktion und Division, denn die Differenz bzw. der Quotient zweier natürlicher Zahlen muss keine natürliche Zahl sein. Auf Erweiterungen der Menge der natürlichen Zahlen auf Zahlenmengen, in denen auch diese Operationen abgeschlossen sind, gehen wir in Kapitel 3.6 und 3.7 ein.

3.1.3 Rechenregeln in $\mathbb{N}_0$

Mithilfe vollständiger Induktion (siehe Kapitel 3.2) und bereits bewiesener Eigenschaften kann man beweisen, dass für beliebige Zahlen $x, y, z \in \mathbb{N}_0$ folgende bekannte Rechenregeln gelten:

Rechenregeln

(A1) *Assoziativgesetz der Addition:* $(x + y) + z = x + (y + z)$, d.h. bei der Addition von mehr als zwei Zahlen kommt es nicht auf die Reihenfolge der Ausführung dieser Operation an. Deshalb können die Klammern auch weggelassen werden: $x + y + z$.

(A2) *Kommutativgesetz der Addition:* $x + y = y + x$, d.h. das Ergebnis einer Addition ist unabhängig von der Reihenfolge der Operanden.

(A3) *Neutrales Element der Addition:* $x + 0 = 0 + x = x$, d.h. die Addition mit 0 verändert den anderen Operanden nicht.

(M1) *Assoziativgesetz der Multiplikation:* $(xy)z = x(yz)$.

(M2) *Kommutativgesetz der Multiplikation:* $xy = yx$.

(M3) *Neutrales Element der Multiplikation:* $x \cdot 1 = 1 \cdot x = x$, d.h. die Multiplikation mit 1 verändert den anderen Operanden nicht.

(D1) *Distributivgesetz 1:* $x(y + z) = xy + xz$.

(D2) *Distributivgesetz 2:* $(x + y)z = xz + yz$.

In gewissem Rahmen sind auch die Umkehroperationen *Subtraktion* und *Division* zur Addition bzw. zur Multiplikation für natürliche Zahlen definiert:

- *Subtraktion:* Gilt $x + y = z$, dann gilt $x = z - y$. x heißt dann auch die *Differenz* von z und y. Mithilfe des Kommutativgesetzes (A2) kann man im Übrigen sofort ableiten, dass auch $y = z - x$ gelten muss.
- *Division:* Gilt $xy = z$, dann gilt $x = z : y$. x heißt dann auch der *Quotient* von z und y. Anstelle von $z : y$ notieren wir auch z/y oder $\frac{z}{y}$.

Die Relation $\leq \subseteq \mathbb{N}_0 \times \mathbb{N}_0$ definiert durch $x \leq y$ genau dann, wenn eine Zahl $d \in \mathbb{N}_0$ existiert mit $x + d = y$ legt auf $\mathbb{N}_0$ eine totale Ordnung fest (siehe Abschnitt 2.1.3). Für diese Ordnung gelten die beiden folgenden *Verträglichkeitsregeln*, auch *Monotonieregeln* genannt:

Monotonieregeln

(O1) sind $x, y, z \in \mathbb{N}_0$ und ist $x \leq y$, dann gilt auch $x + z \leq y + z$,

(O2) sind $x, y, z \in \mathbb{N}_0$ und ist $x \leq y$, dann gilt auch $xz \leq yz$.

Anstelle von $x \leq y$ schreiben wir auch $y \geq x$.

Summationssymbol

Zur Notation der Summe von mehreren Summanden $x_1, x_2, \ldots, x_n$ verwendet man oft auch das Summationssymbol $\sum$:

$$x_1 + x_2 + \ldots + x_n = \sum_{i=1}^{n} x_i$$

Läuft der *Summationsindex* i nicht zwischen den Grenzen 1 und n, sondern zwischen u und o mit $u, o \in \mathbb{N}_0$ und $u \leq o$, dann notieren wir

$$x_u + x_{u+1} + \ldots + x_o = \sum_{i=u}^{o} x_i$$

u heißt untere und o heißt obere *Index-* oder *Summationsgrenze*. Für den Fall $u > o$ legen wir $\sum_{i=u}^{o} x_i = 0$ fest.

Analog wird für die Multiplikation mehrerer Faktoren $x_u, x_{u+1}, \ldots x_o$

Multiplikationssymbol

$$x_u \cdot x_{u+1} \cdot \ldots \cdot x_o = \prod_{i=u}^{o} x_i$$

geschrieben. Für den Fall $u > o$ legen wir $\prod_{i=u}^{o} x_i = 0$ fest.

Sowohl bei der Addition als auch bei der Multiplikation kann die Anzahl der Summanden bzw. die Anzahl der Faktoren unendlich sein, d.h. es kann $u = -\infty$ oder $o = \infty$ oder $u = -\infty$ und $o = \infty$ sein. Man spricht dann von einer unendlichen Summe bzw. von einem unendlichen Produkt.

Aus den Distributivgesetzen lässt sich das folgende verallgemeinerte Distributivgesetz ableiten:

Verallgemeinertes Distributivgesetz

$$\begin{aligned}\left(\sum_{i=1}^{m} x_i\right)\left(\sum_{j=1}^{n} y_j\right) &= (x_1 + x_2 + \ldots + x_m)(y_1 + y_2 + \ldots + y_n) \\ &= x_1y_1 + x_1y_2 + \ldots + x_1y_n \\ &\quad + \; x_2y_1 + x_2y_2 + \ldots + x_2y_n \\ &\quad \vdots \\ &\quad + \; x_my_1 + x_my_2 + \ldots + x_my_n \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} x_iy_j\end{aligned}$$

Wegen der Kommutativität der Addition gilt

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_iy_j = \sum_{j=1}^{n}\sum_{i=1}^{m} x_iy_j$$

3.1.4 Zusammenfassung

Natürliche Zahlen sind ein Beispiel für eine Zählstruktur. Zählstrukturen werden durch die Peano-Axiome festgelegt: Eine Zählstruktur enthält ein kleinstes Element, mit dem das Zählen beginnt, und verschiedene Elemente stehen für verschiedene Anzahlen. Aufgrund der rekursiven Struktur der natürlichen Zahlen lassen sich alle arithmetischen Operationen jeweils rekursiv und schrittweise aufeinander aufbauend festlegen, und mithilfe vollständiger Induktion lassen sich die bekannten arithmetischen Rechenregeln nachweisen. Die natürlichen Zahlen sind abgeschlossen gegenüber Addition und Multiplikation, d.h. die Addition bzw. die Multiplikation zweier natürlicher Zahlen ist wieder eine natürliche Zahl. Subtraktion und Division hingegen sind nur eingeschränkt möglich, ihr Ergebnis muss keine natürliche Zahl sein.

3.2 Vollständige Induktion und verallgemeinertes Rekursionsschema

[Vollständige Induktion und verallgemeinertes Rekursionsschema]
Die Menge der natürlichen Zahlen ist induktiv definiert mithilfe der Peano-Axiome. Das Axiom (P3) gibt eine Methode vor, die Vollständige Induktion, mit der bewiesen werden kann, dass ein Prädikat auf alle natürlichen Zahlen zutrifft.

In früheren Kapiteln haben wir Strukturen, wie z.B. die Syntax aussagenlogischer Formeln, und Eigenschaften von Strukturen, wie z.B. die Semantik aussagenlogischer Formeln, rekursiv definiert. Um Aussagen über solche Strukturen und deren Eigenschaften zu beweisen, bietet es sich deshalb an, solche Beweise nach einem allgemeinen rekursiven Schema durchzuführen.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- die Beweismethode der vollständigen Induktion verstehen und anwenden können,
- das verallgemeinerte Rekursionschema erklären und anwenden können.

3.2.1 Vollständige Induktion

Vollständige Induktion ist eine Methode, mit der bewiesen werden kann, dass ein Prädikat auf die Menge der natürlichen Zahlen zutrifft, d.h. dass ein Prädikat $P : \mathbb{N}_0 \to \mathbb{B}$ die Lösungsmenge $R_P = \mathbb{N}_0$ besitzt, d.h. dass P von allen natürlichen Zahlen erfüllt wird (siehe Abschnitt 2.2.2). Die Menge der natürlichen Zahlen wird in Kapitel 3.1.1 eingeführt. Grundlage für die Vollständige Induktion ist dabei das Axiom P3 (siehe Seite 118).

Ein Beweis mit vollständiger Induktion erfolgt dementsprechend gemäß folgendem Schema:

Induktionsanfang

- *Induktionsanfang:* Zeige, dass $P(0)$ gilt, d.h. dass $0 \in R_P$ gilt.

Induktionsschritt
Induktionsannahme
Induktions-behauptung

- *Induktionsschritt:* Zeige: $\forall n \in \mathbb{N}_0 : P(n) \Rightarrow P(n+1)$. Dabei heißt $P(n)$ *Induktionsannahme* oder *Induktionsvoraussetzung* und $P(n+1)$ *Induktionsbehauptung*.

Haben wir für ein Prädikat $P : \mathbb{N}_0 \to \mathbb{B}$ Induktionsanfang und Induktionsschritt gezeigt, dann wissen wir, dass $P(n)$ für alle $n \in \mathbb{N}_0$ gilt, d.h. dass

$$R_P = \{\, n \in \mathbb{N}_0 \mid P(n) \} = \mathbb{N}_0$$

gilt: Das Prädikat P trifft auf alle natürlichen Zahlen zu.

Bemerkung 3.1 Veranschaulichen kann man diese Methode an folgendem Problem: Wann bin ich in der Lage, eine (unendliche) Leiter hinauf zu steigen. Nun, ich werde es schaffen, wenn ich in der Lage bin, die folgenden beiden

Schritte zu tun: Ich muss die erste Sprosse besteigen können (das entspricht dem Induktionsanfang), und wenn ich auf irgendeiner Stufe stehe (das entspricht der Induktionsannahme), dann muss ich auf die nächste Sprosse steigen können (damit habe ich den Induktionsschritt vollzogen). □

Beispiel 3.3 **a)** Zeige: $\forall n \in \mathbb{N}_0 : \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Das Prädikat

$$P(n) = \sum_{i=0}^{n} i = \frac{n(n+1)}{2} \tag{3.2}$$

behauptet also, dass sich $\sum_{i=0}^{n} i$, die Summe der Zahlen von 0 bis n, für jede natürliche Zahl n durch den Ausdruck $\frac{n(n+1)}{2}$berechnen lässt.

Wir beweisen die Behauptung mit vollständiger Induktion gemäß dem obigen Schema:

Induktionsanfang: Zeige, dass $P(0)$ gilt. Es ist

$$\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$$

also gilt $P(0)$.

Induktionsschritt: Zeige unter der Annahme, dass $P(n)$ gilt, auch $P(n+1)$ gilt. In unserem Fall müssen wir also zeigen, dass

$$P(n+1) = \sum_{i=0}^{n+1} i = \frac{(n+1)(n+2)}{2}$$

gilt unter der Vorraussetzung, dass

$$P(n) = \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

gilt.

Die Summe der Zahlen von 0 bis $n+1$ ist gleich der Summe der Zahlen von 1 bis n plus $n+1$, d.h. es ist

$$\sum_{i=0}^{n+1} i = \sum_{i=0}^{n} i + (n+1) \tag{3.3}$$

Es gilt nun:

$$\begin{aligned}
\sum_{i=0}^{n+1} i &= \sum_{i=0}^{n} i + (n+1) && \text{wegen (3.3)} \\
&= \frac{n(n+1)}{2} + (n+1) && \text{wegen der Induktionsannahme} \\
&= \frac{n(n+1) + 2(n+1)}{2} \\
&= \frac{(n+1)(n+2)}{2}
\end{aligned}$$

Damit haben wir den Induktionsschritt durchgeführt.

Insgesamt haben wir also gezeigt, dass unser Prädikat (3.2) für alle natürlichen Zahlen erfüllt ist.

b) Sei $x \in \mathbb{Z}$ mit $x \neq 1$, dann gilt, dass $x^n - 1$ für alle $n \in \mathbb{N}_0$ durch $x - 1$ teilbar ist.

Zunächst formulieren wir diese Behauptung als Prädikat:

$$P(n) = \forall n \in \mathbb{N}_0 : \frac{x^n - 1}{x - 1} \in \mathbb{Z} \tag{3.4}$$

Wir beweisen diese Behauptung durch vollständige Induktion:

Induktionsanfang: $P(0)$ gilt offensichtlich.

Induktionsschritt: Zeige unter der Annahme, dass $P(n)$ gilt, auch $P(n+1)$ gilt. In unserem Fall müssen wir also zeigen, dass

$$\frac{x^{n+1} - 1}{x - 1} \in \mathbb{Z} \tag{3.5}$$

gilt unter der Voraussetzung, dass

$$\frac{x^n - 1}{x - 1} \in \mathbb{Z} \tag{3.6}$$

gilt.

Wir formen (3.5) geeignet um:

$$\frac{x^{n+1} - 1}{x - 1} = \frac{x \cdot (x^n - 1) + (x - 1)}{x - 1} = x \cdot \frac{x^n - 1}{x - 1} + 1$$

Hieraus folgt mit der Induktionsvoraussetzung (3.6) und der Voraussetzung $x \in \mathbb{Z}$, dass (3.5) gilt.

Insgesamt haben wir also gezeigt, dass unser Prädikat (3.4) für alle natürlichen Zahlen erfüllt ist.

c) In Satz 1.19 auf Seite 74 haben wir bereits mithilfe von Bitvektoren bewiesen, dass, wenn eine Menge M m Elemente hat, die Potenzmenge $\mathcal{P}(M)$ von M 2^m Elemente hat. Jetzt geben wir einen weiteren Beweis mithilfe vollständiger Induktion an. Dazu formalisieren wir diese Behauptung als Prädikat:

$$P(m) = \forall m \in \mathbb{N}_0 : |M| = m \Rightarrow |\mathcal{P}(M)| = 2^m$$

Induktionsanfang: Zeige $P(0)$. Sei also $m = 0$, d.h. $M = \emptyset$ und damit $\mathcal{P}(M) = \{\emptyset\}$, woraus folgt: $|\mathcal{P}(M)| = 1$. Da $2^0 = 1$ ist, gilt also die Behauptung für $m = 0$.

Induktionsschritt: Zeige, dass $|M| = m + 1 \Rightarrow |\mathcal{P}(M)| = 2^{m+1}$ gilt, unter der Voraussetzung, dass $|\mathcal{P}(M)| = 2^m$ gilt für $|M| = m$.

Sei $M = \{a_1, a_2, \ldots, a_m, a_{m+1}\}$ sowie $M' = \{b_1, b_2, \ldots, b_m\}$, $b_i \in M$, $1 \leq i \leq m$, und $\{a\} = M - M'$. Nach Induktionsvoraussetzung ist $|\mathcal{P}(M')| = 2^m$.

Sei etwa $\mathcal{P}(M') = \{D_1, D_2, \ldots, D_{2^m}\}$. Alle $D_j \in \mathcal{P}(M')$, $1 \leq j \leq 2^m$, sind auch Elemente von $\mathcal{P}(M)$. Es fehlen noch die Elemente D_j, zu denen das Element a hinzugefügt wird: $D_j \cup \{a\}$, $1 \leq j \leq 2^m$. Insgesamt enthält $\mathcal{P}(M)$ also $2^m + 2^m = 2 \cdot 2^m = 2^{m+1}$ Elemente.

Damit ist der Induktionsschritt gezeigt.

d) Sei $x \in \mathbb{R}$, $x > -1$, $x \neq 0$, dann gilt: $\forall n \in \mathbb{N}_2 : (1+x)^n > 1 + nx$.[21]
Wir beweisen diese Behauptung mit vollständiger Induktion:

Induktionsanfang: Zeige $P(2)$. Für $n = 2$ gilt

$$\begin{aligned}(1+x)^2 &= 1 + 2x + x^2 \\ &> 1 + 2x && \text{da } x^2 > 0\end{aligned}$$

Induktionsschritt: Zeige, dass $(1+x)^{n+1} > 1+(n+1)x$ gilt, unter Voraussetzung, dass $(1+x)^n > 1 + nx$ gilt, für $x \in \mathbb{R}$, $x > -1$, $x \neq 0$. Es gilt:

$$\begin{aligned}(1+x)^{n+1} &= (1+x)^n(1+x) \\ &> (1+nx)(1+x) && \text{nach Induktionsvoraussetzung} \\ &= 1 + x + nx + nx^2 \\ &> 1 + nx + x && \text{da } nx^2 > x \\ &= 1 + (n+1)x\end{aligned}$$

e) Als letztes Beispiel wollen wir das Prädikat

$$P(n) = \forall n \in \mathbb{N}_7 : 2n^2 + 1 < 2^n$$

beweisen.

Induktionsanfang: Zeige $P(7)$. Es gilt $2 \cdot 7^2 + 1 = 99$ und $2^7 = 128$, und damit $P(7)$.

Induktionsschritt: Wir nehmen an, dass $P(n) = 2n^2 + 1 < 2^n$ gilt, und zeigen, dass dann auch $P(n+1) = 2(n+1)^2 + 1 < 2^{n+1}$ gilt.

Es ist

$$\begin{aligned}2(n+1)^2 + 1 &= 2(n^2 + 2n + 1) + 1 \\ &= 2n^2 + 1 + 4n + 2 \\ &< 2^n + 4n + 2 && \text{wegen der Induktionsannahme} \\ &< 2^n + 2^n && \text{da } 4n + 2 < 2^n \text{ für } n \geq 7 \text{ (siehe} \\ &= 2 \cdot 2^n = 2^{n+1} && \text{Übung 3.3 (1))}\end{aligned}$$

21 Diese Ungleichung wird als Bernoullische Ungleichung bezeichnet, benannt nach Jakob Bernoulli (1654 - 1705), einer der Mitglieder der berühmten schweizerischen Gelehrtenfamilie Bernoulli (niederländischer Herkunft), die im 17. und 18. Jahrhundert als Mathematiker, Physiker, Mediziner und Theologen hervorragende Beiträge zu diesen Wissenschaften geliefert haben.

Bemerkung 3.2 Bei den beiden letzten Beispielen ist der Induktionsanfang nicht $P(0)$, sondern $P(2)$ bzw. $P(7)$. Für kleinere Werte von n gilt die Behauptung nämlich nicht. Dies ist aber unerheblich und kein prinzipieller Verstoß gegen das Prinzip der Vollständigen Induktion, denn wir können jedes Prädikat $P(n)$, für welches der Induktionsanfang $P(c)$ für ein $c \in \mathbb{N}_0$ gilt, transformieren in ein äquivalentes Prädikat $P'(n)$, indem jedes Vorkommen von n durch $n + c$ ersetzt wird. In Beispiel d) wäre die transformierte Behauptung also $P'(n) = (1 + x)^{n+2} > 1 + (n + 2)x$, und in Beispiel e) wäre die transformierte Behauptung $P'(n) = 2(n + 7)^2 + 1 < 2^{n+7}$, und beide Behauptungen gelten dann für alle $n \in \mathbb{N}_0$. □

Übungsaufgaben

3.3 Beweisen Sie folgende Behauptungen:

(1) $\forall\, n \in \mathbb{N}$, $n \geq 5$: $2n + 1 \leq 2^{n-1}$.

(2) $\forall\, n \in \mathbb{N}$: $\sum_{i=1}^{n} 2(i-1) = n^2$.

(3) Sei $q \in \mathbb{R}$, $q \neq 1$, dann gilt $\forall n \in \mathbb{N}_0$: $\sum_{i=0}^{n} q^i = \frac{1-q^{n+1}}{1-q}$,

(4) Sei $x, y \in \mathbb{Z}$, dann gilt $\forall\, n \in \mathbb{N}_0$: $\frac{x^n - y^n}{x-y} \in \mathbb{Z}$.

(5) $\forall\, n \in \mathbb{N}$: $\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$.

(6) $\forall\, n \in \mathbb{N}$: $\sum_{i=1}^{2n} (-1)^{i+1} \cdot i^2 = \sum_{i=i}^{2n} i$.

(7) $\forall\, n \in \mathbb{N}$: $\sum_{i=1}^{n} i^3 = \left(\sum_{i=i}^{n} i\right)^2 = \left(\frac{n(n+1)}{2}\right)^2$.

(8) Die *Fermat-Zahlen* sind für $n \geq 0$ definiert durch $A_n = 2^{2^n} + 1$. Zeigen Sie, dass für alle $n \geq 1$ gilt: $\prod_{i=0}^{n-1} A_i = A_n - 2$.

(9) Gilt $\forall\, n \in \mathbb{N}_0$: $n^2 - n + 41 \in \mathbb{P}$? □

3.2.2 Verallgemeinertes Rekursionsschema

Wir haben schon im Kapitel 1.2 rekursive Definitionen benutzt. So haben wir im Abschnitt 1.2.2 die Syntax der Aussagenlogik definiert, indem wir zunächst – als Rekursionsanfang – atomare Formeln festgelegt haben und anschließend – im Rekursionsschritt – beschrieben, wie aus bereits gegebenen Formeln neue gebildet werden können. In der Definition 2.12 auf Seite 103 haben wir die n-fache Komposition einer Relation R mit sich selbst rekursiv definiert: Zunächst haben wir als Rekursionsanfang $R = id$ festgelegt und im Rekursionsschritt, wie R^n aus R^{n-1} berechnet wird.

Tatsächlich sind rekursive Definition und vollständige Induktion nicht auf auf natürlichen Zahlen definierte Funktionen beschränkt. Insbesondere in der Informatik sind viele wichtige Datenstrukturen, wie z.B. Listen und Bäume, rekursive Strukturen, auf denen Operationen wie Einfügen, Löschen und Suchen von Elementen entsprechend als rekursive Prozeduren programmiert werden können. Rekursion ist ein allgmeines Problemlöseprinzip, welches in vielen Fällen zu „eleganten“ Lösungen führt.

Verallgemeinertes Rekursionsschema

Beweise von Eigenschaften rekursiv definierter Strukturen könne mithilfe des folgenden verallgemeinerten Induktionsprinzips geführt werden: Sei M eine induktiv definierte Menge und $P : M \to \mathbb{B}$ ein totales Prädikat. Falls gilt:

- *Induktionsanfang:* Gilt $P(x)$ für alle explizit angegebenen Elemente von M und
- *Induktionsschritt:* gilt ür alle $x_1, \ldots, x_k \in M$ und für jedes daraus nach den Definitionsregeln von M erzeugbare $y \in M$: $P(x_1), \ldots, P(x_k) \Rightarrow P(y)$,

dann gilt für alle $x \in M$: $P(x)$. Haben wir für ein Prädikat $P : M \to \mathbb{B}$ Induktionsanfang und Induktionsschritt gezeigt, dann wissen wir, dass P für alle Elemente von M zutrifft.

Beispiel 3.4 Im Abschnitt 1.2.2 haben wir die Menge $\mathcal{A}$ der aussagenlogischen Formeln induktiv definiert. Wir beweisen nun das Prädikat P über dieser Menge, welches besagt, dass in jeder aussagenlogischen Formel, die Anzahl der öffnenden Klammern gleich der Anzahl der schließenden Klammern ist. Wir formalisieren zunächst dieses Prädikat. Für eine Formel $\alpha \in \mathcal{A}$ bezeichne $\alpha_($ die Anzahl der öffnenden und $\alpha_)$ die Anzahl der schließenden Klammern in α. Unser Prädikat lautet damit: $P(\alpha) = \alpha_(= \alpha_)$. Mit dem verallgemeinerten Induktionsschema zeigen wir nun, dass für alle $\alpha \in \mathcal{A}$ gilt $P(\alpha)$.

Induktionsanfang: Die explizit angegebenen Elemente von $\mathcal{A}$ sind die aussagenlogischen Konstanten 0 und 1 sowie die aussagenlogischen Variablen $v \in V$. Diese atomaren Formeln enthalten keine Klammern, d.h. es gilt

$$\begin{array}{lcl} 0_(& = & 0 = 0_) \\ 1_(& = & 0 = 1_) \\ v_(& = & 0 = v_), \text{ für alle } v \in V \end{array}$$

Somit gilt $P(0)$ und $P(1)$ sowie $P(v)$ für alle $v \in V$. Damit ist der Induktionsanfang für unser Prädikat P gezeigt.

Induktionsschritt: Für gegebene Formeln α,$\beta \in \mathcal{A}$ können aufgrund der Definitionsregeln von $\mathcal{A}$ die Formeln

(1) $\gamma_1 = (\alpha \wedge \beta)$,

(2) $\gamma_2 = (\alpha \vee \beta)$ sowie

(3) $\gamma_3 = \neg\alpha$

gebildet werden. Nach Induktionsvoraussetzung gelten $P(\alpha)$ und $P(\beta)$, d.h. $\alpha_(= \alpha_)$ bzw. $\beta_(= \beta_)$.

Wir zeigen nun, dass auch $P(\gamma_i)$, d.h. $\gamma_{i(} = \gamma_{i)}$, für $i = 1, 2, 3$, gilt:

Zu (1): Es gilt

$$\begin{aligned} \gamma_{1(} &= \alpha_(+ \beta_(+ 1 \\ &= \alpha_) + \beta_) + 1 \qquad \text{wegen Induktionsvoraussetzung} \\ &= \gamma_{1)} \end{aligned}$$

Zu (2): Der Beweis für diesen Fall ist identisch zum Fall (1).

Zu (3): Dieser Fall gilt offensichtlich, denn bei der Negation einer Formel kommen keine weiteren Klammern hinzu.

Damit ist auch der Induktionsschritt für alle Fälle gezeigt, und damit gilt gemäß dem verallgemeinerten Induktionsprinzip: für alle $\alpha \in \mathcal{A}$: $P(\alpha)$. □

Übungsaufgaben

3.4 (1) Für die aussagenlogischen Formeln gilt nicht nur, dass die Anzahl der öffnenden gleich der Anzahl der schließenden Klammern ist, sondern auch dass in jedem Präfix einer Formel die Anzahl der schließenden Klammern nicht größer als die Anzahl der öffnenden Klammern ist. Beweisen Sie diese Aussage!

(2) Die Menge $\mathcal{A}$ der *arithmetischen Ausdrücke* über $\mathbb{N}_0$ kann induktiv wie folgt definiert werden:

(i) $a \in \mathbb{N}_0 \Rightarrow a \in \mathcal{A}$: Jede natürliche Zahl a ist ein arithmetischer Ausdruck.

(ii) $x, y \in \mathcal{A} \Rightarrow (x + y), (x \cdot y) \in \mathcal{A}$: Sind x und y arithmetische Ausdrücke, dann auch $(x + y)$ und $(x \cdot y)$.

(iii) Genau die gemäß den Regeln (i) und (ii) bildbaren Ausdrücke gehören zu $\mathcal{A}$.

Beweisen Sie: In jedem Ausdruck von $\mathcal{A}$ ist die Anzahl von Zahlen aus $\mathbb{N}_0$ um 1 höher als die Anzahl der Operatorsymbole $+$ oder $\cdot$. □

3.2.3 Zusammenfassung

Vollständige Induktion und das verallgemeinerte Rekursionsschema sind Beweisprinzipien, mit denen Prädikate, die für die Menge der natürlichen Zahlen erfüllt sind, bzw. Prädikate, die für rekursiv definierte Strukturen erfüllt sind, bewiesen werden können. Beide Prinzipien bestehen aus zwei Schritten: Zunächst wird im Induktionsanfang bzeigt, dass die Behauptung für eine kleinste Zahl bzw. für alle explizit angegebenen Elemente einer induktiv definierten Menge

gelten. Unter der Annahme, dass die Behauptung für Elemente der Menge gilt, muss dann gezeigt werden, dass die Behauptung auch dann für alle Elemente gilt, die aus denen, für die die Behauptung angenommen wird, erzeugt werden, ebenfalls gilt.

3.3 Fibonacci-Zahlen

In diesem Kapitel wollen wir die Fibonacci-Folge betrachten. Diese taucht interessanterweise auf die eine oder andere Art bei vielen Phänomenen in Mathematik, Informatik und Naturwissenschaften aber auch in Bereichen der Kunst auf. Wir betrachten die Fibonacci-Folge hier nur als weiteres Beispiel einer rekursiv definierten Funktion, und wir werden nur ein paar wenige ihrer Eigenschaften betrachten.

Die Folge wird in der Regel auf das berühmte „Kaninchenproblem“ zurückgeführt. Dieses Problem wird im *Liber Abbaci* von dem wohl größten europäischen Mathematiker vor der Renaissance, *Leonardo Pisano* (*Leonardo von Pisa*, genannt *Leornado Fibonacci*, als Abkürzung von *Filius Bonaccii*, 1170 - 1240) als Übungsaufgabe gestellt:

> Die Vermehrung von Kaninchenpaaren geschehe wie folgt: Am Anfang gebe es kein Paar, nach dem ersten Monat sei ein Paar – durch wen auch immer geschaffen – da. Jedes Kaninchenpaar ist nach dem zweiten Monat seiner Existenz geschlechtsreif, und jedes Kaninchenpaar bringe von da an jeden Monat ein weiteres Paar zur Welt. Frage: Wie viele Kaninchenpaare F_n gibt es nach n Monaten?

Zum Zeitpunkt 0 gibt es kein Paar, also ist $F_0 = 0$. Nach einem Monat gibt es ein Paar P_1, also ist $F_1 = 1$. Nach zwei Monaten ist dieses Paar geschlechtsreif geworden, wir nennen es P_1^g, und es ist $F_2 = 1$. Nach drei Monaten hat P_1^g ein neues Paar P_2 gezeugt, so dass $F_3 = 2$ ist. Nach vier Monaten hat P_1^g ein weiteres Paar P_3 gezeugt, und P_2 ist geschlechtsreif geworden, was wir mit P_2^g notieren. Es ist $F_4 = 3$. Nach fünf Monaten haben P_1^g und P_2^g je ein Paar, P_4 und P_5, gezeugt, und P_3 ist geschlechtsreif geworden, was wir wieder entsprechend kennzeichnen: P_3^g. Es ist $F_5 = 5$.

Übungsaufgaben

3.5 Überlegen Sie selbst in dieser Art weiter! Sie werden herausfinden, dass $F_6 = 8$ und $F_7 = 13$ ist. □

Gibt es eine Gesetzmäßigkeit, mit der sich F_n für jedes $n \in \mathbb{N}_0$ berechnen lässt? Im n-ten Monat gibt es alle Paare, die es auch im vorhergehenden Monat $n-1$ gegeben hat, sowie so viele Paare, wie es im Monat $n-2$ gegeben hat, denn diese sind im Monat n alle geschlechtsreif und bekommen je ein Paar Nachwuchs. Aus dieser Überlegung ergibt sich die folgende rekursive Definition für die Fibonacci-Funktion

$$F : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$$

die für jeden Monat n die Anzahl der Kaninchenpaare F_n angibt:[22]

$$F_0 = 0, \;\; F_1 = 1, \;\; F_n = F_{n-1} + F_{n-2},\, n \geq 2 \tag{3.7}$$

Es gilt also z.B.

$$\begin{array}{rcrcrcrcr}
F_2 & = & F_1 & + & F_0 & = & 1 & + & 0 & = & 1 \\
F_3 & = & F_2 & + & F_1 & = & 1 & + & 1 & = & 2 \\
F_4 & = & F_3 & + & F_2 & = & 2 & + & 1 & = & 3 \\
F_5 & = & F_4 & + & F_3 & = & 3 & + & 2 & = & 5 \\
F_6 & = & F_5 & + & F_4 & = & 5 & + & 3 & = & 8 \\
F_7 & = & F_6 & + & F_5 & = & 8 & + & 5 & = & 13 \\
F_8 & = & F_7 & + & F_6 & = & 13 & + & 8 & = & 21 \\
F_9 & = & F_8 & + & F_7 & = & 21 & + & 13 & = & 34 \\
F_{10} & = & F_9 & + & F_8 & = & 34 & + & 21 & = & 55
\end{array}$$

Mithilfe analytischer Methoden aus der Theorie der Differenzengleichungen, auf die wir im Rahmen dieses Buches nicht eingehen, kann man einen geschlossenen Ausdruck zur Berechnung von F_n herleiten:

$$F_n = \frac{1}{\sqrt{5}}(\phi^n - \phi'^n) \tag{3.8}$$

Dabei ist

$$\begin{array}{rcrcr}
\phi & = & \frac{1}{2}(1+\sqrt{5}) & = & 1,6180339\ldots \\
\phi' & = & \frac{1}{2}(1-\sqrt{5}) & = & -0,6180339\ldots
\end{array}$$

Es gilt sogar:[23]

$$F_n = round\left(\frac{1}{\sqrt{5}}\phi^n\right)$$

Die Anzahl der Kaninchenpaare F_n wächst also exponentiell in der Anzahl der Monate n.

Goldener Schnitt

Die Zahl ϕ stellt den *Goldenen Schnitt* dar. Ein geometrisches Verhältnis von

22 Es ist üblich bei der Fibonacci-Funktion die eher bei Folgen übliche Schreibweise F_n anstelle der Funktionsschreibweise $F(n)$ zu verwenden.

23 *round* ist die bekannte Rundungsfunktion.

Größen in Kunst- oder Bauwerken galt von alters her (bei Euklid; in der Renaissance „göttliche Proportion“) als ästhetisch, falls es durch diese Zahl ausgedrückt werden konnte.[24]

Die Fibonacci-Zahlen können auch durch fortgesetzte Multiplikation der Matrix

$$\begin{pmatrix} F_0 & F_1 \\ F_1 & F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

berechnet werden. Dazu erklären wir zunächst, wie zwei 2×2-Matrizen miteinander multipliziert werden:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{pmatrix} \tag{3.9}$$

Es gilt nun für alle $n \geq 1$

$$\begin{pmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{pmatrix} = \begin{pmatrix} F_0 & F_1 \\ F_1 & F_2 \end{pmatrix}^n = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n$$

Wir beweisen dies durch vollständige Induktion über n:

Induktionsanfang: Für $n = 1$ gilt:

$$\begin{pmatrix} F_{1-1} & F_1 \\ F_1 & F_{1+1} \end{pmatrix} = \begin{pmatrix} F_0 & F_1 \\ F_1 & F_2 \end{pmatrix}^1 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

Induktionsschritt: Es gilt

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^{n+1} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{wegen Induktionsvoraussetzung}$$

$$= \begin{pmatrix} F_n & F_{n-1}+F_n \\ F_{n+1} & F_n+F_{n+1} \end{pmatrix} \quad \text{wegen (3.9)}$$

$$= \begin{pmatrix} F_n & F_{n+1} \\ F_{n+1} & Fn+ \end{pmatrix} \quad \text{wegen (3.7)}$$

24 Eine Strecke AB wird durch einen Teilpunkt C gemäß dem Goldenen Schnitt geteilt, falls gilt $\frac{|AC|}{|CB|} = \frac{|CB|}{|AB|}$. Dabei bedeutet $|XY|$ die Länge der Strecke $\overline{XY}$. Wenn man $|AC| = 1$ und $|CB| = x$ setzt, dann wird das Streckenverhältnis durch $\frac{1}{x} = \frac{x}{1+x}$, umgeformt durch $x^2-x-1 = 0$ ausgedrückt. Lösen wir diese Gleichung auf, dann erhalten wir als Lösungen ϕ und ϕ'.

Übungsaufgaben

3.6 Zeigen Sie, dass für die Folge der Fibonacci-Zahlen folgende Gleichungen gelten:

(i) $\forall\, n \in \mathbb{N}$: $\sum_{i=1}^{n} F_{2i-1} = F_{2n}$,

(ii) $\forall\, n \in \mathbb{N}_0$: $\sum_{i=0}^{n} F_{2i} = F_{2n+1} - 1$,

(iii) $\forall\, n \in \mathbb{N}$: $1 + \sum_{i=1}^{n} F_i = F_{n+2}$,

(iv) $\forall\, n \in \mathbb{N}$: $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$.

3.4 Ackermannfunktion

Als weitere Beispiele zu rekursiv definierten Funktionen betrachten wir die Ackermannfunktion und Varianten davon. Die Ackermannfunktion ist von großer Bedeutung in der Berechenbarkeitstheorie (siehe Kapitel 4.3). Sie ist eine Funktion, die mit Zählschleifen nicht berechnet werden kann, weil sie schneller als alle Funktionen wächst, die mit Zählschleifen berechnet werden können. Einen Eindruck dieses Wachstums geben die folgenden Betrachtungen.

Ackermann-funktion

Definition 3.1 Die *Ackermannfunktion* $A : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ ist definiert durch

$$A(x,y) = \begin{cases} y+1, & x = 0 \\ A(x-1,1), & x \neq 0 \wedge y = 0 \\ A(x-1, A(x,y-1)), & \text{sonst} \end{cases}$$

Übungsaufgaben

3.7 Berechnen Sie, um einen ersten Eindruck von dieser Funktion zu bekommen, $A(3,3)$! □

Folgerung 3.1 Für alle $n \geq 0$ gilt: **a)** $A(1,n) = n+2$, **b)** $A(2,n) = 2n+3$,

c) $A(3,n) = 2^{n+3} - 3$, **d)** $A(4,n) = \underbrace{2^{2^{\cdot^{\cdot^{2}}}}}_{n+3 \text{ Zweien}} - 3$. □

Übungsaufgaben

3.8 Beweisen Sie Folgerung 3.1! □

Wir definieren nun die Familie $\{ A_i \}_{i\geq 0}$ von Funktionen $A_i : \mathbb{N}_0 \to \mathbb{N}_0$ rekursiv wie folgt:

$$A_0(x) = \begin{cases} 1, & x = 0 \\ 2, & x = 1 \\ x + 2, & \text{sonst} \end{cases}$$

$$A_{n+1}(x) = A_n^x(1) = \underbrace{A_n(A_n(\ldots A_n}_{x\text{-mal}}(1)\ldots))$$

Und mithilfe dieser Funktionen legen wir die Funktion

$$ack : \mathbb{N}_0 \to \mathbb{N}_0$$

fest:

$$ack(x) = A_x(x) \tag{3.10}$$

Gemäß diesen Definitionen gilt z.B.:

$$\begin{aligned}
ack(0) &= A_0(0) = 1 \\
ack(1) &= A_1(1) = A_0^1(1) = 2 \\
ack(2) &= A_2(2) = A_1^2(1) = A_1(A_1(1)) = A_1(2) \\
&= A_0^2(1) = A_0(A_0(1)) = A_0(2) = 2 + 2 = 4 \\
ack(3) &= A_3(3) = A_2^3(1) = A_2(A_2(A_2(1))) = A_2(A_2(2)) = A_2(4) \\
&= A_1^4(1) = A_1(A_1(A_1(A_1(1)))) = A_1(A_1(A_1(2))) = A_1(A_1(4)) \\
&= A_1(A_1(4)) = A_1(A_0^4(1)) = A_1(A_0(A_0(A_0(A_0(1))))) \\
&= A_1(A_0(A_0(A_0(2)))) = A_1(A_0(A_0(4))) = A_1(A_0(6)) \\
&= A_1(8) = \ldots \\
&\vdots \\
&= 16
\end{aligned}$$

Zur Berechnung von $ack(4)$ bestimmen wir zunächst (1) $A_2(x)$ und dann (2) $A_3(x)$.

Zu (1): Es gilt:

$$A_2(x) = 2^x \text{ für alle } x \in \mathbb{N}_0 \tag{3.11}$$

Wir beweisen dies mit vollständiger Induktion:

Induktionsanfang: Es gilt: $A_2(0) = A_1^0(1) = 1 = 2^0$. Für $x = 0$ stimmt also die Behauptung.

Induktionsschritt: Es gilt

$$\begin{aligned}
A_2(x+1) &= A_1^{x+1}(1) \\
&= A_1(A_1^x(1)) \\
&= A_1(A_2(x)) \\
&= A_1(2^x) && \text{nach Induktionsvoraussetzung} \\
&= A_0^{2^x}(1) \\
&= \underbrace{A_0(A_0(\ldots A_0(A_0}_{2^x\text{-mal}}(1))\ldots)) \\
&= \underbrace{A_0(A_0(\ldots A_0}_{(2^x-1)\text{-mal}}(2)\ldots)) \\
&= \underbrace{2+2+\ldots+2}_{2^x\text{-mal}} \\
&= 2 \cdot 2^x = 2^{x+1}
\end{aligned}$$

womit der Induktionsschritt vollzogen ist.

Zu (2): Die x-mal iterierte Zweierpotenz ist die Funktion $iter_2 : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$\begin{aligned}
iter_2(0) &= 1 \\
iter_2(n+1) &= 2^{iter_2(n)},\ n \geq 1
\end{aligned}$$

Es gilt also z.B.

$$iter_2(1) = 2^{iter_2(0)} = 2^1 = 2 = 2 \tag{3.12}$$

$$iter_2(2) = 2^{iter_2(1)} = 2^2 = 2^2 = 4 \tag{3.13}$$

$$iter_2(3) = 2^{iter_2(2)} = 2^4 = 2^{2^2} = 16 \tag{3.14}$$

$$iter_2(4) = 2^{iter_2(3)} = 2^{16} = 2^{2^{2^2}} = 65\,536 \tag{3.15}$$

$$iter_2(5) = 2^{iter_2(4)} = 2^{65\,536} = 2^{2^{2^{2^2}}} = \ldots \tag{3.16}$$

Man erkennt, dass $iter_2(x)$ die x-mal iterierte 2er-Potenz ist.

Wir beweisen nun mit vollständiger Induktion, dass gilt

$$A_3(x) = iter_2(x) \tag{3.17}$$

Induktionsanfang: Es gilt: $A_3(0) = A_2^0(1) = 1 = iter_2(0)$. Für $x = 0$ stimmt also die Behauptung.

Induktionsschritt: Es gilt

$$\begin{aligned}
A_3(x+1) &= A_2^{x+1}(1) \\
&= A_2(A_2^x(1)) \\
&= A_2(A_3(x)) \\
&= A_2(iter_2(x)) && \text{nach Induktionsvoraussetzung} \\
&= 2^{iter_2(x)} && \text{wegen (3.11)} \\
&= iter_2(x+1) && \text{wegen der Definition von } iter_2
\end{aligned}$$

womit der Induktionsschritt vollzogen ist.

Nun können wir $ack(4)$ berechnen:

$$\begin{aligned}
ack(4) &= A_4(4) \\
&= A_3^4(1) \\
&= A_3^3(A_3(1)) \\
&= A_3^3(2) && \text{wegen (3.17) und (3.12)} \\
&= A_3^2(A_3(2)) \\
&= A_3^2(4)) && \text{wegen (3.17) und (3.13)} \\
&= A_3(A_3(4)) \\
&= A_3(65536) && \text{wegen (3.17) und (3.15)} \\
&= iter_2(65536) \\
&= 2^{2^{2^{\cdot^{\cdot^{\cdot^2}}}}} && \text{65536-mal iterierte Zweierpotenz}
\end{aligned}$$

Man schätzt die Anzahl der Atome im „bekannten“ Universum auf etwa 2^{350}, im Vergleich zu $ack(4)$ eine sehr kleine Zahl. Die Ackermannfunktion hat ein immenses Wachstumsverhalten. Man kann sogar zeigen, dass die Ackermannfunktion stärker wächst als Funktionen, die mit Zählschleifen berechnet werden können. Bei Zählschleifen liegt die Anzahl der Schleifendurchläufe vor Ausführung der Schleife fest, d.h. die Schleifenbedingung kann – im Unterschied zu Bedingungsschleifen – während der Schleifenausführung nicht verändert werden.

3.5 Abzählbarkeit von Mengen

Die Menge der natürlichen Zahlen ist ja gerade zum Abzählen geschaffen worden. Insofern erscheint es als „natürlich“ diese Menge als Referenzmenge für abzählbare Mengen zu wählen.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- den Begriff der Abzählbarkeit und seine grundlegenden Eigenschaften erklären können,
- die Abzählbarkeit von Mengen beweisen können,
- das Prinzip der Diagonalisierung verstehen und anwenden können.

3.5.1 Definitionen und grundlegende Eigenschaften

Wir wollen eine Menge abzählbar nennen, falls ihre Elemente eindeutig mit natürlichen Zahlen nummeriert werden können.

Abzählbarkeit

Definition 3.2 **a)** Eine Menge M heißt *abzählbar* genau dann, wenn sie endlich ist oder wenn $|M| = |\mathbb{N}_0|$ gilt.

Überabzählbarkeit

b) Ist eine Menge nicht abzählbar, dann nennen wir sie *überabzählbar*. □

Eine unendliche abzählbare Menge M ist *gleichmächtig* zur Menge der natürlichen Zahlen, denn jedem Element aus M kann genau eine natürliche Zahl und jeder natürlichen Zahl kann genau ein Element aus M zugeordnet werden. Ist $f : M \to \mathbb{N}_0$ die Abzählung mit $f(m) = i$, dann schreiben wir auch m_i. Die Elemente aus M besitzen eine eindeutige Nummer, können also abgezählt werden: $m_0, m_1, m_2, \ldots$

Folgerung 3.2 A und B seien Mengen.

a) Ist $A \subseteq B$ und ist B abzählbar, dann ist auch A abzählbar: Jede Teilmenge einer abzählbaren Menge ist abzählbar.

b) Ist $A \subseteq B$ und ist A überabzählbar, dann ist auch B überabzählbar: Jede Obermenge einer nicht abzählbaren Menge ist nicht abzählbar. □

3.5.2 Beispiele und Diagonalisierung

Beispiel 3.5 **a)** Aus den Beispielen 2.18 (Seite 110) und 2.19 (Seite 114) wissen wir, dass das kartesische Produkt $\mathbb{N}_0^k$ für jedes $k \in \mathbb{N}$, die Menge $\mathbb{Z}$ der ganzen Zahlen und die Menge $\mathbb{Q}$ der rationalen Zahlen abzählbar sind, da sie gleichmächtig zur Menge der natürlichen Zahlen sind, bzw. dass die Menge $\mathbb{R}$ der reellen Zahlen, jedes Intervall von reellen Zahlen sowie die Potenzmenge $\mathcal{P}(\mathbb{N})$ der natürlichen Zahlen überabzählbar sind, da sie mächtiger als die Menge der natürlichen Zahlen sind.

b) Die Menge $\mathbb{N}^{\mathbb{N}} = \{\varphi : \mathbb{N} \to \mathbb{N} \mid \varphi \text{ total}\}$ der totalen Funktionen der Menge der natürlichen Zahlen in sich ist überabzählbar.

Wir nehmen an, dass $\mathbb{N}^{\mathbb{N}}$ abzählbar ist, d.h. es gibt eine bijektive Abbildung $f : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, die $\mathbb{N}^{\mathbb{N}}$ abzählt. $\varphi_1, \varphi_2, \varphi_3 \ldots$ sei die durch f festgelegte Abzählung von $\mathbb{N}^{\mathbb{N}}$. $\mathbb{N}$ ist abzählbar, und $x_1, x_2, x_3 \ldots$ sei eine solche Abzählung. Mit den beiden Abzählungen können wir folgende Matrix betrachten:

	x_1	x_2	x_3	$\ldots$	x_j	$\ldots$
φ_1	$\varphi_1(x_1)$	$\varphi_1(x_2)$	$\varphi_1(x_3)$	$\ldots$	$\varphi_1(x_j)$	$\ldots$
φ_2	$\varphi_2(x_1)$	$\varphi_2(x_2)$	$\varphi_2(x_3)$	$\ldots$	$\varphi_2(x_j)$	$\ldots$
φ_3	$\varphi_3(x_1)$	$\varphi_3(x_2)$	$\varphi_3(x_3)$	$\ldots$	$\varphi_3(x_j)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
φ_i	$\varphi_i(x_1)$	$\varphi_i(x_2)$	$\varphi_i(x_3)$	$\ldots$	$\varphi_i(x_j)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Da alle Funktionen φ_i total sind, ist diese Tabelle komplett ausgefüllt.

Diagonalisierung

Wir definieren mithilfe der Diagonalen dieser Matrix die Funktion φ_D wie folgt:

$$\varphi_D(x_k) = \varphi_k(x_k) + 1 \tag{3.18}$$

Die Funktion φ_D ist so konstruiert, dass sie sich mindestens für ein Argument, nämlich jeweils für das entsprechende Element in der Diagonale der Matrix, von jeder Funktion φ_k unterscheidet. Das führt im Folgenden zu einer widersprüchlichen Aussage.

Zunächst stellen wir fest, dass φ_D eine totale Funktion ist, d.h. es muss $\varphi_D \in \mathbb{N}^{\mathbb{N}}$ sein. Das bedeutet aber, dass φ_D in der Abzählung $\varphi_1, \varphi_2, \varphi_3, \ldots$ von $\mathbb{N}^{\mathbb{N}}$ vorkommen muss. Damit muss es eine Nummer s geben mit $\varphi_D = \varphi_s$. Für diese Beziehungen gilt

$$\varphi_s(x_s) = \varphi_D(x_s) = \varphi_s(x_s) + 1$$

was offensichtlich einen Widerspruch darstellt. Unsere Annahme muss also falsch sein. Damit haben wir bewiesen, dass $\mathbb{N}^{\mathbb{N}}$ überabzählbar ist. □

Das Beweisprinzip, das wir im letzten Beispiel angewendet haben, heißt *Diagonalisierung*; es geht auf Cantor zurück.

Übungsaufgaben

3.9 Wir wissen bereits, dass die Potenzmenge $\mathcal{P}(\mathbb{N})$ der natürlichen Zahlen überabzählbar ist, weil wir in Beispiel 2.19 gezeigt haben, dass sie gleichmächtig zur Menge $\mathbb{R}$ der reellen Zahlen ist. Zeigen Sie mithilfe einer Diagonalisierung, dass $\mathcal{P}(\mathbb{N})$ überabzählbar ist! □

3.5.3 Abschlusseigenschaften abzählbarer Mengen

Wir wollen nun die Abschlusseigenschaften der Klasse der abzählbaren Mengen gegenüber den gängigen Mengenverknüpfungen betrachten. Dass das kartesische

	1		2		3		4	...
A_1	a_{11}	$\swarrow$	a_{12}	$\swarrow$	a_{13}	$\swarrow$	a_{14}	...
A_2	a_{21}	$\swarrow$	a_{22}	$\swarrow$	a_{23}	...		
A_3	a_{31}	$\swarrow$	a_{32}	...				
A_4	a_{41}	...						
$\vdots$								

Abb. 3: Abzählung von $\bigcup_{i\in I} A_i$

Produkt von endlichen vielen abzählbaren Mengen abzählbar ist, folgt unmittelbar aus der Tatsache, dass das kartesische Produkt $\mathbb{N}_0^k$ für jedes $k \in \mathbb{N}$ abzählbar ist.

Satz 3.2 **a)** Es seien A und B abzählbare Mengen, dann sind auch (1) $A \cap B$, (2) $A - B$ und (3) $A \cup B$ abzählbar.

b) Es sei I eine unendliche, abzählbare (Index-) Menge, und die Mengen A_i, $i \in I$, seien ebenfalls abzählbar, dann ist auch $\bigcup_{i\in I} A_i$ abzählbar.

Beweis **a)** (1) und (2) folgen unmittelbar aus den Eigenschaften $A \cap B \subseteq A$ bzw. $A - B \subseteq A$ mithilfe Folgerung 3.2 a). Zu (3) gehen wir von einer Bijektion $f : A \to \mathbb{N}_0$ und einer Bijektion $g : B \to \mathbb{N}$ aus. Damit definieren wir $h : A \cup B \to \mathbb{Z}$ durch

$$h(x) = \begin{cases} f(x), & x \in A \\ -g(x), & x \in B - A \end{cases}$$

h ist somit eine totale, injektive Funktion von $A \cup B$ nach $\mathbb{Z}$, womit $|A \cup B| \leq |\mathbb{Z}|$ und damit die Azählbarkeit von $A \cup B$ folgt.

b) Da die Menge I unendlich und abzählbar ist, können wir als Indexmenge auch die Menge $\mathbb{N}$ wählen. Die Mengen A_i sind abzählbar, so können wir ihre Elemente entsprechend der entsprechenden Abzählung wie folgt aufschreiben: $a_{i1}, a_{i2}, a_{i3}, \ldots$ etwa realisiert durch die Bijektion $\varphi_i : \mathbb{N} \to A_i$ definiert durch $\varphi_i(j) = a_{ij}$. Wir nehmen uns die Matrix 2 von Seite 111 zur Hilfe und tragen die Elemente der Mengen A_i gemäß den Funktionswerten $\varphi_i(j)$ in dieses Schema ein, womit sich die Matrix in Abbildung 3 ergibt. Die Abzählung in dieser Tabelle veranschaulicht so die Bijektion $\varphi : \mathbb{N} \to \bigcup_{i\in I} A_i$ definiert durch $\varphi(a_{ij}) = c_2(i, j)$. □

3.5.4 Zusammenfassung

Die Menge der natürlichen Zahlen ist die Referenzmenge für die Abzählbarkeit von Mengen. Jede Menge, die gleichmächtig zur Menge der natürlichen Zahlen (oder zu einer Teilmenge natürlicher Zahlen) ist, d.h. die bijektiv auf die Menge der natürlichen Zahlen (bzw. auf eine Teilmenge natürlicher Zahlen) abgebildet werden kann, ist abzählbar. Die Elemente einer abzählbaren Menge können eineindeutig nummeriert werden. Mithilfe einer Diagonalisierung kann gezeigt werden, dass eine Menge nicht abzählbar ist.

3.6 Die Menge der ganzen Zahlen

Die Subtraktion $x - y$ ist in $\mathbb{N}_0$ nur dann möglich ist, falls $x \geq y$ ist. Wir werden nun eine neue Zahlenmenge einführen, in der wir sowohl die Menge der natürlichen Zahlen und das in dieser Menge mögliche Rechnen „wiederfinden“ werden, in der wir aber auch die Subtraktion ohne Einschränkung durchführen können. Des Weiteren werden wir die Rechenregeln für diese neuen Zahlen auflisten.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- verstehen, wie die ganzen Zahlen und das Rechnen mit diesen mithilfe einer geeigneten Äquivalenzrelation auf den natürlichen Zahlen eingeführt werden können,
- die Rechenregeln für ganze Zahlen kennen und anwenden können.

3.6.1 Konstruktion der ganzen Zahlen

Wir werden die Menge der ganzen Zahlen $\mathbb{Z}$ mithilfe einer Äquivalenzrelation über Paaren von natürlichen Zahlen konstruieren.

Die Relation $\mathcal{Z} \subseteq (\mathbb{N} \times \mathbb{N}) \times (\mathbb{N} \times \mathbb{N})$ sei definiert durch:

$$(a, b)\, \mathcal{Z}\, (c, d) \text{ genau dann, wenn } a + d = b + c \tag{3.19}$$

Die Relation $\mathcal{Z}$ verwendet nur bekannte Begriffe: natürliche Zahlen bzw. Paare von natürlichen Zahlen sowie die Addition von natürlichen Zahlen.

Wir zeigen nun, dass $\mathcal{Z}$ eine Äquivalenzrelation ist:

$\mathcal{Z}$ ist reflexiv: Für alle $(a, b) \in \mathbb{N} \times \mathbb{N}$ gilt $a + b = b + a$, woraus mit (3.19) folgt, dass für alle $(a, b) \in \mathbb{N} \times \mathbb{N}$ gilt: $(a, b)\, \mathcal{Z}\, (a, b)$.

$\mathcal{Z}$ ist symmetrisch: Sei $(a, b)\, \mathcal{Z}\, (c, d)$, d.h. $a + d = b + c$. Mit den bekannten Rechenregeln folgt daraus, dass $c + b = d + a$ gilt, und hieraus mit (3.19), dass $(c, d)\, \mathcal{Z}\, (a, b)$ gilt.

$\mathcal{Z}$ ist transititv: Sei $(a,b)\,\mathcal{Z}\,(c,d)$ und $(c,d)\,\mathcal{Z}\,(e,f)$, d.h. es ist $a+d=b+c$ und $c+f=d+e$. Hieraus folgt, dass $(a+d)+(c+f)=(b+c)+(d+e)$ ist, und daraus folgt, dass $a+f=b+e$ ist. Mit (3.19) folgt $(a,b)\,\mathcal{Z}\,(e,f)$.

Wir wollen nun die Äquivalenzklassen von $\mathcal{Z}$ bestimmen. Es gilt beispielsweise

$$\begin{aligned}[(7,3)]_{\mathcal{Z}} &= \{\,(7,3),(5,1),(6,2),(8,4),\dots\,\}\\ [(2,2)]_{\mathcal{Z}} &= \{\,(2,2),(1,1),(3,3),(4,4),\dots\,\}\\ [(1,3)]_{\mathcal{Z}} &= \{\,(2,4),(3,5),(4,6),(5,7),\dots\,\}\end{aligned}$$

Konstruktion von $\mathbb{Z}$

Wir wollen als Repräsentant jeder Äquivalenzklasse das Paar wählen, welches die Zahl 1 mindestens in einer Komponente enthält. Dann gilt allgemein für $a,b\in\mathbb{N}$ mit $a,b\geq 2$

$$[(a,1)]_{\mathcal{Z}} = \{\,(c,d)\,|\,c=d+(a-1)\,\} \tag{3.20}$$
$$[(1,1)]_{\mathcal{Z}} = \{\,(x,x)\,|\,x\in\mathbb{N}\,\} \tag{3.21}$$
$$[(1,b)]_{\mathcal{Z}} = \{\,(c,d)\,|\,c+(b-1)=d\,\} \tag{3.22}$$

was sich leicht mithilfe von (3.19) nachrechnen lässt. Dabei ist mit $a-1$ der Vorgänger von a und mit $b-1$ der Vorgänger von b gemeint (da $a,b\geq 2$ ist, existieren diese Vorgänger).

Rechenoperationen für ganze Zahlen

Auf der Menge der Äquivalenzklassen, die wir mit $(\mathbb{N}\times\mathbb{N})/\mathcal{Z}$ bezeichnen, führen wir Rechenoperationen, nämlich die Addition $\oplus$, die Subtraktion $\ominus$ sowie die Multiplikation $\odot$ ein:

$$[(a,b)]_{\mathcal{Z}}\oplus[(c,d)]_{\mathcal{Z}} = [(a+c,b+d)]_{\mathcal{Z}} \tag{3.23}$$
$$[(a,b)]_{\mathcal{Z}}\ominus[(c,d)]_{\mathcal{Z}} = [(a+d,b+c)]_{\mathcal{Z}} \tag{3.24}$$
$$[(a,b)]_{\mathcal{Z}}\odot[(c,d)]_{\mathcal{Z}} = [(ac+bd,bc+ad)]_{\mathcal{Z}} \tag{3.25}$$

Es sei darauf hingewiesen, dass diese Rechenoperationen ausschließlich mit natürlichen Zahlen und den dafür vollständig definierten Operationen Addition $+$ und Multiplikation $\cdot$ definiert sind.

Bevor wir die neu definierten Rechenoperationen weiter betrachten, wollen wir zur Vereinfachung den Äquivalenzklassen Namen geben. Da $a,b\in\mathbb{N}$ mit $a,b\geq 2$ ist, gilt $a-1,b-1\in\mathbb{N}$. Damit führen wir folgende Bezeichnungen ein:

$$\begin{aligned}\overline{a-1} &= [(a,1)]_{\mathcal{Z}}\\ 0 &= [(1,1)]_{\mathcal{Z}}\\ \underline{b-1} &= [(1,b)]_{\mathcal{Z}}\end{aligned} \tag{3.26}$$

Es ist also z.B. $\overline{4}=[(5,1)]_{\mathcal{Z}}$ und $\underline{2}=[(1,3)]_{\mathcal{Z}}$.

Ganze Zahlen

Wir wollen nun diese Bezeichner für die Äquivalenzklassen *ganze Zahlen* nennen und die Menge der ganzen Zahlen mit $\mathbb{Z}$ bezeichnen. Es ist also

$$\mathbb{Z} = \{\,\dots,\underline{3},\underline{2},\underline{1},0,\overline{1},\overline{2},\overline{3},\dots\,\}$$

Wir haben mithilfe der natürlichen Zahlen, mit den auf diesen vollständig definierten Operationen $+$ und $\cdot$ und mit der Äquivalenzrelation $\mathcal{Z}$ die Zahlenmenge $\mathbb{Z}$ mit vollständig auf ihr definierten Operationen $\oplus$, $\ominus$ und $\odot$ geschaffen. In dieser neuen Rechenstruktur wollen wir ein paar Beispiele rechnen:

$$\begin{array}{lllll}
[(7,3)]_{\mathcal{Z}} & \oplus & [(9,2)]_{\mathcal{Z}} & = & [(16,5)]_{\mathcal{Z}} \quad \text{bzw.} \quad \overline{4} \oplus \overline{7} = \overline{11} \\
[(7,3)]_{\mathcal{Z}} & \ominus & [(9,2)]_{\mathcal{Z}} & = & [(9,12)]_{\mathcal{Z}} \quad \text{bzw.} \quad \overline{4} \ominus \overline{7} = \underline{3} \\
[(7,3)]_{\mathcal{Z}} & \oplus & [(3,7)]_{\mathcal{Z}} & = & [(10,10)]_{\mathcal{Z}} \quad \text{bzw.} \quad \overline{4} \oplus \underline{4} = 0 \\
[(7,3)]_{\mathcal{Z}} & \odot & [(9,2)]_{\mathcal{Z}} & = & [(69,41)]_{\mathcal{Z}} \quad \text{bzw.} \quad \overline{4} \odot \overline{7} = \overline{28} \\
[(7,3)]_{\mathcal{Z}} & \odot & [(3,7)]_{\mathcal{Z}} & = & [(42,58)]_{\mathcal{Z}} \quad \text{bzw.} \quad \overline{4} \odot \underline{4} = \underline{16}
\end{array}$$

Nun, spätestens an diesen Beispielen bemerken wir, dass es sich um die uns bekannten ganzen Zahlen und für diese definierte Rechenoperationen Addition, Subtraktion und Multiplikation handelt. Wenn wir x anstelle von $\overline{x}$, $-x$ anstelle von $\underline{x}$ notieren und diese Zahlen positive ganze Zahlen bzw. negative ganze Zahlen nennen, und außerdem $+$ für $\oplus$, $-$ für $\ominus$ sowie $\cdot$ für $\odot$ schreiben, benutzen wir die uns aus der Schule vertrauten Schreibweisen und Bezeichnungen für das Rechnen mit ganzen Zahlen.

Abgeschlossenheit von Addition, Multiplikation und Subtraktion

Wir können zudem feststellen, dass die natürlichen Zahlen den positiven Zahlen entsprechen und dass wir den Zahlenraum der natürlichen Zahlen um die 0 und die negativen ganzen Zahlen erweitert haben. In diesem Zahlenraum ist nun – im Gegensatz zu den natürlichen Zahlen – die Subtraktion uneingeschränkt ausführbar.

$\mathbb{Z}$ als Erweiterung von $\mathbb{N}$

Dass die ganzen Zahlen eine Erweiterung der natürlichen Zahlen darstellen, d.h. dass die positiven ganzen Zahlen mit der auf ihnen definierten Addition und Multiplikation den natürlichen Zahlen mit ihrer Addition und Multiplikation entsprechen, also – mathematisch gesprochen – die Strukturgleichheit dieser beiden Rechenstrukturen gilt, kann wieder mithilfe eines Isomorphismus beschrieben werden. Dazu sei

$$\mathbb{Z}_+ = \{\, [(a,1)]_{\mathcal{Z}} \mid a \in \mathbb{N},\ a \geq 1 \,\}$$

die Menge der positiven ganzen Zahlen einschließlich der Null (dargestellt in der ursprünglichen Notation (3.26) als Äquivalenzklassen, um sie von der Notation der natürlichen Zahlen zu unterscheiden). Dann gelten für die Abbildung

$$\varphi : \mathbb{Z}_+ \to \mathbb{N}_0$$

definiert durch

$$\varphi([(a,1)]_{\mathcal{Z}}) = a - 1 \tag{3.27}$$

die Eigenschaften: φ ist bijektiv und genügt den Strukturgleichungen

$$\varphi\left([(a,1)]_{\mathcal{Z}} \oplus [(b,1)]_{\mathcal{Z}}\right) = \varphi\left([(a,1)]_{\mathcal{Z}}\right) + \varphi\left([(b,1)]_{\mathcal{Z}}\right) \tag{3.28}$$

sowie

$$\varphi\left([(a,1)]_{\mathcal{Z}} \otimes [(b,1)]_{\mathcal{Z}}\right) = \varphi\left([(a,1)]_{\mathcal{Z}}\right) \cdot \varphi\left([(b,1)]_{\mathcal{Z}}\right) \tag{3.29}$$

Übungsaufgaben

3.10 Beweisen Sie, dass die oben definierte Abbildung φ bijektiv ist und dass sie die Strukturgleichungen (3.28) und (3.29) erfüllt! □

Isomorphismus

Ein Isomorphismus kann als eineindeutige Umbenennung der Elemente einer Rechenstruktur mit Elementen einer anderen Rechenstruktur aufgefasst werden, wobei diese Umbenennung mit den Rechenoperationen verträglich sein muss. Diese Verträglichkeit wird durch die Strukturgleichungen für die einzelnen Operationen beschrieben. Bis auf die Benennung der Elemente handelt es sich quasi um dieselben Rechenstrukturen.

In unserem Beispiel werden die positiven ganzen Zahlen, d.h. die Äquivalenzklassen $[a,1]_{\mathcal{Z}}$ von $\mathcal{Z}$ mit $a \geq 1$ eineindeutig den natürlichen Zahlen $a-1$ zugeordnet, so dass die Umbenennung der Summe (des Produktes) von zwei ganzen Zahlen gleich der Summe (Produkte) der Umbenennungen, d.h. gleich der Summe (dem Produkt) der entsprechenden natürlichen Zahlen, ist.

3.6.2 Rechenregeln in $\mathbb{Z}$

Neben den Regeln A1, A2, A3, M1, M2 und M3 (siehe Abschnitt 3.1.3), die in $\mathbb{Z}$ ebenfalls gelten, gilt für $x, y \in \mathbb{Z}$ noch:

Additives Inverses

(A4) *Additives Inverses:* $x + y = 0$. Dabei gilt $y = -x$, und $-x$ heißt die zu x bezüglich der Addition *inverse* oder die zu x *negative Zahl*.[25]

Die Definition der Relation $\leq$ kann ebenfalls übernommen werden: $\leq \subseteq \mathbb{Z} \times \mathbb{Z}$ definiert durch $x \leq y$ genau dann, wenn $\exists\, d \in \mathbb{N}_0 : x + d = y$ definiert eine totale Ordnung auf $\mathbb{Z}$.

Monotonieregeln

Für diese Ordnung gilt ebenfalls die Regel O1, während bei der Regel O2 unterschieden werden muss, ob der Faktor positiv oder negativ ist:

(O2+) $\forall\, x, y, z \in \mathbb{Z}$ mit $x \leq y$ und $z \geq 1$ gilt $xz \leq yz$,

(O2−) $\forall\, x, y, z \in \mathbb{Z}$ mit $x \leq y$ und $z \leq -1$ gilt $xz \geq yz$.

Absolutbetrag

Der *Absolutbetrag* einer ganzen Zahl ist festgelegt durch die Funktion

$$|\ |\ : \mathbb{Z} \to \mathbb{N}_0$$

25 Formal betrachtet hat das Vorzeichen „−“ von x eine andere Bedeutung als der Operator „−“ in einer Subtraktion $a - b$ zweier ganzer Zahlen a und b. Dies haben wir bei Einführung der ganzen Zahlen durch die Notationen $\underline{x}$ bzw. $\ominus$ auch ganz bewusst unterschieden. Es gilt allerdings $a + (-b) = a - b$ oder genauer $a \oplus \underline{b} = a \ominus \overline{b}$, weswegen wir beim Rechnen nicht zwischen dem Operator „−“ und dem Vorzeichen „−“ streng unterscheiden müssen.

definiert durch

$$|x| = \begin{cases} x, & \text{falls } x \geq 0 \\ -x, & \text{falls } x < 0 \end{cases}$$

Für den Absolutbetrag gelten die folgenden Regeln:

(ABS1) *Dreiecksungleichung:* $\forall\, x, y \in \mathbb{Z} : |x + y| \leq |x| + |y|$, **Dreiecksungleichung**

(ABS2) $\forall x, y \in \mathbb{Z} : |x - y| \leq |x| + |y|$,

(ABS3) $\forall x, y \in \mathbb{Z} : ||x| - |y|| \leq |x| - |y|$,

(ABS4) $\forall x, y \in \mathbb{Z} : |xy| = |x||y|$.

3.6.3 Zusammenfassung

Die ganzen Zahlen können mithilfe einer Äquivalenzrelation über Paare von natürlichen Zahlen eingeführt werden. Die in der Menge der ganzen Zahlen abgeschlossenen Operationen Addition, Subtraktion und Multiplikation werden als Verknüpfungen der entsprechenden Äquivalenzklassen eingeführt. Dabei werden diese Operationen – auch die Subtraktion – alleine durch die Addition und die Multiplikation natürlicher Zahlen definiert. Die Addition und die Multiplikation auf den positiven ganzen Zahlen stimmt dabei mit der Addition und der Multiplikation der entsprechenden natürlichen Zahlen überein. Insofern können die ganzen Zahlen als Erweiterung der natürlichen Zahlen betrachtet werden, in der nun auch die Subtraktion abgeschlossen ist. Diese Erweiterung kann mithilfe des Isomorphiebegriffs mathematisch beschrieben werden.

3.7 Die Menge der rationalen Zahlen

Wir haben im vorigen Kapitel die Rechenstruktur $\mathbb{N}$ erweitert zur Rechenstruktur $\mathbb{Z}$, in der auch die Subtraktion uneingeschränkt ausführbar ist. Die beschränkte Ausführung der Division bleibt aber weiterhin bestehen: Die Division $x : y$ für zwei Zahlen $x \in \mathbb{Z}$ und $y \in \mathbb{Z} - \{0\}$ ist nur ausführbar, falls es ein $z \in \mathbb{Z}$ gibt mit $x = yz$. So existiert etwa der Quotient $4 : 3$ in $\mathbb{Z}$ nicht, da es keine ganze Zahl z gibt mit $4 = 3z$. Das heißt: Nicht alle Gleichungen der Art $ax = b$ mit $a, b \in \mathbb{Z}$ mit $a \neq 0$ sind in $\mathbb{Z}$ lösbar (sondern nur solche, bei denen a ein Teiler von b ist).

In diesem Kapitel werden wir eine neue Zahlenmenge, die Menge $\mathbb{Q}$ der rationalen Zahlen, einführen, in der wir zum einen solche Gleichungen lösen können und in der wir zum anderen die Menge der ganzen Zahlen und das in dieser Menge mögliche Rechnen erhalten, insofern $\mathbb{Z}$ zu $\mathbb{Q}$ erweitern. Auch für $\mathbb{Q}$ werden wir weitere Rechenregeln auflisten.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- verstehen, wie die rationalen Zahlen und das Rechnen mit diesen mithilfe einer geeigneten Äquivalenzrelation auf den ganzen Zahlen eingeführt werden können,
- die Rechenregeln für rationale Zahlen kennen und anwenden können.

3.7.1 Konstruktion der rationalen Zahlen

Analog zur Menge der ganzen Zahlen konstruieren wir die rationalen Zahlen ebenfalls mithilfe einer Äquivalenzrelation: Die Relation

$$\mathcal{Q} \subseteq (\mathbb{Z} \times (\mathbb{Z} - \{0\}) \times (\mathbb{Z} \times \mathbb{Z} - \{0\})$$

sei definiert durch:

$$(a, b)\, \mathcal{Q}\, (c, d) \text{ genau dann, wenn } ad = bc \tag{3.30}$$

Die Relation $\mathcal{Q}$ verwendet nur bekannte Begriffe: ganze Zahlen bzw. Paare von ganzen Zahlen sowie die Multiplikation von ganzen Zahlen.

Der Beweis, dass $\mathcal{Q}$ eine Äquivalenzrelation ist, erfolgt analog dem Beweis im vorigen Kapitel, der zeigt, dass die Relation $\mathcal{Z}$ eine Äquivalenzrelation ist:

$\mathcal{Q}$ ist reflexiv: Für alle $(a, b) \in \mathbb{Z} \times \mathbb{Z} - \{0\}$ gilt $ab = ba$, woraus mit (3.30) folgt, dass für alle $(a, b) \in \mathbb{Z} \times \mathbb{Z} - \{0\}$ gilt: $(a, b)\, \mathcal{Q}\, (a, b)$.

$\mathcal{Q}$ ist symmetrisch: Sei $(a, b)\, \mathcal{Q}\, (c, d)$, d.h. $ad = bc$. Mit den bekannten Rechenregeln folgt daraus, dass $cb = da$ gilt, und hieraus mit (3.30), dass $(c, d)\, \mathcal{Q}\, (a, b)$ gilt.

$\mathcal{Q}$ ist transitiv: Sei $(a, b)\, \mathcal{Q}\, (c, d)$ und $(c, d)\, \mathcal{Q}\, (e, f)$, d.h. es ist $ad = bc$ und $cf = de$. Hieraus folgt, dass $adf = bcf = bde$ ist, und daraus folgt, da $d \neq 0$ ist, dass $af = be$ ist. Mit (3.30) folgt $(a, b)\, \mathcal{Q}\, (e, f)$.

Konstruktion von $\mathbb{Q}$

Die Äquivalenzklassen von $\mathcal{Q}$ sind für $x, y \in \mathbb{Z}$ mit $y \neq 0$ gegeben durch:

$$[(x, y)]_{\mathcal{Q}} = \{\, (qx, qy) \mid q \in \mathbb{Z} - \{0\} \,\} \tag{3.31}$$

Rechenoperationen für rationale Zahlen

Auf der Menge $\mathbb{Q} = (\mathbb{Z} \times \mathbb{Z} - \{0\})/\mathcal{Q}$ der Äquivalenzklassen von $\mathcal{Q}$, die wir *Menge der rationalen Zahlen* nennen wollen, führen wir Rechenoperationen ein:[26]

- Addition und Subtraktion: $[(a, b)]_{\mathcal{Q}} \pm [(c, d)]_{\mathcal{Q}} = [(ad \pm bc, bd)]_{\mathcal{Q}}$
- Multiplikation: $[(a, b)]_{\mathcal{Q}} \cdot [(c, d)]_{\mathcal{Q}} = [(ac, bd)]_{\mathcal{Q}}$

26 Dabei benutzen wir jetzt sofort die üblichen Notationen und nicht erst aus formalen Gründen, wie im vorigen Kapitel bei der Einführung der ganzen Zahlen, neue Symbole.

- Division: $[(a,b)]_{\mathcal{Q}} : [(c,d)]_{\mathcal{Q}} = [(ad,bc)]_{\mathcal{Q}}$

Diese (neuen) Rechenoperationen sind ausschließlich mit ganzen Zahlen und den dafür vollständig definierten Operationen Addition, Subtraktion und Multiplikation definiert. Dabei stellen die Äquivalenzklassen $[(x,y)]_{\mathcal{Q}}$ mit $y = 1$ genau die ganzen Zahlen dar. Auch hier kann man mithilfe eines Isomorphismus zeigen, dass die rationalen Zahlen als Erweiterung der ganzen Zahlen betrachtet werden können.

Für die neuen Zahlen $[(a,b)]_{\mathcal{Q}}$ mit $a \neq 0$ gilt wegen (3.31):

$$[(a,b)]_{\mathcal{Q}} \cdot [(b,a)]_{\mathcal{Q}} = [(ab,ba)]_{\mathcal{Q}} = [(ab \cdot 1, ab \cdot 1)]_{\mathcal{Q}} = [(1,1)]_{\mathcal{Q}}$$

$[(b,a)]_{\mathcal{Q}}$ heißt das *(multiplikative) Inverse* zu $[(a,b)]_{\mathcal{Q}}$.

Weiterhin gilt für alle Zahlen $[(a,b)]_{\mathcal{Q}}, [(c,d)]_{\mathcal{Q}} \in \mathbb{Q}$ mit $c \neq 0$:

$$\begin{aligned}([(a,b)]_{\mathcal{Q}} \cdot [(c,d)]_{\mathcal{Q}}) : [(c,d)]_{\mathcal{Q}} &= [(ac,bd)]_{\mathcal{Q}} : [(c,d)]_{\mathcal{Q}} \\ &= [(acd,bdc)]_{\mathcal{Q}} \\ &= [(cda,cdb)]_{\mathcal{Q}} \\ &= [(a,b)]_{\mathcal{Q}}\end{aligned}$$

In der Einleitung dieses Kapitels haben wir das Ziel formuliert, die ganzen Zahlen so zu erweitern, dass Gleichungen der Art $ax = b$ für $a \neq 0$ lösbar sind. Der folgende Satz besagt, dass dies in der Menge $\mathbb{Q}$ der rationalen Zahlen möglich ist.

Lösung von $ax = b$ in $\mathbb{Q}$

Satz 3.3 Für alle Zahlen $[(a,b)]_{\mathcal{Q}}, [(c,d)]_{\mathcal{Q}} \in \mathbb{Q}$ mit $a \neq 0$ gibt es genau eine Zahl $[(x,y)]_{\mathcal{Q}} \in \mathbb{Q}$ mit $[(a,b)]_{\mathcal{Q}} \cdot [(x,y)]_{\mathcal{Q}} = [(c,d)]_{\mathcal{Q}}$.

Beweis Wir setzen $x = bc$ und $y = ad$, dann gilt wegen (3.31):

$$[(a,b)]_{\mathcal{Q}} \cdot [(bc,ad)]_{\mathcal{Q}} = [(abc,abd)]_{\mathcal{Q}} = [(c,d)]_{\mathcal{Q}}$$

Somit ist die Gleichung $[(a,b)]_{\mathcal{Q}} \cdot [(x,y)]_{\mathcal{Q}} = [(c,d)]_{\mathcal{Q}}$ für gegebene rationale Zahlen $[(a,b)]_{\mathcal{Q}}$ und $[(c,d)]_{\mathcal{Q}}$ mit $a \neq 0$ lösbar.

Wir müssen noch die Eindeutigkeit der Lösung $x = bc$ und $y = ad$ zeigen. Wir nehmen an, es gebe eine weitere Lösung $[(x',y')]_{\mathcal{Q}}$. Es muss dann gelten

$$[(a,b)]_{\mathcal{Q}} \cdot [(x,y)]_{\mathcal{Q}} = [(a,b)]_{\mathcal{Q}} \cdot [(bc,ad)]_{\mathcal{Q}} = [(abc,abd)]_{\mathcal{Q}} = [(c,d)]_{\mathcal{Q}}$$

sowie

$$[(a,b)]_{\mathcal{Q}} \cdot [(x',y')]_{\mathcal{Q}} = [(ax',by')]_{\mathcal{Q}} = [(c,d)]_{\mathcal{Q}}$$

Es folgt, dass

$$[(abc,abd)]_{\mathcal{Q}} = [(ax',by')]_{\mathcal{Q}}$$

sein muss und damit $x' = bc$ und $y' = ad$, womit die Eindeutigkeit gezeigt ist. □

Brüche
Zähler
Nenner

In der uns vertrauten Notation werden rationale Zahlen nicht als Paare $[(a,b)]_{\mathcal{Q}}$, sondern als *Brüche* $\frac{a}{b}$ dargestellt. Dabei heißt a der *Zähler* und b der *Nenner* des Bruches $\frac{a}{b}$.

3.7.2 Rechenregeln in $\mathbb{Q}$

Multiplikatives Inverses

Die Rechenregeln A1 - 4 und M1 - 3 aus $\mathbb{Z}$ gelten analog auch in $\mathbb{Q}$. Zudem gilt wegen Satz 3.3 die Rechenregel

(M4) *Multiplikatives Inverses:* Zu jeder rationalen Zahl $x \neq 0$ existiert eine rationale Zahl y mit $xy = 1$. Wir nennen $x^{-1} = y$ das (multiplikative) Inverse von x.

Aus Satz 3.3 folgt zudem für $x = \frac{a}{b}$ mit $a \neq 0$: $x^{-1} = \frac{b}{a}$.

Für den Absolutbetrag gilt neben den Regeln ABS1 - 4 noch die Regel

(ABS5) $|\frac{a}{b}| = \frac{|a|}{|b|}$.

Die Relation $\leq \subseteq \mathbb{Q} \times \mathbb{Q}$ definiert durch

$$\frac{a}{b} \leq \frac{c}{d} \text{ genau dann, wenn es } \frac{x}{y} \text{ gibt mit } x \geq 0,\ y \geq 1 \text{ und } \frac{a}{b} + \frac{x}{y} = \frac{c}{d}$$

legt eine totale Ordnung auf $\mathbb{Q}$ fest. Für diese Ordnung gelten die Regeln O1, O2+ und O2−.

Dichtheit von $\mathbb{Q}$

Bereits in Beispiel 2.7 b) auf Seite 96 haben wir festgestellt, dass bezüglich dieser Ordnung die Menge der rationalen Zahlen im Gegensatz zur Menge der ganzen Zahlen und damit auch im Gegensatz zur Menge der natürlichen Zahlen dicht ist, d.h. zwischen zwei verschiedenen rationalen Zahlen a und c mit $a \leq c$ existiert immer eine von a und c verschiedene rationale Zahl b mit $a \leq b \leq c$. Für gegebene a und c braucht man nur $b = \frac{a+c}{2}$ zu wählen, denn es gilt:

$$a = \frac{a+a}{2} \leq \frac{a+c}{2} \leq \frac{c+c}{2} = c \tag{3.32}$$

3.7.3 Zusammenfassung

Die rationalen Zahlen können mithilfe einer Äquivalenzrelation über Paare von ganzen Zahlen eingeführt werden. Die in der Menge der rationalen Zahlen abgeschlossenen Operationen Addition, Subtraktion, Multiplikation und Division werden als Verknüpfungen der entsprechenden Äquivalenzklassen eingeführt. Dabei werden diese Operationen – auch die Division – alleine durch Addition, Subtraktion und Multiplikation ganzer Zahlen definiert. Addition, Subtraktion und Multiplikation auf den rationalen Zahlen mit Nenner 1 stimmen dabei mit Addition, Subtraktion und Multiplikation der entsprechenden ganzen Zahlen überein. Insofern können die rationalen Zahlen als Erweiterung der ganzen Zahlen betrachtet werden, in der nun auch die Division abgeschlossen ist. Diese Erweiterung kann mithilfe des Isomorphiebegriffs mathematisch beschrieben werden.

3.8 Rechenstrukturen

Nachdem wir mit der Einführung der Zahlenmengen $\mathbb{N}_0$, $\mathbb{Z}$ und $\mathbb{Q}$ schrittweise immer weitere Rechenmöglichkeiten und Rechenregeln kennen gelernt haben, wollen wir diese nun abstrakt betrachten und strukturieren. Ausgehend von der durch die Peano-Axiome festgelegten Menge der natürlichen Zahlen haben wir diese erweitert zu den ganzen Zahlen, um unbeschränkt subtrahieren zu können, und dann zu den rationalen Zahlen, um unbeschränkt dividieren zu können. Die rationalen Zahlen haben allerdings weitere Einschränkungen. So existiert z.B. in $\mathbb{Q}$ keine Lösung der Gleichung $x^2 = 2$, denn $\sqrt{2}$ ist, wie wir wissen (siehe Beispiel 1.39 auf Seite 69), keine rationale Zahl. Wir werden sehen, wie man $\mathbb{Q}$ so zu einer Rechenstruktur erweitern kann, in der die Gleichung $x^2 = 2$ lösbar ist und in der mit dieser Lösung dann auch wie mit den rationalen Zahlen gerechnet werden kann.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- die Rechenstrukturen Gruppe, Ringe und Körper erklären und Beispiele dafür angeben können,
- beispielhaft verstehen, wie Körper erweitert werden können,
- nachweisen können, dass solche Erweiterungen wieder Körper bilden.

3.8.1 Gruppen, Ringe, Körper

Wir bezeichnen im Folgenden die Menge von Objekten, mit denen gerechnet werden soll, die sogenannte *Trägermenge*, mit M und die Rechenoperationen auf dieser Menge allgemein mit $*$, $*_1$ und $*_2$. Rechenstrukturen $(M, *)$ mit einer Operation nennen wir *einsortig* und Rechenstrukturen $(M, *_1, *_2)$ mit zwei Operationen *zweisortig*. Die grundsätzliche Anforderung an eine Rechenstruktur ist die *Abgeschlossenheit*, d.h. sind $a, b \in M$, dann muss auch $a * b \in M$ sein für jede Operation $*$.

Trägermenge

Einsortige, zweisortige Rechenstruktur

Abgeschlossenheit

$\mathcal{H} = (M, *)$ heißt *Halbgruppe* genau dann, wenn die Operation $*$ *assoziativ* auf M ist, d.h. wenn für alle $a, b, c \in M$ gilt

Halbgruppe

$$(a * b) * c = a * (b * c) \tag{3.33}$$

Beispiele für Halbgruppen sind $(\mathbb{N}, +)$ und $(\mathbb{Z}, \cdot)$.

$\mathcal{M} = (M, *)$ heißt *Monoid* genau dann, wenn $\mathcal{M}$ eine Halbgruppe ist und ein Element $e \in M$ existiert, so dass für alle $a \in M$ gilt

Monoid

$$a * e = a = e * a \tag{3.34}$$

e heißt *Einselement* von $\mathcal{M}$. Man kann zeigen, dass das Einselement eines Monoids eindeutig ist. Ein Synonym für Einselement ist *neutrales Element*.

Einselement
Neutrales Element

Beispiele für Monoide sind $(\mathbb{N}_0, +)$ mit dem (additiven) Einselement 0 und $(\mathbb{Z}, \cdot)$ mit dem (multiplikativen) Einselement 1.

Gruppe

$\mathcal{G} = (M, *)$ heißt *Gruppe* genau dann, wenn $\mathcal{G}$ ein Monoid ist und für alle $a \in M$ ein $b \in M$ existiert mit der Eigenschaft

$$a * b = e = b * a \tag{3.35}$$

Inverses

b heißt *invers* zu a oder $Inverses$ von a. Man kann zeigen, dass das Inverse eindeutig ist. Für das Inverse von a schreiben wir in der Regel a^{-1}. Es gelten für e und alle $a, b \in M$ die Eigenschaften:

$$e^{-1} = e \tag{3.36}$$

$$\big((\, a \,)^{-1} \big)^{-1} = a \tag{3.37}$$

$$(\, a * b \,)^{-1} = b^{-1} * a^{-1} \tag{3.38}$$

Ist die Operation $*$ *kommutativ*, d.h. gilt

$$a * b = b * a \tag{3.39}$$

für alle $a, b \in M$, dann heißen $\mathcal{H}$, $\mathcal{M}$ bzw. $\mathcal{G}$ *kommutativ* oder auch *abelsch*.

Beispiele für kommutative Gruppen sind $(\mathbb{Z}, +)$ und $(\mathbb{Q}^*, \cdot)$ mit $\mathbb{Q}^* = \mathbb{Q} - \{0\}$.

Übungsaufgaben

3.11 (1) Sei M eine Menge. Zeigen Sie, dass die Rechenstrukturen $(\mathcal{P}(M), \cup)$ sowie $(\mathcal{P}(M), \cap)$ komutative Monoide, aber keine Gruppen bilden.

(2) Überlegen Sie, dass die Menge der bijektiven Funktionen einer Menge in sich selbst mit der Komposition als Verknüpfung eine im Allgemeinen nicht kommutative Gruppe bildet! □

Ring

$\mathcal{R} = (M, *_1, *_2)$ heißt *Ring* genau dann, wenn

- $(M, *_1)$ eine kommutative Gruppe bildet (deren Einselement bezeichnen wir mit e_1),
- $(M, *_2)$ eine Halbgruppe bildet,
- die (*Links-* bzw. *Rechts-*) *Distributivgesetze*

$$\begin{aligned} a *_2 (\, b *_1 c \,) &= (\, a *_2 b \,) *_1 (\, a *_2 c \,) \\ (\, a *_1 b \,) *_2 c &= (\, a *_2 c \,) *_1 (\, b *_2 c \,) \end{aligned} \tag{3.40}$$

für alle $a, b, c \in M$ gelten.

Ist $*_2$ ebenfalls kommutativ, dann heißt $\mathcal{R}$ *kommutativer Ring*. Ist $(M, *_2)$ ein Monoid, dann heißt $\mathcal{R}$ *Ring mit Einselement* (welches wir mit e_2 bezeichnen). Bei kommutativen Ringen reicht ein Distributivgesetz aus, weil aus diesem mithilfe der Kommutativität das andere gefolgert werden kann.

Kommutativer Ring

Ring mit Einselement

Ein Beispiel für einen kommutativen Ring mit Einselement ist die Menge der ganzen Zahlen mit den Operationen Addition und Multiplikation: $(\mathbb{Z}, +, \cdot)$. Das Einselement der Addition ist die 0, das Einselement der Multiplikation ist die 1. Da $\mathbb{Z}$ ein „Prototyp“ für Ringe ist, bezeichnet man auch im Allgemeinen die Operation $*_1$ als Additon und die Operation $*_2$ als Multiplikation und notiert diese mit $+$ bzw. mit $\cdot$, und man schreibt im Allgemeinen $\mathcal{R} = (M, +, \cdot)$ anstelle von $\mathcal{R} = (M, *_1, *_2)$. Dementsprechend nennt man e_1 das *additive Einselement* und e_2 das *multiplikative Einselement* und notiert diese auch im Allgemeinen mit 0 bzw. mit 1.

In Abschnitt 2.1.4 haben wir auf $\mathbb{Z}$ für $m \in \mathbb{N}$ die Äquivalezrelation $\equiv_m$ betrachtet. Diese partitioniert $\mathbb{Z}$ in m *Restklassen* $[\,i\,]_m$:

$$[\,i\,]_m = \{x \in \mathbb{Z} \mid x = k \cdot m + i,\, k \in \mathbb{Z}\}\,, \quad 0 \leq i \leq m-1 \tag{3.41}$$

Wählen wir als Repräsentanten die kleinsten positiven Reste $0, 1, \ldots, m-1$ und fassen diese in der Menge

$$\mathbb{Z}_m = \{0, 1, \ldots, m-1\} \tag{3.42}$$

zusammen und definieren darauf die Addition $+_m$ sowie die Multiplikation $\cdot_m$ durch

$$\begin{aligned} [\,i\,]_m +_m [\,j\,]_m &= [\,i+j\,]_m \\ [\,i\,]_m \cdot_m [\,j\,]_m &= [\,i \cdot j\,]_m \end{aligned} \tag{3.43}$$

so erhalten wir als neue Rechenstruktur $(\mathbb{Z}_m, +_m, \cdot_m)$, den *Restklassenring modulo* m. Dieser bildet wie $\mathbb{Z}$ einen kommutativen Ring mit Einselement. Bei diesem schreibt man $+$ anstelle von $+_m$ und $\cdot$ anstelle von $\cdot_m$, falls aus dem Zusammenhang klar ist, dass in $\mathbb{Z}_m$ gerechnet wird.

Restklassenring

Sei $\mathcal{R}$ ein Ring. Existiert zu $a \in \mathcal{R}$ mit $a \neq 0$ ein $b \in \mathcal{R}$ mit $b \neq 0$ und $a \cdot b = 0$, dann heißt a *Nullteiler* in $\mathcal{R}$. So enthält $\mathbb{Z}_{12}$ die Nullteiler $2, 3, 4, 6, 9, 10$, denn es gilt z.B. $2 \cdot 6 = 0$, $3 \cdot 4 = 0$, $4 \cdot 9 = 0$ und $6 \cdot 10 = 0$ in $\mathbb{Z}_{12}$. Besitzt ein Ring keine Nullteiler, dann nennen wir ihn *nullteilerfrei*. $\mathbb{Z}$ und $\mathbb{Z}_7$ sind Beispiele für nullteilerfreie Ringe.

Nullteiler

Einen nullerteilerfreien kommutativen Ring mit Einselement nennt man *Integritätsbereich*. $\mathbb{Z}$, $\mathbb{Z}_7$, $\mathbb{Z}_{11}$ und $\mathbb{Z}_{13}$ sind Beispiele für Integritätsbereiche, $\mathbb{Z}_4$, $\mathbb{Z}_{12}$ unbd $\mathbb{Z}_{20}$ sind keine Integritätsbereiche.

Integritätsbereich

Die bezüglich der Multiplikation invertierbaren Elemente eines Rings mit Einselement nennt man auch *Einheiten*. Man kann zeigen, dass $a \in \mathcal{R}$ eine Einheit ist genau dann, wenn a kein Nullteiler von $\mathcal{R}$ ist. Die Einheiten eines Rings $\mathcal{R}$

Einheit

fasst man in der Menge $\mathcal{R}^*$ zusammen. $\mathcal{R}^*$ ist niemals leer, denn es gilt immer $1 \in \mathcal{R}^*$. Man kann zeigen, dass $\mathcal{R}^*$ für jeden Ring $\mathcal{R}$ eine Gruppe, die so genannte *Einheitengruppe* von $\mathcal{R}$, bildet.

Es gilt z.B.

$$\begin{aligned}\mathbb{Z}^* &= \{1, -1\}\\ \mathbb{Z}_4^* &= \{1, 3\}\\ \mathbb{Z}_7^* &= \{1, 2, 3, 4, 5, 6\}\\ \mathbb{Z}_{11}^* &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\\ \mathbb{Z}_{12}^* &= \{1, 5, 7, 11\}\\ \mathbb{Z}_{20}^* &= \{1, 3, 7, 9, 11, 13, 17, 19\}\end{aligned}$$

Kürzungsregel

In Integritätsbereichen gilt die *Kürzungsregel*: Sei $\mathcal{I}$ ein Integritätsbereich und $a, b, c \in \mathcal{I}$ mit $c \neq 0$ und $ac = bc$, dann folgt $a = b$. Denn aus $ac = bc$ folgt $(a-b)c = 0$. Wegen der Nullteilerfreiheit von $\mathcal{I}$ und weil $c \neq 0$ ist, muss $a - b = 0$ und damit $a = b$ sein. Wir können also in Integritätsbereichen beide Seiten einer Gleichung durch denselben Faktor (ungleich Null) „dividieren“, auch wenn dieser Faktor nicht invertierbar ist. Liegt z.B. in $\mathbb{Z}$ die Gleichung $2a = 2b$ vor, dann folgt daraus auch in $\mathbb{Z}$, dass $a = b$ ist, obwohl wir nicht durch 2 dividieren könnnen, denn in $\mathbb{Z}$ existiert kein multiplikatives Inverses zur 2.

Körper

$\mathcal{K} = (M, +, \cdot)$ bildet einen *Körper* genau dann, wenn

- $(M, +, \cdot)$ einen kommutativen Ring mit Einselement bildet,
- $(M - \{0\}, \cdot)$ eine Gruppe bildet.

Ein kommutativer Ring mit Einselement bildet also einen Körper genau dann, wenn bezüglich der Multiplikation alle Elemente außer der 0 invertierbar sind, wenn also $\mathcal{K}^* = \mathcal{K} - \{0\}$ gilt. Wegen der Kommutativität braucht man bei Körpern von den beiden Distributivgesetzen (3.40) nur eines fordern.

Die Zahlenmengen $\mathbb{Q}$, $\mathbb{R}$ und $\mathbb{C}$ sind „Prototypen“ für Körper. Man kann zeigen, dass endliche Integritätsbereiche immer Körper sind. Des Weiteren kann man zeigen, dass $\mathbb{Z}_m$ nullteilerfrei, damit ein Integritätsbereich und damit ein Körper ist, genau dann, wenn m eine Primzahl ist. Für $p \in \mathbb{P}$ schreibt man anstelle von $\mathbb{Z}_p$ auch $\mathbb{F}_p$ (oder auch $\mathbb{GF}(p)$).

3.8.2 Köpererweiterungen

Fragen wir im Körper $\mathbb{Q}$ der rationalen Zahlen nach einer Lösung der Gleichung $x^2 = 2$, stellen wir fest, dass dort keine Lösung existiert. Es stellt sich die Frage nach der Erweiterung von $\mathbb{Q}$ um Elemente, so dass die Gleichung in dem erweiterten Körper lösbar ist.

Erweiterungskörper

Definition 3.3 Sei $\mathcal{K}$ ein Körper und $\alpha \notin \mathcal{K}$. Bildet $\mathcal{K}(\alpha) = \{\, a + b \cdot \alpha \mid a, b \in \mathcal{K} \,\}$ ebenfalls einen Körper, dann heißt $\mathcal{K}(\alpha)$ *Erweiterungskörper* von $\mathcal{K}$ oder der um α *adjungierte* Körper. □

Beispiel 3.6 $\mathbb{Q}(\sqrt{2})$ ist ein Erweiterungskörper von $\mathbb{Q}$. Die neutralen Elemente von $\mathbb{Q}$ bleiben erhalten, denn 0 und 1 sind auch neutrale Elemente in $\mathbb{R}$, also auch für $\sqrt{2}$.

Weiterhin gilt: $\sqrt{2} \cdot \sqrt{2} = 2 \in \mathbb{Q}$. Deshalb kommen zu $\mathbb{Q}$ nur die Produkte $b\sqrt{2}$ und damit die Summen $a+b\sqrt{2}$ für $a, b \in \mathbb{Q}$ hinzu. Die Trägermenge von $\mathbb{Q}(\sqrt{2})$ ist also: $\{\, a + b\sqrt{2} \mid a, b \in \mathbb{Q} \,\}$.

Es lässt sich leicht nachrechnen, dass $\mathbb{Q}(\sqrt{2})$ abgeschlossen gegenüber Addition und Multiplikation ist:

$$(a + b\sqrt{2}) + (c + d\sqrt{2}) = (a + c) + (b + d)\sqrt{2} \in \mathbb{Q}(\sqrt{2}) \tag{3.44}$$

sowie

$$(a + b\sqrt{2}) \cdot (c + d\sqrt{2}) = (ac + 2bd) + (ad + bc)\sqrt{2} \in \mathbb{Q}(\sqrt{2}) \tag{3.45}$$

Wir müssen noch zeigen, dass zu jedem Element aus dieser Menge sowohl das additive als auch das multiplikative Inverse ebenfalls Elemente der Menge sind. Dann wissen wir, dass $\mathbb{Q}(\sqrt{2})$ einen Körper bildet, da alle anderen Axiome gelten, weil $\mathbb{Q}$ und $\mathbb{R}$ Körper sind.

Das additive Inverse zu $a + b\sqrt{2}$ ist offensichtlich

$$-(a + b\sqrt{2}) = (-a) + (-b)\sqrt{2} = -a - b\sqrt{2} \in \mathbb{Q}(\sqrt{2}) \tag{3.46}$$

Wir zeigen noch die Existenz des multiplikativen Inversen: $c + d\sqrt{2}$ ist invers zu $a + b\sqrt{2}$ für $a, b \in \mathbb{Q}$ mit $b \neq 0$, falls $(a + b\sqrt{2}) \cdot (c + d\sqrt{2}) = 1$ ist. Dies ist der Fall, falls

$$ac + bc\sqrt{2} + ad\sqrt{2} + 2bd = 1$$

ist, und dies gilt, falls die Gleichungen

$$ac + 2bd = 1 \tag{3.47}$$
$$bc + ad = 0 \tag{3.48}$$

gelten. Wir lösen Gleichung (3.48) nach c auf:

$$c = -\frac{ad}{b} \tag{3.49}$$

Wir setzen diesen Term für c in Gleichung (3.47) ein und erhalten

$$-\frac{a^2 d}{b} + 2bd = \frac{2b^2 - a^2}{b} d = 1$$

woraus

$$d = \frac{b}{2b^2 - a^2} \tag{3.50}$$

folgt. Einsetzen von (3.50) in (3.49) liefert

$$c = -\frac{a}{2b^2 - a^2} \tag{3.51}$$

Aus (3.51) und (3.50) folgt, dass

$$-\frac{a}{2b^2 - a^2} + \frac{b}{2b^2 - a^2}\sqrt{2}$$

das multiplikative Inverse zu $a+b\sqrt{2}$ ist. Diese Zahlen existieren für alle $a, b \in \mathbb{Q}$ mit $b \neq 0$, denn die Nenner werden genau dann 0, wenn $b^2 = 2a^2$ ist, und diese Gleichung ist für kein $a, b \in \mathbb{Q}$ mit $b \neq 0$ lösbar. Es gilt also

$$\left(a + b\sqrt{2}\right)^{-1} = -\frac{a}{2b^2 - a^2} + \frac{b}{2b^2 - a^2}\sqrt{2} \tag{3.52}$$

für alle $a, b \in \mathbb{Q}$ mit $b \neq 0$.

Der Fall $b = 0$ ist uninteressant, da dann $a + b\sqrt{2} = a$ ist, und $a \neq 0$ ist als rationale Zahl invertierbar. □

Übungsaufgaben

3.12 Verifizieren Sie, dass die obigen Überlegungen unabhängig von $\sqrt{2}$ sind, dass sie also für alle Körpererweiterungen

$$\mathbb{Q}(\sqrt{\alpha}) = \left\{a + b\sqrt{\alpha} \mid a, b \in \mathbb{Q}\right\}$$

mit $\alpha \in \mathbb{Q}$ und $\sqrt{\alpha} \notin \mathbb{Q}$ gleichermaßen angewendet werden können! □

Wir können die Körpererweiterung $\mathbb{Q}(\sqrt{\alpha})$ auch ohne $\sqrt{\alpha}$ darstellen, nämlich durch die Rechenstruktur

$$(\mathbb{Q} \times \mathbb{Q}, +_\alpha, \cdot_\alpha)$$

wobei die Addition $+$ definiert ist durch (siehe (3.44))

$$(a, b) +_\alpha (c, d) = (a + c, b + d)$$

und die Multiplikation durch (siehe (3.45))

$$(a, b) \cdot_\alpha (c, d) = (ac + \alpha bd, ad + bc)$$

Die Abbildung $\varphi : \mathbb{Q}(\sqrt{\alpha}) \to \mathbb{Q} \times \mathbb{Q}$ definiert durch

$$\varphi(a + b\sqrt{\alpha}) = (a, b)$$

ist ein Isomorphismus.

Übungsaufgaben

3.13 Beweisen Sie diese Aussage! □

Es lässt sich leicht verifizieren, dass φ total, injektiv und surjektiv, also bijektiv ist. Wir zeigen noch, dass φ die Strukturgleichungen bezüglich den Additions- und den Multiplikationsoperationen besitzt:

$$\begin{aligned}\varphi((a+b\sqrt{\alpha})+(c+d\sqrt{\alpha})) &= \varphi((a+c)+(b+d)\sqrt{\alpha})\\ &= (a+c,b+d)\\ &= (a,b)+_\alpha (c,d)\\ &= \varphi(a+b\sqrt{\alpha})+_\alpha \varphi(c+d\sqrt{\alpha})\end{aligned}$$

$$\begin{aligned}\varphi((a+b\sqrt{\alpha})\cdot(c+d\sqrt{\alpha})) &= \varphi((ac+\alpha bd)+(ad+bc)\sqrt{\alpha})\\ &= (ac+\alpha bd, ad+bc)\\ &= (a,b)\cdot_\alpha (c,d)\\ &= \varphi(a+b\sqrt{\alpha})\cdot_\alpha \varphi(c+d\sqrt{\alpha})\end{aligned}$$

Aus den obigen Überlegungen folgt zudem, dass $-_\alpha(a,b) = (-a,-b)$ und

$$(a,b)^{-1_\alpha} = \left(-\frac{a}{\alpha b^2-a^2}, \frac{b}{\alpha b^2-a^2}\right)$$

ist.

Die Strukturen $(\mathbb{Q}(\sqrt{\alpha}),+,\cdot)$ und $(\mathbb{Q}\times\mathbb{Q},+_\alpha,\cdot_\alpha)$ sind also identisch (bis auf die Bezeichnung der Elemente). Mithilfe der totalen Injektion $\phi:\mathbb{Q}\to\mathbb{Q}\times\mathbb{Q}$ definiert durch $\phi(a)=(a,0)$ kann die Struktur $(\mathbb{Q}(\sqrt{\alpha}),+,\cdot)$ in die Struktur $(\mathbb{Q}\times\mathbb{Q},+_\alpha,\cdot_\alpha)$ eingebettet werden, d.h. $(\mathbb{Q}\times\mathbb{Q},+_\alpha,\cdot_\alpha)$ ist tatsächlich eine Erweiterung von $(\mathbb{Q}(\sqrt{\alpha}),+,\cdot)$.

Übungsaufgaben

3.14 Verifizieren Sie diese Einbettung anhand der Addition und der Multiplikation! □

Nun, es gilt offensichtlich $(a,0)+_\alpha(c,0)=(a+c)$ sowie $(a,0)\cdot_\alpha(c,0)=(ac,0)$.

3.8.3 Zusammenfassung

Durch Abstraktion bekannter Rechenregeln für Zahlenmengen kommt man zu algebraischen Rechenstrukturen wie Halbgruppen, Monoide und Gruppen mit einer Operationen sowie Ringen, Integritätsbereichen und Körpern mit zwei Verknüpfungen. Selbst Körper wie die Menge der rationalen Zahlen sind nicht abgeschlossen gegen alle Operationen. So können die rationalen Zahlen um irrationale Wurzeln erweitert werden, so dass die neue Struktur wieder einen Körper bildet und die alte darin eingebettet ist. Der um die Wurzel erweiterte Körper kann gänzlich ohne die Wurzel (ohne die neue Zahl), alleine durch Paare von rationalen Zahlen mit geeigneter Addition und Multiplikation dargestellt werden.

3.9 Die Mengen der reellen und der komplexen Zahlen

Im vorigen Abschnitt haben wir gesehen, wie die Menge $\mathbb{Q}$ der rationalen Zahlen um (einzelne) irrationale Zahlen erweitert werden kann. Diese so erweiterten Zahlenmengen bleiben immer noch abzählbar, denn aus einer Abzählung für $\mathbb{Q}$ kann man eine Abzählung für $\mathbb{Q} \times \mathbb{Q}$ herleiten. Da die Menge $\mathbb{R}$ der reellen Zahlen, die neben den rationalen auch alle irrationalen Zahlen enthält, überabzählbar ist (siehe Abschnitt 3.5), werden wir bisherige Erweiterungsmethoden ($\mathbb{N}$ auf $\mathbb{Z}$, $\mathbb{Z}$ auf $\mathbb{Q}$, Erweiterungen von $\mathbb{Q}$), denen letztendlich immer die Menge der natürlichen Zahlen zugrunde liegen, nicht mehr anwenden können. So benötigt die Einführung der Menge $\mathbb{R}$ der reellen Zahlen Methoden aus der Analysis, die über den inhaltlichen Rahmen dieses Buches hinausgehen. Deshalb deuten wir die Konstruktion der reellen Zahlen nur beispielhaft an. Mithilfe der rationalen Zahlen können dann die komplexen Zahlen allerdings wieder mit der Erweiterungsidee des vorigen Abschnitts eingeführt werden.

3.9.1 Reelle Zahlen

Intervallschachtelung

Eine möglicher Ansatz zur Konstruktion der reeellen Zahlen sind rationale *Intervallschachtelungen*. Ist $\langle\, a_n \,\rangle_{n\in\mathbb{N}_0} = \langle\, a_0, a_1, a_2, \dots \,\rangle$ eine monoton wachsende Folge rationaler Zahlen, d.h. es ist $a_{n+1} \geq a_n$, und ist $\langle\, b_n \,\rangle_{n\in\mathbb{N}_0} = \langle\, b_0, b_1, b_2, \dots \,\rangle$ eine monoton fallende Folge rationaler Zahlen, d.h. es ist $b_{n+1} \leq b_n$, und ist zudem $a_n \leq b_n$ und $\lim_{n\to\infty}(b_n - a_n) = 0$, d.h. die Elemente der Differenzenfolge $\langle\, b_n - a_n \,\rangle_{n\in\mathbb{N}_0}$ werden mit wachsendem n beliebig klein, dann heißt die Folge $I_n = [\, a_n, b_n \,]$ von Intervallen eine Intervallschachtelung.

Beispiel 3.7 Offensichtlich ist durch die Folge von Intervallen

$$I_n = \left[1 - \frac{1}{n}, 1 + \frac{1}{n} \right]$$

eine solche Intervallschachtelung gegeben. □

Man kann nun zeigen, dass alle Intervalle einer solchen Intervallschachtelung genau ein gemeinsames Element besitzen. Dieses Element kann eine rationale Zahl sein wie in obigem Beispiel, in dem genau die Zahl 1 in jedem Intervall enthalten ist. Es gibt aber auch Intervallschachtelungen, die in diesem Sinne keine rationale Zahl bestimmen, solche Zahlen werden irrational genannt.

Beispiel 3.8 Wir definieren die Folgen $\langle\, b_n \,\rangle$ und $\langle\, a_n \,\rangle$ durch:

$$\begin{aligned} b_0 &= 2 \\ b_{n+1} &= \frac{b_n^2 + 2}{2b_n} \\ a_n &= \frac{2}{b_n} \end{aligned}$$

Man kann zeigen (etwa mithilfe vollständiger Induktion), dass $I_n = [\, a_n, b_n \,]$ eine Intervallschachtelung ist und dass

$$a_n \leq \sqrt{2} \leq b_n$$

für alle $n \in \mathbb{N}_0$ gilt. Daraus folgt, dass diese Intervallschachtelung genau die Zahl $\sqrt{2}$ bestimmt. □

Heron-Verfahren

Die Folge $\langle\, b_n \,\rangle$ ist eine Anwendung des *Heron-Verfahrens*,[27] mit dem die Quadratwurzel einer Zahl $\alpha \in \mathbb{Q}_+$ beliebig genau berechnet werden kann: Zur Berechnung von $\sqrt{\alpha}$ bilde man die rekursive Folge

$$b_{n+1} = \frac{b_n^2 + \alpha}{2b_n}$$

mit beliebigem Rekursionsanfang $b_0 \neq 0$ (günstig ist, $b_0 = \frac{1}{2}(\alpha + 1)$ zu wählen). Es gilt dann

$$\sqrt{\alpha} \leq b_{n+1} \leq b_n$$

für alle $n \in \mathbb{N}_0$ sowie

$$\lim_{n \to \infty} b_n = \sqrt{\alpha}$$

Man kann nun eine neue Rechenstruktur einführen: Die Menge aller möglichen Intervallschachtelungen bildet die Trägermenge, und für Intervallschachtelungen

27 Heron von Alexandria (Geburts- und Sterbedatum nicht genau bekann, lebte wohl im 1. Jahrhundert) war ein griechischer Mathematiker und Ingenieur, der für seine Zeit erstaunliche Erkenntnisse über Natur und Technik gewann, Bücher darüber schrieb und Geräte und Maschinen konzipierte und baute. Er hatte bereits erste Ideen zur Automatisierung von Maschinen und Programmierung von Geräten.

kann man eine Addition und eine Multiplikation einführen. Des Weiteren kann man überlegen, dass Intervallschachtelungen im Hinblick auf die Zahl, die sie darstellen nicht eindeutig sind. So wird z.B. die Zahl 1 auch durch die Intervallschachtelungen

$$I_n(c) = \left[1 - \frac{c}{n}, 1 + \frac{c}{n} \right], \; c \in \mathbb{Q}_+$$

definiert. Allerdings kann man eine Äquivalenzrelation über alle Intervallschachtelungen definieren, bei der sich alle Schachtelungen als untereinander äquivalent herausstellen, die dieselbe Zahl bestimmen. Die entsprechenden Äquivalenzklassen bilden dann genau die reellen Zahlen.

Die Menge $\mathbb{R}$ der reellen Zahlen erfüllt alle bisher schon aufgelisteten Rechenegeln A1 - 4, M1 - 4, D1 - 2, ABS1 - 5 sowie O1, O2$+$ und O2$-$.

3.9.2 Komplexe Zahlen

In $\mathbb{R}$ können wir nun Gleichungen der Art $x^2 = \alpha$ mit $\alpha \geq 0$ lösen: Die Lösungen $x = \pm\sqrt{\alpha}$ sind reelle Zahlen. Die Gleichungen $x^2 = -\alpha$ mit $\alpha > 0$ besitzen allerdings keine Lösungen in der Menge $\mathbb{R}$, denn es gibt keine reelle Zahl β mit der Eigenschaft $\beta^2 = -\alpha$. Um Gleichungen dieser Art zu lösen, führt man die sogenannte *imaginäre Zahl* i ein, und zwar durch die sie definierende Eigenschaft $i^2 = -1$. Mit dieser Zahl und der Menge der reellen Zahlen definieren wir die Menge der komplexen Zahlen als

Imaginäre Zahl

$$\mathbb{C} = \{ a + bi \mid a, b \in \mathbb{R} \}$$

Anstelle von $a+bi$ notiert man auch $a+ib$. $\mathit{Re}(z) = a$ heißt *Realteil* und $\mathit{Im}(z) = b$ heißt *Imaginärteil* der komplexen Zahl $z = a + bi$. $\overline{z} = a - bi$ heißt die zu z *konjugierte komplexe Zahl*.

Realteil
Imaginärteil
Konjugierte

Die obige Gleichung $x^2 = -\alpha$ mit $\alpha \in \mathbb{R}$ und $\alpha > 0$ besitzt in $\mathbb{C}$ die Lösung $z = i\sqrt{\alpha}$. Es gilt $\mathit{Re}(z) = 0$ und $\mathit{Im}(z) = \sqrt{\alpha}$.

$\mathbb{C}$ ist eine Erweiterung von $\mathbb{R}$, d.h. anstelle von $\mathbb{C}$ könnten wir auch $\mathbb{R}(i)$ schreiben (vergleiche Abschnitt 3.8.2). Ebenso können wir komplexe Zahlen $a + ib$ als Paare (a, b) reeller Zahlen betrachten, dabei sei $\overline{(a, b)} = (a, -b)$ die Konjugierte zu (a, b). Die Paare $(a, 0)$ entsprechen genau den reellen Zahlen.

Rechenoperationen für komplexe Zahlen

Auf diesen Paaren definieren wir eine Addition und eine Multiplikation wie folgt:

$$(a, b) + (c, d) = (a + c, b + d) \tag{3.53}$$

$$(a, b) \cdot (c, d) = (ac - bd, bc + ad) \tag{3.54}$$

Das Rechnen in der neuen Menge $\mathbb{C}$ wird also mit den Elementen der bekannten Menge $\mathbb{R}$ und den dort bekannten Rechenoperationen eingeführt. Wir gehen hier also genauso wie in den vorherigen Kapiteln vor, in denen wir mit bekannten

Zahlen und darauf definierten Rechenoperationen neue Zahlen und neue Rechenoperationen eingeführt haben.

Übungsaufgaben

3.15 Verifizieren Sie, dass die beiden Definitionen (3.53) und (3.54) auf den Paaren $(a, 0)$ genau der Addition bzw. der Multiplikation reeller Zahlen entspricht! □

Die Rechenstruktur der reellen Zahlen ist also ein Teil der Rechenstruktur der komplexen Zahlen. Insofern kann $\mathbb{C}$ als Erweiterung von $\mathbb{R}$ aufgefasst werden.

Die Addition zweier komplexer Zahlen ist also durch die Addition der beiden Real- sowie der beiden Imaginärteile festgelegt. Wir wollen die Definition der Multiplikation verifizieren: Es gilt

$$(a + bi) \cdot (c + di) = ac + adi + cbi + bdi^2 = (ac - bd) + (bc + ad)i$$

womit Real- und Imaginärteil des Produktes gleich dem Ergebnispaar der Definition (3.54) sind.

Rechenregeln für $\mathbb{C}$

Rechnen Sie nun nach, dass die Addition und die Multiplikation komplexer Zahlen die Rechenregeln A1 - 2, M1 - 2 und D1 - 2 erfüllen!

Rechnen Sie auch nach, dass $(0, 1) \cdot (0, 1) = (-1, 0)$ gilt, was der definierenden Eigenschaft $i^2 = -1$ der imaginären Zahl entspricht. Rechnen Sie des Weiteren nach, dass

$$(a, b) \cdot \overline{(a, b)} = (a^2 + b^2, 0) \tag{3.55}$$

gilt sowie dass für $z, z_1, z_2 \in \mathbb{C}$ die folgenden Beziehungen gelten:

$$\overline{\overline{z}} = z \tag{3.56}$$

$$\overline{z_1 + z_2} = \overline{z_1} + \overline{z_2} \tag{3.57}$$

$$\overline{z_1 \cdot z_2} = \overline{z_1} \cdot \overline{z_2} \tag{3.58}$$

$$\overline{\left(\frac{z_1}{z_2}\right)} = \frac{\overline{z_1}}{\overline{z_2}} \tag{3.59}$$

Übungsaufgaben

3.16 Überlegen Sie sich, welche komplexen Zahlen das neutrale Element der Addition bzw. das neutrale Element der Multiplikation sind, und bestimmen Sie das additive Inverse sowie das multiplikative Inverse zu einer komplexen Zahl z! □

Neutrale Elemente

Nun, das neutrale Element der Addition (x, y) muss die Bedingung

$$(a, b) + (x, y) = (a, b)$$

erfüllen. Aus der Definition (3.53) folgt unmittelbar, dass nur $(x, y) = (0, 0)$ diese Bedingung erfüllt. $(0, 0)$ entspricht der 0 in $\mathbb{R}$.

Das neutrale Element (x, y) der Multiplikation muss die Bedingung

$$(a, b) \cdot (x, y) = (a, b)$$

erfüllen. Ausrechnen gemäß (3.54) ergibt das Gleichungssystem

$$\begin{aligned} ax - by &= b \\ bx - ay &= a \end{aligned}$$

Für $(a, b) \neq (0, 0)$ ergeben sich die Lösungen $x = 1$ und $y = 0$. $(1, 0)$ ist also das neutrale Element der Multiplikation in $\mathbb{C}$; es entspricht der 1 in $\mathbb{R}$.

Additives Inverse

Für das additive Inverse (x, y) zur Zahl (a, b) muss gelten: $(a, b)+(x, y) = (0, 0)$. Aus (3.53) folgt unmittelbar, dass $x = -a$ und $y = -b$ sein muss. Es gilt also: $-(a, b) = (-a, -b)$.

Multiplikatives Inverse

Nun wollen wir noch das multiplikative Inverse (x, y) zur Zahl $(a, b) \neq (0, 0)$ bestimmen. Es muss gelten: $(a, b) \cdot (x, y) = (1, 0)$. Wir erhalten das Gleichungssystem

$$\begin{aligned} ax - by &= 1 \\ bx - ay &= 0 \end{aligned}$$

Multiplikation der ersten Gleichung mit a und der zweiten Gleichung mit b und anschließende Addition der beiden Gleichungen liefert die Gleichung $(a^2 + b^2)x = a$ woraus

$$x = \frac{a}{a^2 + b^2}$$

folgt. Durch Einsetzen dieses Wertes in die zweite Gleichung erhalten wir

$$y = \frac{-b}{a^2 + b^2}$$

Wir haben also berechnet, dass

$$(a, b)^{-1} = \left(\frac{a}{a^2 + b^2}, \frac{-b}{a^2 + b^2} \right)$$

das multiplikative Inverse von $(a, b) \neq (0, 0)$ ist.

Übungsaufgaben

3.17 Rechnen Sie zur Probe nach, dass $(a, b) \cdot (a, b)^{-1} = (1, 0)$ gilt! □

Vergleichbar zum Abschnitt 3.8.2, in dem wir eine Isomorphie zwischen der Erweiterung $\mathbb{Q}(\sqrt{\alpha})$ und $\mathbb{Q} \times \mathbb{Q}$ mit jeweils geeignet definierten Additionen und Multiplikationen festgestellt haben, haben wir mit den obigen Überlegungen eine Isomorphie zwischen $\mathbb{C}$ $(= \mathbb{R}(i))$ und $\mathbb{R} \times \mathbb{R}$ nachgeweisen.

Wir wollen jetzt noch den komplexen Zahlen einen reellen Wert zuweisen. Dabei gehen wir von einer geometrischen Vorstellung aus: Wir betrachten die komplexen Zahlen (x, y) als Punkte im zweidimensionalen kartesischen Koordinatensystem. Wir ordnen der Zahl $z = (x, y)$ als *Betrag* $|z|$ den (euklidischen) Abstand des Punktes (x, y) vom Ursprung $(0, 0)$ zu. „Nach Pythagoras“ ergibt sich

Betrag $|z|$ einer komplexen Zahl

$$|z| = \sqrt{x^2 + y^2} \tag{3.60}$$

Übungsaufgaben

3.18 Rechnen Sie nach, dass für $z, z_1, z_2 \in \mathbb{C}$ folgende Beziehungen gelten: (1) $|z_1 + z_2| \leq |z_1| + |z_2|$, (2) $|z_1 \cdot z_2| \leq |z_1| \cdot |z_2|$, (3) $|z|^2 = z \cdot \overline{z}$. □

Man kann Punkte (x, y) und damit komplexe Zahlen $z = x + iy$ im zweidimensionalen Raum auch durch *Polarkoordinaten* darstellen, nämlich durch den Abstand $r = |z|$ dieses Punktes vom Ursprung $(0, 0)$ und durch den Winkel α zwischen der x-Achse und der Strecke vom Ursprung zum Punkt (x, y). Die Polarkoordinaten der komplexen Zahl $z = x + iy$ sind also gegeben durch das Paar (r,α). r heißt – wie schon bekannt – der *Betrag*, und der Winkel α heißt *Argument* von z. Da für jeden Winkel α für $k \geq 0$ die Punkte $(r,\alpha + 2k\pi)$ identisch sind, schränkt man den Winkel α ein auf das Intervall $-\pi < \alpha \leq \pi$. Damit ist die Darstellung (r,α) der Zahl z eindeutig. Dieses Argument α einer komplexen Zahl z heißt dann ihr *Hauptargument*.

Polarkoordinaten

Argument einer komplexen Zahl

Hauptargument einer komplexen Zahl

Mithilfe des Hauptargumentes α und der Winkelfunktionen $\cos$ und $\sin$ können wir Real- und Imaginärteil einer komplexen Zahl $z = x + iy$ ausdrücken, denn es gilt

$$\cos\alpha = \frac{x}{r} \text{ und } \sin\alpha = \frac{y}{r}$$

und damit

$$\mathit{Re}(z) = x = r \cdot \cos\alpha \text{ bzw. } \mathit{Im}(z) = y = r \cdot \sin\alpha$$

Damit erhalten wir als weitere Darstellung der komplexen Zahl $z = x + iy$:

$$z = r(\cos\alpha + i\ \sin\alpha)$$

mit

$$r = \sqrt{x^2 + y^2}$$

und

$$\alpha = \begin{cases} +\arccos\frac{x}{r}, & y \geq 0 \\ -\arccos\frac{x}{r}, & y < 0 \end{cases}$$

Wir wollen komplexe Zahlen mithilfe ihrer Polarkoordinatendarstellungen multiplizieren und dividieren. Für

$$z_1 = r_1(\cos\alpha_1 + i\,\sin\alpha_1) \text{ und } z_2 = r_2(\cos\alpha_2 + i\,\sin\alpha_2)$$

gilt mithilfe der Additionsgesetze der Winkelfunktionen:

$$\begin{aligned} z_1 z_2 &= r_1 r_2(\cos\alpha_1\,\cos\alpha_2 - \sin\alpha_1\,\sin\alpha_2 + i\,(\cos\alpha_1\,\sin\alpha_2 + \sin\alpha_1\,\cos\alpha_2)) \\ &= r_1 r_2(\cos(\alpha_1+\alpha_2) + i\,\sin(\alpha_1+\alpha_2)) \end{aligned}$$

Analog erhält man für $z_2 \neq 0$:

$$\frac{z_1}{z_2} = \frac{r_1}{r_2}(\cos(\alpha_1-\alpha_2) + i\,\sin(\alpha_1-\alpha_2))$$

Aus der Formel für die Multiplikation folgt unmittelbar eine Formel für die Potenzierung einer komplexen Zahl $z = r(\cos\alpha + i\,\sin\alpha)$:

$$z^n = r^n(\cos(n\,\alpha) + i\,\sin(n\,\alpha))$$

Wurzeln komplexer Zahlen

Hieraus können wir eine Formel für die n-te Wurzel einer komplexen Zahl z herleiten. Dazu sei

$$z = r(\cos\alpha + i\,\sin\alpha) \text{ und } y = \rho(\cos\beta + i\,\sin\beta)$$

mit $y^n = z$, d.h. mit

$$r(\cos\alpha + i\,\sin\alpha) = \rho^n(\cos(n\,\beta) + i\,\sin(n\,\beta))$$

Daraus folgt, dass $\rho = \sqrt[n]{r}$ sowie $\cos\alpha = \cos(n\,\beta)$ und $\sin\alpha = \sin(n\,\beta)$ sein müssen. Dabei können sich die Winkel α und $n\,\beta$ nur um Vielfache von 2π unterscheiden. Es gilt also

$$\beta = \frac{\alpha}{n} + \frac{2k\pi}{n} \text{ für } k \in \mathbb{Z}$$

Damit ergibt sich insgesamt

$$\sqrt[n]{z} = \sqrt[n]{r}\left(\cos\left(\frac{\alpha}{n} + \frac{2k\pi}{n}\right) + i\,\sin\left(\frac{\alpha}{n} + \frac{2k\pi}{n}\right)\right),\ k \in \mathbb{Z}$$

Übungsaufgaben

3.19 Überlegen Sie, was dies geometrisch bedeutet! Wo befinden sich die n-ten Wurzeln in der komplexen Ebene? □

Geometrisch bedeutet dies, dass die n-ten Wurzeln einer komplexen Zahl z auf einem Kreis um den Ursprung mit dem Radius $\sqrt[n]{r}$ mit dem Abstand $\frac{2\pi}{n}$ liegen. Es reicht also eine Wurzel zu bestimmen, die anderen ergeben sich durch $n-1$ Drehungen dieses Punktes um $\frac{2\pi}{n}$.

Einheitswurzeln

Die n-ten Wurzeln der komplexen Zahl $z=1$ heißen *Einheitswurzeln*. Sie liegen auf dem Einheitskreis und haben die Polarkoordinaten $(1, \cos\frac{2k\pi}{n})$, $k \in \mathbb{Z}$.

3.9.3 Algebraische und transzendente Zahlen

Allgemein gilt der sogenannte *Fundamentalsatz der Algebra*, den wir ohne Beweis angeben, da dieser mathematische Kenntnisse verlangt, die nicht innerhalb dieses Buches behandelt werden.

Fundamentalsatz der Algebra Jede Gleichung

$$a_n x^n + a_{n-1}x^{n-1} + \ldots a_1 x + a_0 = 0$$

mit $a_i \in \mathbb{C}$, $0 \leq i \leq n$, $n \geq 1$, besitzt eine Lösung in $\mathbb{C}$. □

Algebraische Zahl

Definition 3.4 Eine Zahl $\alpha \in \mathbb{C}$ heißt *algebraisch* genau dann, wenn sie eine Lösung der Gleichung

$$a_n x^n + a_{n-1}x^{n-1} + \ldots a_1 x + a_0 = 0 \tag{3.61}$$

mit $a_i \in \mathbb{Z}$, $0 \leq i \leq n$, $n \geq 1$, ist. n heißt der *Grad* der Gleichung.

Es sei $\tilde{\mathbb{Q}}$ die Menge der algebraischen Zahlen. □

Folgerung 3.3 **a)** Es gilt $\mathbb{Q} \subset \tilde{\mathbb{Q}}$: Jede rationale Zahl ist algebraisch.

b) Alle Quadratwurzeln $\sqrt{\alpha}$ für $\alpha \in \mathbb{Q}_+$ sind algebraisch.

c) $\tilde{\mathbb{Q}}$ ist abzählbar.

Beweis **a)** Jede rationale Zahl $\frac{p}{q}$, $p \in \mathbb{Z}$, $q \in \mathbb{N}$ ist Lösung der Gleichung $qx - p = 0$, damit gilt $\mathbb{Q} \subseteq \tilde{\mathbb{Q}}$. Die echte Teilmengenbeziehung $\mathbb{Q} \subset \tilde{\mathbb{Q}}$ folgt mithilfe von b).

b) Jede Quadratwurzel $\sqrt{\alpha}$ ist Lösung der Gleichung $x^2 - \alpha = 0$.

c) Die Gleichungen der Art (3.61) vom Grad n werden eindeutig bestimmt durch die Koeffizientenfolge $a_0, \ldots, a_n$, $a_i \in \mathbb{Z}$, $0 \leq i \leq n$, $n \geq 1$. Die Menge aller Gleichungen vom Grad n entspricht also der Menge $\times_{i=0}^{n} \mathbb{Z}^i$. Diese Menge ist, wie wir wissen, abzählbar. Alle möglichen Gleichungen entsprechen der Menge

$$\bigcup_{n=1}^{\infty} \times_{i=0}^{n} \mathbb{Z}^i$$

welche eine abzählbare Vereinigung von abzählbaren Mengen ist, die somit ebenfalls abzählbar ist. Jede Gleichung von Grad n hat höchstens n Lösungen. Insgesamt ergibt sich damit die Behauptung. □

Transzendente Zahlen

Die Menge der $\mathbb{T} = \mathbb{C} - \tilde{\mathbb{Q}}$ der komplexen, nicht algebraischen Zahlen nennt man *transzendent*. Da $\mathbb{C}$ überabzählbar und $\tilde{\mathbb{Q}}$ abzählbar ist, muss $\mathbb{T}$ überabzählbar sein. Bekannte Beispiele für transzendente Zahlen sind die Kreiszahl π und die Euler-Zahl $e = \lim_{n\to\infty}(1 + \frac{1}{n})^n$.

3.9.4 Zusammenfassung

Die Menge $\mathbb{R}$ der reellen Zahlen lässt sich mithilfe von Intervallschachtelungen konstruieren. Die Erweiterung der reellen Zahlen um die imaginäre Zahl i als Lösung der Gleichung $x^2 = -1$ führt zur Menge der reellen Zahlen $\mathbb{C} = \{\, a + bi \mid a, b \in \mathbb{R} \,\}$. Eine komplexe Zahl $z = a + bi$ lässt sich als Punkt (a, b) im zweidimensionalen kartesischen Raum $\mathbb{R} \times \mathbb{R}$ interpretieren. Daraus ergibt sich eine weitere Darstellung komplexer Zahlen, die für viele Anwendungen von Bedeutung ist: die Darstellung in Polarkoordinaten (r,α). Denn es ist $z = r(\cos\alpha + i\sin\alpha)$ mit $r = |z| = \sqrt{a^2 + b^2}$ der Abstand des Punktes (a, b) vom Ursprung $(0, 0)$, und α ist der Winkel zwischen der x-Achse und der Strecke vom Ursprung zum Punkt (a, b).

Die Menge $\tilde{\mathbb{Q}}$ der algebraischen Zahlen ist die Menge der Lösungen von Gleichungen n-ten Grades mit ganzzahligen Koeffizienten. Diese Menge ist abzählbar. Die Menge der nicht algebraischen Zahlen, die Menge der transzendenten Zahlen $\mathbb{T}$, ist überabzählbar.

4 Berechenbarkeit

In den vorigen Kapiteln haben wir an vielen Beispielen gesehen, dass Rekursion ein sehr geeignetes Mittel zur Lösung von Problemen ist. Im Allgemeinen kann man ein Problem $\pi = (\mathbf{F}, \mathbf{A}, f_\pi)$ formalisieren, indem man die Menge der möglichen Fragen $\mathbf{F}$, die Menge der möglichen Antworten $\mathbf{A}$ sowie die Spezifikation $f_\pi : \mathbf{F} \to \mathbf{A}$ angibt, die jeder Frage $x \in \mathbf{F}$, die richtige Antwort $f_\pi(x) = y \in \mathbf{A}$ – soweit existierend – zuordnet. So kann man z.B. die Addition beschreiben durch

Problem
Problemspezifikation

$$add = (\mathbb{N}_0 \times \mathbb{N}_0, \mathbb{N}_0, f_{add})$$

mit

$$f_{add}(x, y) = x + y$$

Gesucht ist nun ein Algorithmus, der dieses Problem löst, d.h. zu gegebenen zwei natürlichen Zahlen deren Summe berechnet.

Berechnungsverfahren sind von alters her bekannt, sie wurden erfunden und angewendet, um Probleme des täglichen Lebens oder in Naturwissenschaften und Technik zu lösen, ohne dass überhaupt festgelegt worden wäre, was denn ein Berechnungsverfahren, ein Algorithmus überhaupt ist. Zu Beginn des letzten Jahrhunderts tauchte dann die Frage danach auf, was überhaupt berechenbar ist. Ein solche Frage kann natürlich nicht beantwortet werden, wenn man den Begriff *Algorithmus* nicht formal präzisiert. So wurde Ende der zwanziger Jahre und in den dreißiger Jahren des vorigen Jahrhunderts mit unterschiedlichen Ansätzen versucht, den Begriff der *Berechenbarkeit* mathematisch zu präzisieren. Dabei betrachtet man Probleme auf natürlichen Zahlen, d.h. Probleme der Art

$$\pi = (\mathbb{N}_0^k, \mathbb{N}_0, f_\pi),\ k \geq 0 \tag{4.1}$$

Einer dieser Ansätze geht davon aus, dass es einige wenige, sehr einfache Funktionen gibt, die man per se als berechenbar ansieht. Aus diesen Funktionen können dann mit rekursiven Konzepten neue Funktion konstruiert werden, die dann ebenfalls als berechenbar angesehen werden. Dieser Ansatz wird in diesem Kapitel einführend vorgestellt.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- das Konzept der primitiv-rekursiven und der μ-rekursiven Funktionen zur Definition von Berechenbarkeit verstehen und erklären können,
- in der Lage sein, für einfache arithmetische Funktionen primitiv-rekursive oder nötigenfalls μ-rekursive Funktionen zu konstruieren,
- die Begriffe Aufzählbarkeit, Entscheidbarkeit und Semi-Entscheidbarkeit und deren Zusammenhänge erläutern können,
- den Begriff der Reduktion von Mengen kennen und auf einfache Problemstellungen anwenden können,
- die Grenzen der Berechenbarkeit verstehen.

K.-U. Witt, *Mathematische Grundlagen für die Informatik*,
DOI 10.1007/978-3-658-03079-7_4, © Springer Fachmedien Wiesbaden 2013

4.1 Primitiv-rekursive Funktionen

Da wir die Berechenbarkeit von k-stelligen Problemen auf natürlichen Zahlen festlegen legen wollen – siehe (4.1) – sollte der Ausgangspunkt dafür die (axiomatische) Festlegung der natürlichen Zahlen sein (siehe Kapitel 3.1). Deshalb gehen wir davon aus, dass folgende *Grundfunktionen* als gegeben angesehen werden können:

Grundfunktionen

Nullfunktionen

Für $k \geq 0$ sind die *k-stelligen Nullfunktionen*

$$\mathcal{O}^k : \mathbb{N}_0^k \to \mathbb{N}_0$$

definiert durch

$$\mathcal{O}^k(x_1, \ldots, x_k) = 0$$

Nachfolgerfunktion

Die *Nachfolgerfunktion*

$$\nu : \mathbb{N}_0 \to \mathbb{N}_0$$

ist definiert durch

$$\nu(x) = x + 1$$

Projektionen

Für $k \geq 1$ sind die *k-stelligen Projektionen*

$$\pi_i^k : \mathbb{N}_0^k \to \mathbb{N}_0,\ 1 \leq i \leq k$$

definiert durch

$$\pi_i^k(x_1, \ldots, x_k) = x_i$$

Aus diesen Grundfunktionen können mithilfe von zwei Konstruktionsschemata weitere Funktionen zusammengesetzt werden:

Komposition

Für $m \in \mathbb{N}$ und $k \in \mathbb{N}_0$ ist die *Komposition* einer Funktion

$$g : \mathbb{N}_0^m \to \mathbb{N}_0$$

mit Funktionen

$$h_1, \ldots, h_m : \mathbb{N}_0^k \to \mathbb{N}_0$$

definiert durch die Funktion

$$\mathcal{C}\,[g; h_1, \ldots, h_m] : \mathbb{N}_0^k \to \mathbb{N}_0$$

mit

$$\mathcal{C}\,[g; h_1, \ldots, h_m]\,(x_1, \ldots, x_k) = g(h_1(x_1, \ldots, x_k), \ldots, h_m(x_1, \ldots, x_k))$$

Der Komposition von Funktionen entspricht die Sequenz von Anweisungen in Programmen: $y_i := h_i(x_1, \ldots, x_k)$, $1 \leq i \leq m$, sind die Ergebnisse der Anweisungen h_i, die dann in die Anweisung g einfließen, um $y := g(y_1, \ldots, y_m)$ zu berechnen (siehe Abbildung 4). Etwas abstrakter ausgedrückt können die h_i als Unterprogramme aufgefasst werden, die vom Programm g aufgerufen werden.

```
read(x_1, ..., x_k);
  y_1 := h_1(x_1, ..., x_k);
  y_2 := h_2(x_1, ..., x_k);
      ⋮
  y_m := h_m(x_1, ..., x_k);
    y := g(y_1, ..., y_m);
write(y)
```

Abb. 4: Komposition als Sequenz von Anweisungen

Primitive Rekursion

Für $k \in \mathbb{N}_0$ ist die *primitive Rekursion* der Funktionen

$$g : \mathbb{N}_0^k \to \mathbb{N}_0$$

und

$$h : \mathbb{N}_0^{k+2} \to \mathbb{N}_0$$

definiert durch die Funktion

$$\mathcal{PRK}\,[g, h] : \mathbb{N}_0^{k+1} \to \mathbb{N}_0$$

mit

$$\mathcal{PRK}\,[g,h]\,(x_1,\ldots,x_k,0) = g(x_1,\ldots,x_k)$$

$$\mathcal{PRK}\,[g,h]\,(x_1,\ldots,x_k,\nu(y)) = h(x_1,\ldots,x_k,y,\mathcal{PRK}\,[g,h]\,(x_1,\ldots,x_k,y))$$

Sei $y = 2 = \nu(1)$, dann gilt

$$\begin{aligned}
&\mathcal{PRK}\,[g,h]\,(x_1,\ldots,x_k,\nu(1))\\
&\quad = h(x_1,\ldots,x_k,1,\mathcal{PRK}\,[g,h]\,(x_1,\ldots,x_k,1))\\
&\quad = h(x_1,\ldots,x_k,1,\mathcal{PRK}\,[g,h]\,(x_1,\ldots,x_k,\nu(0)))\\
&\quad = h(x_1,\ldots,x_k,1,h(x_1,\ldots,x_k,0,\mathcal{PRK}\,[g,h]\,(x_1,\ldots,x_k,0)))\\
&\quad = h(x_1,\ldots,x_k,1,h(x_1,\ldots,x_k,0,g(x_1,\ldots,x_k)))
\end{aligned}$$

Dieses Rekursionsschema ist eine abstrakte, mathematische Beschreibung dessen, was eine Zählschleife in prozeduralen Programmiersprachen bewirkt. Ist

$$f = \mathcal{PRK}\,[g,h] \tag{4.2}$$

dann bewirkt der Aufruf $f(x_1,\ldots,x_k,y)$ Folgendes: y wird als Zählvariable betrachtet. Am Schleifenanfang ($y = 0$) findet mithilfe von g eine Initialisierung statt. Deren Ergebnis geht in die erste Schleifenrunde ein. In jeder Schleifenrunde wird dann mit dem, was in der vorherigen Runde berechnet wurde, mit dem Zähler y und den Schleifenvariablen $x_1,\ldots,x_k$ das Ergebnis dieser Runde berechnet. Die Schleife wird y-mal durchlaufen. In einer prozeduralen Programmiersprache könnte die Schleife $f(x_1,\ldots,x_k,y)$ wie in Abbildung 5 dargestellt aussehen.

```
read(x_1,...,x_k,y);
 f := g(x_1,...,x_k);
 for i := 0 to y − 1 do
     f := h(x_1,...,x_k,i,f)
 endfor
write(f)
```

Abb. 5: Primitive Rekursion als Zählschleife

Definition 4.1 Die Menge $\mathcal{PR}$ der *primitiv-rekursiven Funktionen* ist die kleinste Menge von Funktionen $f : \mathbb{N}_0^k \to \mathbb{N}_0$, $k \geq 0$, die alle Nullfunktionen, alle Projektionen und die Nachfolgerfunktion enthält und die unter Komposition und primitiver Rekursion abgeschlossen ist. □

Beispiel 4.1 **a)** Die Konstante $1 : \mathbb{N}_0^0 \to \mathbb{N}_0$ definiert durch $1() = 1$ ist eine primitiv-rekursive Funktion, denn es gilt

$$1 = \mathcal{C}\left[\nu; \mathcal{O}^0\right] \tag{4.3}$$

mit

$$1() = C\left[\nu; \mathcal{O}^0\right]() = \nu(\mathcal{O}^0()) = \nu(0) = 1$$

b) Jede Konstante $c : \mathbb{N}_0^0 \to \mathbb{N}_0$ definiert durch $c() = c$ ist eine primitiv-rekursive Funktion, denn es gilt

$$c = \underbrace{\mathcal{C}\left[\nu; \mathcal{C}\left[\nu; \ldots; \mathcal{C}\left[\nu;\right.\right.}_{c-\text{mal}} \left.\left.\mathcal{O}^0\right] \ldots\right]] \tag{4.4}$$

mit

$$c() = \underbrace{\mathcal{C}\left[\nu; \mathcal{C}\left[\nu; \ldots; \mathcal{C}\left[\nu;\right.\right.}_{c-\text{mal}} \left.\left.\mathcal{O}^0\right] \ldots\right]]() = \nu^c(\mathcal{O}^0()) = \nu^c(0) = c$$

c) Die Identität $id : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch $id(x) = x$ ist primitiv-rekursiv, denn es gilt

$$id = \pi_1^1$$

mit

$$id(x) = \pi_1^1(x) = x$$

d) Die Addition $add : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch $add(x, y) = x + y$ ist primitiv-rekursiv, denn es ist

$$add = \mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right] \tag{4.5}$$

Es gilt z.B.

$$\begin{aligned}
add(3,2) &= \mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu, \pi_3^3\right]\right](3,2) \\
&= \mathcal{C}\left[\nu; \pi_3^3\right](3, 1, \mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right](3,1)) \\
&= \nu(\pi_3^3(3, 1, \mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right](3,1))) \\
&= \nu(\mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right](3,1)) \\
&= \nu(\mathcal{C}\left[\nu; \pi_3^3\right](3, 0, \mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right](3,0))) \\
&= \nu(\nu(\pi_3^3(3, 0, \mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right](3,0))) \\
&= \nu(\nu(\mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right](3,0))) \\
&= \nu(\nu(\pi_1^1(3,0))) \\
&= \nu(\nu((3))) \\
&= \nu(4) \\
&= 5
\end{aligned}$$

e) Die Multiplikation $mult : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch $mult(x, y) = x \cdot y$ ist primitiv-rekursiv, denn es gilt

$$mult = \mathcal{PRK}\left[\mathcal{O}^1, \mathcal{C}\left[add; \pi_1^3, \pi_3^3\right]\right] \tag{4.6}$$

f) Die Quadrierung $sqr : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch $sqr(x) = x^2$ ist primitiv-rekursiv, denn es gilt

$$sqr = \mathcal{C}\left[mult; \pi_1^1, \pi_1^1\right] \tag{4.7}$$

g) Die Vorgängerfunktion $pre : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$pre(x) = \begin{cases} 0, & x = 0 \\ x - 1, & x \geq 1 \end{cases}$$

ist primitiv-rekursiv, denn es gilt:

$$pre = \mathcal{PRK}\left[\mathcal{O}^0, \pi_1^2\right] \tag{4.8}$$

h) Die Subtraktion $minus : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiertt durch

$$minus(x, y) = \begin{cases} x - y, & x \geq y \\ 0, & x < y \end{cases}$$

ist primitiv-rekursiv, denn es gilt:

$$minus = \mathcal{PRK}\left[id, \mathcal{C}\left[pre; \pi_3^3\right]\right] \tag{4.9}$$

i) Die Potenzfunktion $exp : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$exp(b, x) = b^x$$

ist primitiv-rekursiv, denn es gilt:

$$exp = \mathcal{PRK}\left[1, \mathcal{C}\left[mult; \pi_1^3, \pi_3^3\right]\right] \tag{4.10}$$

j) Die Signumfunktion $sign : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$sign(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \end{cases}$$

ist primitiv-rekursiv, denn es gilt

$$sign = \mathcal{C}\left[minus; \pi_1^1, pre\right] \tag{4.11}$$

k) Der Test auf Gleichheit $equal : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$equal(x, y) = \begin{cases} 1, x = y \\ 0, x \neq y \end{cases}$$

ist primitiv-rekursiv, denn es gilt

$$equal = \mathcal{C}\left[minus; 1, \mathcal{C}\left[sign; \mathcal{C}\left[add; \mathcal{C}\left[minus; \pi_1^2, \pi_2^2\right], \mathcal{C}\left[minus; \pi_2^2, \pi_1^2\right]\right]\right]\right] \tag{4.12}$$

□

Wie man an den obigen Beispielen sieht, ist die strenge syntaktische Notation der primitiv-rekursiven Funktionen nicht immer sehr verständlich. Deshalb werden wir im Folgenden primitiv-rekursive Funktionen etwas leserlicher notieren (in „Pseudocode"), indem wir sie von „syntaktischem Balast" befreien. So ist für die Signumfunktion im Beispiel j) die Notation

$$sign(x) = x - pre(x)$$

verständlicher als die strenge Notation in (4.11). Analog ist die Darstellung

$$equal(x, y) = 1 - sign((x - y) + (y - x))$$

für den Gleichheitstest lesbarer als die Darstellung (4.12).

Übungsaufgaben

4.1 (1) Berechnen Sie mit dem Schema (4.6) $mult(3, 2)$!

(2) Berechnen Sie mit dem Schema (4.8) $pre(3)$!

(3) Berechnen Sie mit dem Schema (4.9) $minus(5, 2)$!

(4) Berechnen Sie mit dem Schema (4.10) $exp(2, 3)$!

4.2 Geben Sie primitv-rekursive Funktionen an, die die Funktionen

(1) $sum : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$sum(n) = \sum_{i=0}^{n} i$$

(2) sowie die Fakultätsfunktion $fak : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$fak(n) = \begin{cases} 1, & n = 0 \\ n \cdot fak(n-1), & n \geq 1 \end{cases}$$

berechnen! □

Wenn man bereits Erfahrung in der Programmierung hat, kann man sich vielleicht zuerst nach dem Schema in Abbildung 5 eine Zählschleife zur Berechnung von $sum(n)$ wie in Abbildung 6 dargestellt überlegen:

```
read(n);
 sum := 0;
 for i := 0 to n − 1 do
     sum := add(i, sum)
 endfor
write(sum)
```

Abb. 6: Zählschleife zur Berechnung von $\mathrm{sum}(n)$

Hieraus kann man dann unmittelbar die Notation von sum in unserer Programmiersprache der primitiv-rekursiven Funktionen gemäß dem Schema (4.2) ablesen:

$$sum = \mathcal{PRK}\left[\mathcal{O}^0, add\right]$$

Aus Beispiel 3.3 a) auf Seite 125 wissen wir, dass

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

gilt. Damit können wir die Funktion sum auch wie folgt primitiv-rekursiv programmieren:

$$sum = \mathcal{C}\left[div; \mathcal{C}\left[mult; \pi_1^1, \mathcal{C}\left[\nu; \pi_1^1\right]\right], 2\right]$$

Dabei ist div die Divisionsfunktion, die wir im nächsten Abschnitt kennen lernen, und 2 ist die Konstantenfunktion $2()$, die die Zahl 2 berechnet (siehe Beispiel b) oben).

Analog wie bei der Summation überlegen wir uns gemäß dem Muster in Abbildung 5 die in Abbildung 7 dargestellte Zählschleife zur Berechnung der Fakultät und lesen daraus die Notation von fak in unserer Programmiersprache der primitiv-rekursiven Funktionen gemäß dem Schema (4.2) ab:

$$fak = \mathcal{PRK}\left[1, mult\right]$$

4.2 μ-Rekursion

Die Grundfunktionen, Nullfunktionen, Projektionen und Nachfolgerfunktion, sind totale Funktionen. Die Komposition und die primitive Rekursion erhalten

```
read(n);
 fak := 1;
 for i := 0 to n − 1 do
     fak := mult(i, fak)
 endfor
write(fak)
```

Abb. 7: Zählschleife zur Berechnung von fak(n)

diese Eigenschaft, d.h. jedes Element von $\mathcal{PR}$ ist eine totale Funktion. Es gibt aber Funktionen, die wir sicherlich als berechenbar ansehen, die aber nicht total definiert sind. Beispiele sind die Wurzelfunktionen und die Logarithmen. Die Quadratwurzel etwa ist nur auf Quadratzahlen definiert und der Logarithmus zur Basis zwei nur für Zweierpotenzen. Diese nur partiell definierten Funktionen können also nicht primitiv-rekursiv sein. Wir benötigen also noch ein weiteres Konzept, mit denen auch nicht totale Funktionen berechnet werden können.

Definition 4.2 Sei $g : \mathbb{N}_0^{k+1} \to \mathbb{N}_0$, dann ist die Funktion $\mu[g] : \mathbb{N}_0^k \to \mathbb{N}_0$ definiert durch **μ-Rekursion**

$$\mu[g](x_1, \ldots, x_k) = \\ min\{\, z \mid (x_1, \ldots, x_n, y) \in Def(g) \text{ für } y \leq z \text{ und } g(x_1, \ldots, x_k, z) = 0 \,\}$$

Das Ergebnis der *μ-Rekursion* auf g ist also das kleinste z mit $g(x_1, \ldots, x_k, z) = 0$ unter der Voraussetzung, dass $g(x_1, \ldots, x_k, y)$ für $y \leq z$ definiert ist. □

Die μ-Rekursion kann als mathematische Abstraktion von Wiederholungsschleifen (While-Schleifen) in prozeduralen Programmiersprachen betrachtet werden. y ist die Schleifenvariable, die mit Null beginnend hoch gezählt wird, und in jedem Schleifendurchlauf wird geprüft, ob $g(x_1, \ldots, x_k, y) = 0$ ist. Falls das zutrifft, ist $z = y$, die Anzahl der Schleifendurchläufe, das Ergebnis der Schleife. Falls die Bedingung nicht erreicht wird, dann terminiert die Schleife nicht, d.h. $\mu[g](x_1, \ldots, x_k)$ ist nicht definiert. In einer prozeduralen Programmiersprache könnte die Schleife zur Berechnung von

$$f(x_1, \ldots, x_k) = \mu[g](x_1, \ldots, x_k) \tag{4.13}$$

wie in Abbildung 8 dargestellt aussehen.

Definition 4.3 Die Klasse $\mu\mathcal{PR}$ der *μ-rekursiven Fuktionen* ist die kleinste Menge von Funktionen $f : \mathbb{N}_0^k \to \mathbb{N}_0$, $k \geq 0$, die alle Grundfunktionen (Nullfunktionen, Projektionen, Nachfolgerfunktion) enthält und die unter Komposition, primitiver Rekursion und unter Anwendung des μ-Operators abgeschlossen ist. □

```
read(x_1, ..., x_k);
 f := 0;
 while g(x_1, ..., x_k, f) ≠ 0 do
     f := f + 1
 endwhile;
write(f)
```

Abb. 8: μ-Rekursion als Wiederholungsschleife

Beispiel 4.2 **a)** Die Funktion $sub : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$sub(x,y) = \begin{cases} x-y, & x \geq y \\ \bot, & x < y \end{cases}$$

ist μ-rekursiv. Dazu betrachten wir folgende Funktion $g : \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ (in Pseudocode) definiert durch

$$g(x,y,d) = sign(x-y)(x-(y+d)) + sign(y-x)$$

dabei soll „–“ die Funktion $minus$ aus Beispiel 4.1 h) darstellen. Die Funktionen $sign$, $minus$, add und $mult$ sind primitiv-rekursiv (siehe Beispiel 4.1), also ist auch die Funktion g primitiv-rekursiv. Es z.B.

$$g = \mathcal{C}\left[add; \mathcal{C}\left[mult; \mathcal{C}\left[sign; \mathcal{C}\left[minus; \pi_1^3, \pi_2^3\right]\right], \mathcal{C}\left[minus; \pi_1^3, \mathcal{C}\left[add; \pi_2^3, \pi_3^3\right]\right]\right], \mathcal{C}\left[sign; \mathcal{C}\left[minus; \pi_2^3, \pi_3^3\right]\right]\right]$$

Wir betrachten drei Fälle: (1) $x = y$, (2) $x > y$ und (3) $x < y$.

Zu (1): Ist $x = y$, dann ist $g(x,y,d) = 0$ für alle $d \in \mathbb{N}_0$.

Zu (2): Ist $x > y$, dann ist $sign(x-y) = 1$, $sign(y-x) = 0$ und $(x-(y+d)) > 0$ für $d < x-y$, d.h. es ist $g(x,y,d) = 1$ für $d < x-y$ und $g(x,y,,x-y) = 0$.

Zu (3): Ist $x < y$, dann ist $sign(x-y) = 0$ und $sign(y-x) = 1$ für alle $d \in \mathbb{N}_0$, d.h. es ist $g(x,y,d) = 1$ für alle $d \in \mathbb{N}_0$.

Aus diesen Überlegungen folgt, dass

$$sub = \mu[g] \tag{4.14}$$

ist.

Betrachten wir das Muster von Abbildung 8, dann wird sub von dem in Abbildung 9 dargestellten Programm berechnet. Falls $x \geq y$ ist, dann wird sub, welches die in g benutzte Variable d für die Differenz von x und y realisiert, so

```
read(x, y);
 sub := 0;
 while sign(x − y)(x − (y + sub)) + sign(y − x) ≠ 0 do
     sub := sub + 1
 endwhile;
write(sub)
```

Abb. 9: Berechnung der Funktion sub durch μ-Rekursion

lange hoch gezählt, bis $sub = x - y$ ist, und sub wird ausgegeben. Ist $x < y$, dann terminiert die Schleife nicht, d.h. für die Eingabe (x, y) erfolgt keine Ausgabe.

b) Die Funktion $\sqrt{\ } : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$\sqrt{x} = \begin{cases} y, & \text{falls } x = y^2 \\ \bot, & \text{sonst} \end{cases}$$

ist μ-rekursiv. Dazu setzen wir $g(x, y) = x \ominus y^2$, dabei sei „$\ominus$“ eine Darstellung für die Funktion sub. g ist eine μ-rekursive Funktion, denn es gilt

$$g = \mathcal{C}\left[sub; \pi_1^2, \mathcal{C}\left[sqr; \pi_2^2\right]\right]$$

Wir können nun zu gegebenem x die Quadratwurzel berechnen, indem wir y beginnend bei Null hoch zählen und dabei testen, ob $x = y^2$ ist. Falls das erreicht wird, ist $y = \sqrt{x}$, ansonsten terminiert das Verfahren nicht. Es gilt

$$\sqrt{\ } = \mu\left[g\right] = \mu\left[\mathcal{C}\left[sub; \pi_1^2, \mathcal{C}\left[sqr; \pi_2^2\right]\right]\right] \tag{4.15}$$

Abbildung 19 zeigt eine Implementierung dieses Verfahrens.

```
read(x);
 y := 0;
 while x ⊖ y² ≠ 0 do
     y := y + 1
 endwhile;
write(y)
```

Abb. 10: Berechnung der Funktion $\sqrt{\ }$ durch μ-Rekursion

```
read(b, x);
 y := 0;
 while  x ⊖ exp(b, y) ≠ 0  do
     y := y + 1
 endwhile;
write(y)
```

Abb. 11: Berechnung der Logarithmus-Funktion durch μ-Rekursion

c) Die Funktion $\log : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$\log(b,x) = \begin{cases} y, & \text{falls } b^y = x \\ \perp, & \text{sonst} \end{cases}$$

ist μ-rekursiv. Dazu setzen wir $g(b,x,y) = x \ominus exp(b,y)$ (siehe Beispiel 4.1 i). g ist μ-rekursiv, denn es gilt

$$g = \mathcal{C}\left[sub; \pi_2^3, \mathcal{C}\left[exp; \pi_1^3, \pi_3^3\right]\right]$$

Wir können nun zu gegebenem b und x den Logarithmus von x zur Basis b berechnen, indem wir y beginnend bei Null hoch zählen und dabei testen, ob $b^y = x$ ist. Falls das erreicht wird, ist $y = \log_b x$, ansonsten terminiert das Verfahren nicht. Es gilt

$$\log = \mu\left[f\right] = \mu\left[\mathcal{C}\left[sub; \pi_2^3, \mathcal{C}\left[exp; \pi_1^3, \pi_3^3\right]\right]\right] \tag{4.16}$$

Abbildung 11 zeigt eine Implementierung dieses Verfahrens. □

Übungsaufgaben

4.3 Zeigen Sie, dass die folgenden Funktionen μ-rekursiv sind!

(1) Die nirgends definierte Funktion $\omega : \mathbb{N}_0 \to \mathbb{N}_0$ mit $\omega(x) = \perp$ für alle $x \in \mathbb{N}_0$, d.h. mit $Def(\omega) = \emptyset$.

(2) $div : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$div(a,b) = \begin{cases} q, & \text{falls } b \neq 0 \text{ und } q, r \in \mathbb{N}_0 \text{ existieren mit } a = bq + r \\ & \text{und } 0 \leq r < b \\ \perp, & \text{sonst} \end{cases}$$

(3) $mod : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$mod(a,b) = \begin{cases} r, & \text{falls } b \neq 0 \text{ und } q, r \in \mathbb{N}_0 \text{ existieren mit } a = bq + r \\ & \text{und } 0 \leq r < b \\ \perp, & \text{sonst} \end{cases}$$

(4) $\sqrt[n]{} : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$\sqrt[n]{x} = \begin{cases} y, & \text{falls } x = y^n \\ \perp, & \text{sonst} \end{cases}$$

(5) $max : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$max(x,y) = \begin{cases} x, & \text{falls } x \geq y \\ y, & \text{sonst} \end{cases}$$

(6) $min : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$min(x,y) = \begin{cases} x, & \text{falls } x \leq y \\ y, & \text{sonst} \end{cases}$$

(7) $geq : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$geq(x,y) = \begin{cases} 1, & \text{falls } x \geq y \\ 0, & \text{sonst} \end{cases}$$

(8) $() : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$\binom{n}{k} = \begin{cases} \frac{n!}{k! \cdot (n-k)!}, & n \geq k \\ 0, & \text{sonst} \end{cases}$$

□

4.3 Churchsche These

Die μ-rekursiven Funktionen sind ein Konzept, um den Begriff der Berechenbarkeit von Funktionen $f : \mathbb{N}_0^k \to \mathbb{N}_0$ mathematisch zu präzisieren.[28] Daneben gibt es eine Reihe weiterer Konzepte, wie z.B. Turingmaschinen, Universelle Registermaschinen, Goto-Programme, While-Programme, Markov-Algorithmen, λ-Kalkül. Obwohl diese Konzepte teilweise auf sehr unterschiedlichen Ansätzen

28 Dieser Ansatz geht auf Stephen C. Kleene (1909 – 1998) zurück, einem amerikanischen Mathematiker und Logiker, der fundamentale Beiträge zur Logik und zur Theoretischen Informatik geliefert hat.

basieren, kann gezeigt werden, dass sie alle dieselbe Klasse von Funktionen $\{ f \mid f : \mathbb{N}_0^k \to \mathbb{N}_0,\ k \geq 0 \}$ als berechenbar festlegen. Diese Äquivalenz der Berechenbarkeitskonzepte ist die Begründung für die Churchsche These.[29]

Churchsche These

Churchsche These Die Klasse $\mu\mathcal{PR}$ der μ-rekursiven Funktionen ist genau die Klasse der im intuitiven Sinne berechenbaren Funktionen. □

Partiell berechenbare Funktionen $\mathcal{P}$

Wir bezeichnen mit

$$\mathcal{P} = \{ f \mid f : \mathbb{N}_0^k \to \mathbb{N}_0,\ k \geq 0,\ f \text{ ist im intuitiven Sinn berechenbar} \}$$

die *Klasse der partiell berechenbaren Funktionen*. Die Churchsche These besagt, dass $\mu\mathcal{PR} = \mathcal{P}$ ist.

Total berechenbare Funktionen $\mathcal{R}$

Ackermann-funktion

Zu Beginn von Kapitel 4.2 haben wir bereits festgestellt, dass $\mathcal{PR}$, die Klasse der primitiv-rekursiven Funktionen, eine echte Teilklasse von $\mathcal{P}$ ist: $\mathcal{PR} \subset \mathcal{P}$. Des Weiteren haben wir dort überlegt, dass alle primitiv-rekursiven Funktionen total sind. Wenn wir mit $\mathcal{R}$ die Klasse der total berechenbaren Funktionen bezeichnen, dann gilt also $\mathcal{PR} \subseteq \mathcal{R}$. Die Frage ist, ob auch die Umkehrung gilt, d.h., ob alle total berechenbaren Funktionen auch primitiv-rekursiv sind. In den ersten Jahren der Berechenbarkeitstheorie gab es Vermutungen in diese Richtung. Doch F.W. Ackermann[30] konnte 1926 (veröffentlicht 1928) eine Funktion angeben, die total berechenbar, aber nicht primitiv-rekursiv ist. Ackermanns Idee war, eine Folge stark anwachsender Funktionen $\varphi_i(a, b)$, $i \geq 0$, rekursiv wie folgt festzulegen:

$$b + 1,\ a + b,\ a \cdot b,\ a^b,\ \ldots \tag{4.17}$$

Diese Folge hat die Eigenschaft, dass bei jedem Glied die Operation des vorherigen Gliedes $(b-1)$-mal auf a angewandt wird. So bedeutet $a+b$, dass $(b-1)$-mal 1 zu $a + 1$ addiert wird:

$$a + b = a + 1 \underbrace{+1 + 1 + \ldots + 1}_{(b-1)\text{-mal}}$$

$a \cdot b$ bedeutet, dass $(b - 1)$-mal a zu a addiert wird:

$$a \cdot b = a \underbrace{+a + a + \ldots + a}_{(b-1)\text{-mal}}$$

Und a^b bedeutet, dass $(b - 1)$-mal a mit a multipliziert wird:

$$a^b = a \underbrace{\cdot a \cdot a \cdot \ldots \cdot a}_{(b-1)\text{-mal}}$$

29 Diese These wurde vorgeschlagen und begründet von Alonzo Church (1903 – 1995), amerikanischer Mathematiker und Logiker, der, wie sein Schüler Kleene, wesentliche Beiträge zur mathematischen Logik und Berechenbarkeitstheorie geleistet hat. Auf Church geht auch das Berechenbarkeitskonzept des λ-Kalküls zurück.

30 Friedrich Wilhelm Ackermann (1896 - 1962) lieferte wichtige Beiträge zu mathematischen Grundlagen, zur mathematischen Logik und zur Rekursionstheorie.

Für das nächste Folgenglied gibt es keine gängige mathematische Notation. Deshalb ist in (4.17) an der Stelle ... angegeben. Dieses Folgendglied ergibt sich, indem $(b-1)$-mal a mit sich selbst fortgesetzt potenziert wird:

$$a^{a^{\cdot^{\cdot^{a}}}}$$

Ackermann beschrieb diese Folge mit der Funktion $\varphi : \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$, deren drittes Argument die entsprechende Nummer des Gliedes in der Folge (4.17) angibt, d.h. $\varphi(a,b,i) = \varphi_i(a,b)$:

$$\begin{aligned}
\varphi(a,b,0) &= b+1 \\
\varphi(a,b,1) &= a+b \\
\varphi(a,b,2) &= a \cdot b \\
\varphi(a,b,3) &= a^b \\
\varphi(a,b,4) &= \ldots
\end{aligned}$$

Rózsa Péter[31]konnte 1955 eine vereinfachte Definition der Ackermannfunktion φ angeben. Ihre Definition dieser Funktion ist die, die wir bereits aus Definiton 3.1 auf Seite 134 kennen. Diese Definition kommt ohne informelle Beschreibung der fortgesetzten Definition der Folgenglieder bzw. ohne Einführung neuer mathematischer Operationen zu deren formalen Beschreibung aus.

Es kann nun zum einen gezeigt werden, dass die in Kapitel 3.4 auf der Basis der Ackermannfunktion definierte Funktion ack (siehe Gleichung (3.10)auf Seite 135) total berechenbar ist, und zum anderen, dass diese Funktion stärker wächst, als alle primitiv-rekursiven Funktionen.

Satz 4.1 Zu jeder primitv-rekursiven Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ existiert ein $x_f \in \mathbb{N}_0$, so dass $ack(x) > f(x)$ für alle $x \geq x_f$ ist. □

Folgerung 4.1 **a)** Es ist $ack \in \mathcal{R} - \mathcal{PR}$.

b) Es gilt $\mathcal{PR} \subset \mathcal{R}$ sowie

c) $\mathcal{PR} \subset \mathcal{R} \subset \mathcal{P}$. □

Wir werden in Kapitel 4.5 noch sehen, dass $\mathcal{P} \subset \{ f \mid f : \mathbb{N}_0^k \to \mathbb{N}_0,\ k \geq 0 \}$ ist, d.h. es gibt Funktionen, die nicht berechenbar sind, und wir werden auch ein Beispiel einer solchen Funktion angeben.

4.4 utm- und smn-Theorem

Das Berechenbarkeitskonzept der μ-rekursiven Funktionen erfüllt (wie auch die anderen erwähnten Konzepte) zwei wesentliche Anforderungen, die man sowohl aus theoretischer als auch aus praktischer Sicht an solche Konzepte stellen kann:

31 Die ungarische Mathematikerin Rózsa Péter (1905 - 1977) lieferte wesentliche Beiträge zur Theoreie der rekursiven Funktionen.

Anforderungen an Berechenbarkeitskonzepte

(U) die Existenz einer berechenbaren universellen Funktion und

(S) die Möglichkeit der effektiven Programmierung.

4.4.1 Nummerierung der berechenbaren Funktionen

Um diese beiden Anforderungen zu beschreiben und zu analysieren, nehmen wir eine Nummerierung der Menge $\mathcal{P}$ der partiell berechenbaren Funktionen vor. Wenn wir uns noch einmal die Beispiele in den Kapiteln 4.1 und 4.2 ansehen, können wir feststellen, dass μ-rekursive Funktionen syntaktisch betrachtet Zeichenketten sind, die aus Symbolen für die Namen der Grundfunktionen, aus Symbolen für die Komposition, die primitive Rekursion und die μ-Rekursion sowie aus eckigen Klammern bestehen, außerdem kommen Kommata und das Semikolon vor.

Für die folgenden Betrachtungen wollen wir die Exponenten bzw. Indizes bei den Nullfunktionen $\mathcal{O}^k$, $k \geq 0$, und den Projektionen π_i^k, $1 \leq i \leq k$, $k \geq 1$, mithilfe der „Bierdeckelnotation“ kodieren, indem wir folgende Darstellungen vornehmen:

$$\mathcal{O}^k \longrightarrow \mathcal{O}|^k,\ k \geq 0$$
$$\pi_i^k \longrightarrow \pi|^k\#|^i,\ 1 \leq i \leq k,\ k \geq 1$$

Es wird also z.B. $\mathcal{O}^3$ durch $\mathcal{O}|||$ und π_2^5 durch $\pi|||||\#||$ codiert.

Mit diesen Darstellungen der Nullfunktionen und der Projektionen können wir nun μ-rekursive Funktionen syntaktisch als Wörter über dem folgenden 12-elementigen Alphabet auffassen:

$$\Sigma = \{\, |, \#, \mathcal{O}, \pi, \nu, \mathcal{C}, \mathcal{PRK}, \mu, [,], ;, , \,\} \tag{4.18}$$

Mit diesem Alphabet haben wir also eine Codierung

$$\sigma : \mu\mathcal{PR} \rightarrow\ \Sigma^+$$

festgelegt. Für die im Beispiel 4.1 d) programmierte Funktion

$$add = \mathcal{PRK}\left[\pi_1^1, \mathcal{C}\left[\nu; \pi_3^3\right]\right]$$

gilt z.B.

$$\sigma(add) = \mathcal{PRK}\left[\pi|\#|, \mathcal{C}\left[\nu; \pi|||\#||||\right]\right]$$

Jetzt gehen wir noch einen Schritt weiter und ordnen den Buchstaben den zwölf Buchstaben des Alphabetes Σ in der in (4.18) aufgezählten Reihenfolge die Zahlen $1, 2, \ldots, 12$ zu, d.h. wir verwenden z.B. die Bijektion

$$\rho : \Sigma \rightarrow \{\, 1, 2, \ldots, 12 \,\}$$

$\rho(\vert) = 1$	$\rho(\pi) = 4$	$\rho(\mathcal{PRK}) = 7$	$\rho(]) = 10$
$\rho(\#) = 2$	$\rho(\nu) = 5$	$\rho(\mu) = 8$	$\rho(;) = 11$
$\rho(\mathcal{O}) = 3$	$\rho(\mathcal{C}) = 6$	$\rho([) = 9$	$\rho(,) = 12$

Abb. 12: Definition der Bijektion ρ

definiert durch die Tabelle in Abbildung 12. Daraus ergibt sich folgendermaßen eine Nummerierung der Wörter über Σ: Es sei $p_i \in \mathbb{P}$ die i-te Primzahl, dann sei die Abbildung

$$g : \Sigma^+ \to \mathbb{N}$$

definiert durch

$$g(x_1 \dots x_n) = p_1^{\rho(x_1)} \cdot \dots \cdot p_n^{\rho(x_n)}$$

Beispiel 4.3 **a)** Betrachten wir als Beispiel (siehe Beispiel 4.1 c) die Funktion

$$id = \pi_1^1$$

Es gilt

$$\sigma(id) = \pi|\#|$$

und

$$g(\pi|\#|) = 2^4 \cdot 3^1 \cdot 5^2 \cdot 7^1 = 8\,400$$

d.h. der Funktion id wird die Nummer $8\,400$ zugeordnet.

b) Betrachten wir als weiteres Beispiel (siehe Beispiel 4.1 b) die Funktion

$$1 = \mathcal{C}\left[\nu; \mathcal{O}^0\right]$$

Es gilt

$$\sigma(1) = \mathcal{C}\left[\nu; \mathcal{O}^0\right]$$

und

$$\begin{aligned} g\left(\mathcal{C}\left[\nu; \mathcal{O}^0\right]\right) &= 2^6 \cdot 3^9 \cdot 5^5 \cdot 7^{11} \cdot 11^3 \cdot 13^{10} \\ &= 1\,428\,273\,264\,576\,872\,238\,279\,737\,182\,200\,000 \end{aligned}$$

d.h. der Funktion 1 wird die Nummer $1\,428\,\dots$ zugeordnet. □

Man sieht, dass durch die Funktion

$$\tau : \mu\mathcal{PR} \to \mathbb{N}$$

definiert durch

$$\tau = g \circ \rho$$

den μ-rekursiven Funktionen (sehr große) Zahlen zugeordnet werden.

Die Funktion τ ist injektiv und damit umkehrbar. Zu einer Zahl $n \in \mathbb{N}$, die die Nummer einer μ-rekursiven Funktion f ist, kann $f = \tau^{-1}(n)$ berechnet werden. n muss in seine Primfaktoren zerlegt werden, diese werden der Größe nach geordnet und gleiche Faktoren werden zu Potenzen zusammengefasst. Diese sogenannte kanonische Faktorisierung natürlicher Zahlen ist eindeutig. Betrachten wir als Beispiel die Zahl

$$172\,821\,065\,013\,801\,540\,831\,848\,199\,046\,200\,000$$

Ihre kanonische Faktorisierung ist

$$2^6 \cdot 3^9 \cdot 5^5 \cdot 7^{11} \cdot 11^5 \cdot 13^{10}$$

woraus sich mithilfe der Tabelle in Abbildung 12 die μ-rekursive Funktion

$$\mathcal{C}\left[\nu;\nu\right]$$

ergibt.

Die Funktion τ ist nicht surjektiv, d.h. ihre Umkehrung τ^{-1} ist nicht total, d.h. es gibt Zahlen $n \in \mathbb{N}$, die nicht Nummer einer μ-rekursiven Funktion sind, wie z.B. die Zahl 30 oder alle Primzahlen größer als 5. 30 hat die Faktorisierung $2^1 \cdot 3^1 \cdot 5^1$, woraus sich gemäß Tabelle 12 die Zeichenkette $|||$ ergibt, die keine μ-rekursive Funktion darstellt. Wir können die Funktion τ^{-1} aber zu einer surjektiven Funktion ergänzen, indem wir allen Zahlen, die nicht Nummer einer μ-rekursiven Funktion sind, irgendeiner μ-rekursive Funktion zuordnen. Als solche Funktion können wir z.B.

$$\omega : \mathbb{N}_0 \to \mathbb{N}_0$$

definiert durch

$$\omega(n) = \perp$$

wählen (siehe auch Übung 4.3 (1)). Es gilt $D(\omega) = \emptyset$, ω ist die „nirgends definierte Funktion“. ω ist μ-rekursiv, denn für

$$f = \mathcal{C}\left[add;\mathcal{C}\left[\nu;\pi_2^2\right],\pi_1^2\right]$$

gilt[32]

$$\omega = \mu\left[f\right]$$

Jetzt können wir folgende total definierte Umkehrung der Funktion τ^{-1} definieren:

$$h : \mathbb{N} \to \mu\mathcal{PR}$$

mit

$$h(i) = \begin{cases} f, & \text{falls } i \in W(\tau) \text{ und } \tau(f) = i \\ \mu\left[\mathcal{C}\left[add;\mathcal{C}\left[\nu;\pi_2^2\right],\pi_1^2\right]\right], & \text{sonst} \end{cases}$$

32 In Pseudocode ist $f(x,y) = \nu(y) + x$.

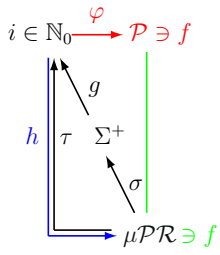


Abb. 13: Nummerierung der partiell berechenbaren Funktionen

Gödelisierung

Die Funktion τ stellt im Übrigen eine sogenannte *Gödelisierung* dar.[33] Gödelisierungen sind total berechenbare, injektive Nummerierungen, deren Inversionen ebenfalls total berechenbar sind.

Mithilfe der obigen Überlegungen bekommen wir eine Nummerierung (siehe Abbildung 13)

$$\varphi : \mathbb{N}_0 \to \mathcal{P}$$

der partiell berechenbaren Funktionen:

$$\varphi(i) = f \text{ genau dann, wenn } h(i) \text{ die Funktion } f \text{ berechnet}$$

Es ist also $\varphi(i) = f$ genau dann, wenn f von der μ-rekursiven Funktion f berechnet wird, die durch die Nummer i codiert ist. Wir unterscheiden hier, was wir ansonsten nicht so streng handhaben, zwischen einer berechenbaren Funktion und ihrer μ-rekursiven Implementierung: die berechenbare Funktion f wird von der μ-rekursiven Funktion f berechnet. So wird z.B. (siehe oben und folgende Besipiele) die Funktion ω durch die Funktion $\mu\left[\mathcal{C}\left[add; \mathcal{C}\left[\nu; \pi_2^2\right], \pi_1^2\right]\right]$, id durch π_1^1 und 1 durch $\mathcal{C}\left[\nu; \mathcal{O}\right]$ berechnet.

Beispiel 4.4 **a)** id wird von $h(8400)$ berechnet (siehe Beispiel 4.3 a): Im Einzelnen ist $\tau(\pi_1^1) = 8400$ und $h(8400) = \pi_1^1$, woraus sich $\varphi(8400) = id$ ergibt (siehe Abbildung 14). Die Funktion id wird also vom Programm π_1^1 mit der Nummer 8400 berechnet.

b) 1 wird von $h(1\,428\,273\,264\,576\,872\,238\,279\,737\,182\,200\,000)$ berechnet (siehe Beispiel 4.3 b): Im Einzelnen ist $\tau(\mathcal{C}\left[\nu; \mathcal{O}\right]) = 1\,428\ldots$ und $h(1\,428\ldots) = \mathcal{C}\left[\nu; \mathcal{O}\right]$ und damit $\varphi(1\,428\ldots) = 1$ (siehe Abbildung 15).

33 Diese Bezeichnung geht auf Kurt Gödel (1906 – 1978) zurück, einem österreichischen Mathematiker, der zu den größten Logikern des vorigen Jahrhunderts gerechnet wird. Er leistete fundamentale Beiträge zur Logik und zur Mengenlehre (im Übrigen mit wesentlicher Bedeutung für die Informatik), außerdem lieferte er interessante Beiträge zur Relativitätstheorie.

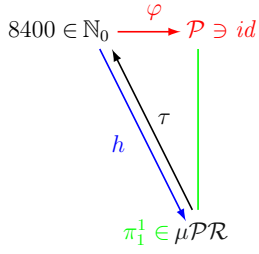


Abb. 14: Nummerierung der Funktion id

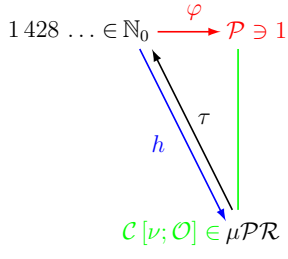


Abb. 15: Nummerierung der Funktion 1

c) 30 kann nicht Nummer einer Funktion sein, denn 30 ist die Codierung von $|||$. Es ist also $h(30) = \mu\left[\mathcal{C}\left[add; \mathcal{C}\left[\nu; \pi_2^2\right], \pi_1^2\right]\right]$ und damit $\varphi(30) = \omega$ (siehe Abbildung 16), weil allen Nummern, die nicht Codierung eines μ-rekursiven Programms sind, die Funktion ω zugeordnet werden soll. □

Im Folgenden schreiben wir, falls $f \in \mathcal{P}$ und f eine k-stellige Funktion von $\mathbb{N}_0^k$ nach $\mathbb{N}_0$ ist, $\varphi_i = f$ anstelle von $\varphi(i) = f$, um „doppelte Argumente“ zu vermeiden: $\varphi_i(x_1, \ldots, x_k) = f(x_1, \ldots, x_k)$ anstelle von $\varphi(i)(x_1, \ldots, x_k) = f(x_1, \ldots, x_k)$.

Nummerierung Programmiersprache

Die Nummerierung $(\mathbb{N}_0, \mathcal{P}, \varphi)$ kann als Programmiersprache aufgefasst werden: Jedes $i \in \mathbb{N}_0$ stellt ein Programm (genauer natürlich eine μ-rekursive Funktion) dar, und φ ordnet jedem Programm (genauer der μ-rekursiven Funktion mit der Nummer) i seine Bedeutung $\varphi_i = f$ zu, nämlich die Funktion, die von diesem

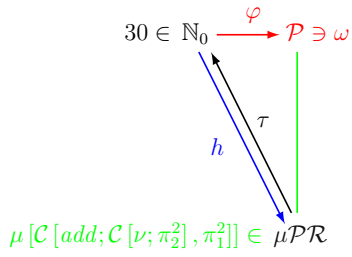


Abb. 16: Zuordnung der Nummer 30 zur Funktion ω

Programm (von dieser μ-rekursiven Funktion) berechnet wird.[34]

4.4.2 Das utm-Theorem

Der folgende Satz, das so genannte *utm-Theorem* besagt, dass die Nummerierung $(\mathbb{N}_0, \mathcal{P}, \varphi)$ die Anforderung (1) von Seite 179 erfüllt.

utm-Theorem

Satz 4.2 Es sei $(\mathbb{N}_0, \mathcal{P}, \varphi)$ eine Nummerierung. Dann gibt es eine berechenbare Funktion $u_\varphi \in \mathcal{P}$ mit

$$u_\varphi(i, x) = \varphi_i(x)$$

□

Universelle Funktion

u_φ heißt *universelle Funktion* von $(\mathbb{N}_0, \mathcal{P}, \varphi)$. Da $u_\varphi \in \mathcal{P}$ ist, gibt es eine μ-rekursive Funktion, die u_φ berechent, d.h. es gibt eine μ-rekursive Funktion $U_\varphi \in \mu\mathcal{PR}$, die alle μ-rekursiven Funktionen $f \in \mu\mathcal{PR}$ berechnet: $U_\varphi(\tau(f), x) = f(x)$.

Die universelle Funktion kann also alle berechenbaren Funktionen berechnen. Das entspricht dem Konzept des universellen Rechners (genauer: eines universellen Programms), der alle Programme ausführen kann. Wenn man sich vorstellt, dass ein Berechenbarkeitskonzept, keine universelle Funktion ermöglicht, bedeutet dies, dass man für jedes Programm einen eigenen Rechner bauen müsste, um dieses auszuführen. Das wäre wahrscheinlich sehr mühsam, und die entsprechende Technologie hätte sicher nicht den Erfolg gehabt, die unsere bekannte IT-Technologie hat, die ganz wesentlich auf der Existenz universeller Maschinen beruht.

34 Nachdem man sich den Unterschied zwischen der μ-rekursiven Funktion $f \in \mu\mathcal{PR}$ mit der Nummer i (es ist also $\tau(f) = i$) und der Funktion $f' \in \mathcal{P}$, die von f berechnet wird (es ist also $\varphi_i = f'$), klar gemacht hat, unterscheidet man im üblichen Sprachgebrauch nicht mehr zwischen f und f' und spricht von der „i-ten berechenbaren Funktion“oder vom „i-ten Programm“oder von der „Funktion mit der Nummer i.“

4.4.3 Das smn-Theorem

Anforderung (2) (siehe Seite 179), die Möglichkeit der effektiven Programmierung soll bedeuten, dass vorhandene Programme zu neuen Programmen zusammengesetzt werden können. Der folgenden Satz, den wir ohne Beweis angeben, belegt, dass in der Programmiersprache $(\mathbb{N}_0, \mathcal{P}, \varphi)$ effektives Programmieren möglich ist.

Satz 4.3 In der Nummerierung $(\mathbb{N}_0, \mathcal{P}, \varphi)$ gibt es eine total berechenbare Funktion $c : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ mit $\varphi_{c(i,j)} = \varphi_i \circ \varphi_j$.

Beweis Seien $h(i) = f$ und $h(j) = f'$ die μ-rekursiven Funktionen mit den Nummern i bzw. j. Dann ist $f'' = \mathcal{C}[f; f']$ ebenfalls eine μ-rekursive Funktion, welche etwa die Nummer $\tau(f'') = k$ hat. Aus den Nummern i und j kann also die Nummer k berechnet werden. Diese Berechnung werde durch die Funktion c geleistet. □

Linker
Binder

Die Funktion c wird in Betriebssystemen durch den *Linker* oder *Binder* realisiert, der Programme mit Unterprogrammen (etwa aus Bibliotheken) zu ausführbaren Programmen zusammenführt.

Eine äquivalente Variante des obigen Satzes beschreibt der folgende Satz, das so genannte *smn-Theorem*.

smn-Theorem

Satz 4.4 Es sei $(\mathbb{N}_0, \mathcal{P}, \varphi)$ eine Nummerierung. Dann gibt es für alle $i \in \mathbb{N}_0$ sowie für alle $(x_1, \ldots, x_m) \in \mathbb{N}_0^m$ und $(y_1, \ldots, y_n) \in \mathbb{N}_0^n$ mit $m, n \geq 1$ eine total berechenbare Funktion $s : \mathbb{N}_0^{m+1} \to \mathbb{N}_0$, so dass

$$\varphi_i(x_1, \ldots, x_m, y_1, \ldots, y_n) = \varphi_{s(i,x_1,\ldots,x_m)}(y_1, \ldots, y_n)$$

gilt. □

Die Parameter $x_1, \ldots, x_m$ können als feste Daten (Konstanten) interpretiert werden. Die Funktion s generiert aus dem Progamm i und diesen Daten das Programm (mit der Nummer) $s(i, x_1, \ldots, x_m)$. Wichtig ist, dass dieser Generator allgemein existiert, d.h. für alle i und alle $(x_1, \ldots, x_m)$.

Betrachten wir z.B. die berechenbare Funktion $f(x_1, x_2, y) = x_1 \cdot y^{x_2}$. Es gibt also ein i mit $\varphi_i = f$. Wenn wir nun etwa $x_1 = 3$ und $x_2 = 4$ festhalten, dann gilt $\varphi_i(3, 4, y) = f(3, 4, y) = 3y^4$, und wir könnten 3 und 4 fest im Programm „verdrahten“, damit wir diese Zahlen – weil sie ja fest bleiben sollen – nicht immer neu eingeben müssen. Das smn-Theorem besagt nun, dass es ein total definiertes Programm s gibt, welches aus dem Programm i und den Konstanten 3 und 4 das Programm $s(i, 3, 4)$ generiert, in das jetzt nur noch y eingegeben werden muss, damit es $3y^4$ berechnet: $f(3, 4, y) = \varphi_i(3, 4, y) = \varphi_{s(i,3,4)}(y)$. Dieses Programm hängt aber nicht von i und 3 und 4 ab, sondern es gilt für alle i, x_1 und x_2. So gilt also z.B. $f(7, 2, y) = \varphi_i(7, 2, y) = \varphi_{s(i,7,2)}(y)$, und s funktioniert auch für eine andere berechenbare Funktion, etwa für $g(x_1, x_2, y) = (x_1 + x_2) \cdot y$: Ist $g = \varphi_j$, dann gilt $g(5, 8, y) = \varphi_j(5, 8, y) = \varphi_{s(j,5,8)}(y)$. Genau so würde auch $g(9, x_2, y) = \varphi_j(9, x_2, y) = \varphi_{s(j,9)}(x_2, y)$ gelten. s ist ein universeller Generator.

Für unsere Programmiersprache $(\mathbb{N}_0, \mathcal{P}, \varphi)$ existiert also ein für alle Programme und Parameter einsetzbarer Binder.

Wenn wir in obigem Satz

$$f(i, x_1, \ldots, x_m, y_1, \ldots, y_n) = \varphi_i(x_1, \ldots, x_m, y_1, \ldots, y_n)$$

setzen, dann ist f eine berechenbare Funktion. Damit können wir den Satz auch wie folgt formulieren.

Folgerung 4.2 Sei $f \in \mathcal{P}$. Dann existiert eine Funktion $t \in \mathcal{R}$, so dass

$$f(i, x_1, \ldots, x_m, y_1, \ldots, y_n) = \varphi_{t(i,x_1,\ldots,x_m)}(y_1, \ldots, y_n)$$

für alle $i, x_1, \ldots, x_m, y_1, \ldots, y_n \in \mathbb{N}_0$ gilt. □

4.4.4 Rekursionssatz und Selbstreproduktionssatz

Die folgenden beiden Sätze sind interessante Folgerungen aus dem utm- und dem smn-Theorem.

Satz 4.5 Zu jeder totalen, berechenbaren Funktion $f \in \mathcal{R}$ existiert eine Zahl $n \in \mathbb{N}_0$ mit $\varphi_{f(n)} = \varphi_n$. **Rekursionssatz**

Beweis $d : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ sei definiert durch $d(i, y) = u_\varphi(u_\varphi(i, i), y)$. Mithilfe des utm-Theorems gilt

$$d(i, y) = u_\varphi(u_\varphi(i, i), y) = u_\varphi(\varphi_i(i), y) = \varphi_{\varphi_i(i)}(y) \tag{4.19}$$

Es gilt $d \in \mathcal{P}$ und gemäß Folgerung 4.2 gibt es ein $t \in \mathcal{R}$ mit

$$d(i, y) = \varphi_{t(i)}(y) \tag{4.20}$$

f und t sind total berechenbar, also ist auch $f \circ t$ total berechenbar. Somit gibt es ein $m \in \mathbb{N}_0$ mit

$$\varphi_m = f \circ t \tag{4.21}$$

und φ_m ist total. Des Weiteren setzen wir (für das feste m):

$$n = t(m) \tag{4.22}$$

Es gilt mit (4.22), (4.20), (4.19) und (4.21)

$$\varphi_n(y) = \varphi_{t(m)}(y) = d(m, y) = \varphi_{\varphi_m(m)}(y) = \varphi_{f(t(m))}(y) = \varphi_{f(n)}(y)$$

womit die Behauptung gezeigt ist. □

Den Rekursionssatz kann man so interpretieren, dass jede total berechenbare Programmtransformation f in $(\mathbb{N}_0, \mathcal{P}, \varphi)$ mindestens ein Programm n in sich selbst transformiert (n also „reproduziert“). Das gilt z.B. auch für den Fall, dass f ein „Computervirus“ ist, der alle Programme verändert. Der Rekursionssatz besagt, dass der Virus mindestens ein Programm unverändert lassen würde.

Eine Folgerung aus dem Rekursionssatz ist der Selbstreproduktionssatz.

Selbstreproduktionssatz

Satz 4.6 Es gibt eine Zahl $n \in \mathbb{N}_0$ mit $\varphi_n(x) = n$ für alle $x \in \mathbb{N}_0$.

Beweis Wir definieren $f : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ durch $f(i, x) = i$. f ist offensichtlich berechenbar. Mit Folgerung 4.2 gibt es somit eine total berechenbare Funktion $t \in \mathcal{R}$ mit $\varphi_{t(i)}(x) = f(i, x) = i$. Nach dem Rekursionssatz gibt es zu t ein $n \in \mathbb{N}_0$ mit $\varphi_n = \varphi_{t(n)}$. Insgesamt folgt $\varphi_n(x) = \varphi_{t(n)}(x) = f(n, x) = n$ für alle $x \in \mathbb{N}_0$, womit die Behauptung gezeigt ist. □

Der Selbstreproduktionssatz besagt, dass es in einer Programmiersprache, welche den Anforderungen (U) und (S) vom Anfang dieses Abschnitts genügt, mindestens ein Programm n gibt, welches unabhängig von den Eingaben x sich selbst („seinen eigenen Quellcode“) ausgibt.

4.5 Aufzählbare und entscheidbare Mengen

In den vorigen Abschnitten haben wir den Begriff der Berechenbarkeit für Funktionen $f : \mathbb{N}_0^k \to \mathbb{N}_0$ einführend betrachtet. Wir werden diesen Begriff nun auf Mengen übertragen, wobei in diesen Zusammenhang von Entscheidbarkeit von Mengen gesprochen wird. Wir werden unterschiedliche Grade von Entscheidbarkeit kennenlernen, und wir werden Zusammenhänge zur Abzählbarkeit von Mengen herstellen, wobei wir dabei von berechenbaren Abzählungen, sogenannten Aufzählungen ausgehen werden. Des Weiteren werden wir nicht berechenbare Funktionen und nicht entscheidbare Mengen kennenlernen.

Lernziele

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- die Begriffe Entscheidbarkeit und Semi-Entscheidbarkeit von Mengen definieren und ihren Zusammenhang erklären können,
- den Begriff der Aufzählbarkeit und seine Äquivalenz zur Semi-Entscheidbarkeit erläutern können,
- den Begriff der Reduktion kennen und auf einfache Mengen anwenden können,
- nicht entscheidbare und nicht semi-entscheidbare Mengen kennen und begründen können, warum diese nicht entscheidbar bzw. nicht semi-entscheidbar sind.

Die genannten Begriffe Entscheidbarkeit, Semi-Entscheidbarkeit und Aufzählbarkeit sind im Allgemeinen für Teilmengen $A \subseteq \mathbb{N}_0^k$, $k \geq 1$, definiert. Wir werden die folgenden Betrachtungen zumeist auf den Fall $k = 1$ beschränken, wobei wir im Hinterkopf behalten, dass wir mit den Cantorschen Tupelfunktionen (siehe Kapitel 2.3) eineindeutige und berechenbare Abbildungen zwischen

$\mathbb{N}_0^k$ und $\mathbb{N}_0$ zur Verfügung haben, so dass wir die im Folgenden dargestellten Definitionen von und Aussagen über Teilmengen von und Funktionen auf $\mathbb{N}_0$ auf Teilmengen von bzw. Funktionen auf $\mathbb{N}_0^k$ übertragen können.

Des Weiteren werden wir im Folgenden, wenn wir begründen wollen, dass eine Funktion f berechenbar ist, dies nicht immer dadurch tun, dass wir eine μ-rekursive Funktion angeben, die f berechnet, sondern wir werden dann (informell) ein Programmiersprachen-ähnlich formuliertes Berechnungsverfahren für f angeben. Die Churchsche These (siehe Kapitel 4.3) besagt, dass diese beiden Berechenbarkeitskonzepte äquivalent sind, d.h. wenn f μ-rekursiv ist, dann gibt es dafür auch ein Programm, und umgekehrt, wenn f durch ein Programm berechnet werden kann, dann ist f auch μ-rekursiv.

4.5.1 Entscheidbare und semi-entscheidbare Mengen

Definition 4.4 Es sei $A \subseteq \mathbb{N}_0$. Dann heißt

a) die Funktion $\chi_A : \mathbb{N}_0 \to \{0,1\}$ definiert durch

Charakteristische Funktion

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \tag{4.23}$$

charakteristische Funktion von A.

b) die Funktion $\chi'_A : \mathbb{N}_0 \to \{0,1\}$ definiert durch

Semi-charakteristische Funktion

$$\chi'_A(x) = \begin{cases} 1, & x \in A \\ \perp, & x \notin A \end{cases} \tag{4.24}$$

semi-charakteristische Funktion von A.

c) A *entscheidbar* oder auch *rekursiv* genau dann, wenn χ_A berechenbar ist.

Entscheidbare, rekursive Menge

d) A *semi-entscheidbar* genau dann, wenn χ'_A berechenbar ist. □

Semi-entscheidbare Menge

Der folgende Satz gibt einen wichtigen Zusammenhang zwischen Entscheidbarkeit und Semi-Entscheidbarkeit einer Menge an, den wir im Folgenden noch verwenden werden.

Satz 4.7 **a)** Eine Menge $A \subseteq \mathbb{N}_0$ ist entscheidbar genau dann, wenn ihr Komplement $\overline{A} = \mathbb{N}_0 - A$ entscheidbar ist.

b) Eine Menge $A \subseteq \mathbb{N}_0$ ist entscheidbar genau dann, wenn A und $\overline{A}$ semi-entscheidbar sind.

Beweis **a)** „⇒“: Sei A entscheidbar, dann ist χ_A berechenbar, dann ist auch $1 - \chi_A$ berechenbar, und diese Funktion ist gleich $\chi_{\overline{A}}$.

„⇐“: Analog.

b) „⇒“: Sei A entscheidbar. Dann ist χ_A brechenbar. Sei f_{χ_A} die μ-rekursive Funktion, die χ_A berechnet. Wir verwenden die Funktion $minus(x, y) = x - y$ aus Beispiel 4.1 h) auf Seite 169 und definieren

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

Es gilt $g(1, 0) = 1$ und $g(1, y) = 0$ für $y \geq 1$ sowie $g(0, y) \neq 0$ für alle $y \geq 0$. Daraus folgt, dass

$$\mu[g](f_{\chi_A}(x)) = \left\{ \begin{array}{ll} 1, & \chi_A(x) = 1 \\ \\ \bot, & \chi_A(x) = 0 \end{array} \right\} = \left\{ \begin{array}{ll} 1, & x \in A \\ \\ \bot, & x \notin A \end{array} \right.$$

ist. Es gilt damit $\chi'_A = \mu[g]$, und damit ist χ'_A berechenbar, und damit ist A semi-entscheidbar.

Jetzt definieren wir

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

Es gilt $g(1, y) \neq 0$ für alle $y \geq 0$ und $g(0, 0) = 1$ und $g(0, y) = 0$ für alle $y \geq 1$. Daraus folgt, dass

$$\mu[g](f_{\chi_A}(x)) = \left\{ \begin{array}{ll} 1, & \chi_A(x) = 0 \\ \\ \bot, & \chi_A(x) = 1 \end{array} \right\} = \left\{ \begin{array}{ll} 1, & x \notin A \\ \\ \bot, & x \in A \end{array} \right\} = \left\{ \begin{array}{ll} 1, & x \in \overline{A} \\ \\ \bot, & x \notin \overline{A} \end{array} \right.$$

ist. Es gilt damit $\chi'_{\overline{A}} = \mu[g]$, und damit ist $\chi'_{\overline{A}}$ berechenbar, und damit ist $\overline{A}$ semi-entscheidbar. □

„⇐“: Seien nun A und $\overline{A}$ semi-entscheidbar. Dann sind χ_A und $\chi_{\overline{A}}$ berechenbar, d.h. es gibt zwei Algorithmen T_A und $T_{\overline{A}}$, die χ_A bzw. $\chi_{\overline{A}}$ berechnen. Die parallele Ausführung beider Algorithmen für eine Eingabe $x \in \mathbb{N}_0$ berechnet χ_A: T_A hält an, wenn $x \in A$ ist, und $T_{\overline{A}}$ hält an, wenn $x \notin A$ ist. Wenn T_A anhält, gibt der Algorithmus eine 1 aus und stoppt den Algorithmus $T_{\overline{A}}$, wenn $T_{\overline{A}}$ anhält, gibt der Algorithmus eine 0 aus und stoppt den Algorithmus T_A. Der parallele Algorithmus stoppt also bei jeder Eingabe und gibt immer die korrekte Ausgabe aus. □

4.5.2 Aufzählbare Mengen

Aufzählbare Menge

Definition 4.5 Eine Menge $A \subseteq \mathbb{N}_0$ heißt *aufzählbar* genau denn, wenn $A = \emptyset$ ist oder wenn eine total berechenbare Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ existiert mit $A = W(f)$. □

Ist eine Menge $A \neq \emptyset$ aufzählbar, dann können ihre Elemente mit einer total berechenbaren Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ aufgezählt werden.:

$$A = \{f(0), f(1), f(2), \ldots\}$$

Gilt $f(i) = a$, dann sagen wir „a hat die Nummer i“ und schreiben dafür auch a_i. Die Aufzählung von A mit f kann also auch durch $A = \{a_0, a_1, a_2, \ldots\}$ notiert werden. Da f nicht injektiv sein muss, kann ein Element von A mehrere (sogar unendlich viele) Nummern bekommen.

Der folgende Satz besagt, dass die Klasse der semi-entscheidbaren Mengen genau die Klasse der rekursiv-aufzählbaren Mengen ist.

Satz 4.8 Eine Menge A ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Beweis „⇒“: Sei A rekursiv aufzählbar. Ist $A = \emptyset$, dann ist $\chi'_A = \omega$ (die nirgends definierte Funktion, siehe Abschnitt 4.4.1), eine berechenbare semi-charakteristische Funktion für A. Ist $A \neq \emptyset$ dann existiert eine total berechenbare Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ mit $A = W(f)$. Wir setzen $g(x, i) = f(i) - x$, damit ist g berechenbar, und die Funktion $\mu[g]$ ist berechenbar. Bei Eingabe von x berechnet $\mu[g](x)$ die kleinste Zahl i, so dass $f(i) = x$ ist, falls x eine Nummer unter f hat, d.h. falls $x \in A$ ist. Falls $x \notin A$ ist, dann gibt es keine Nummer i mit $f(i) = x$, dann ist $\mu[g](x)$ nicht definiert. Es folgt $\chi'_A = \mu[g]$, und damit ist A semi-entscheidbar.

„⇐“: Sei A semi-entscheidbar. Ist $A = \emptyset$, dann ist A per definitonem rekursiv aufzählbar. Ist $A \neq \emptyset$, dann enthält A mindestens ein Element a. Da A semi-entscheidbar ist, ist χ'_A berechenbar; es sei $T_{\chi'_A}$ der Algorithmus der χ'_A berechnet. Wir müssen uns einen Algorithmus T_f überlegen, der jeder Zahl $i \in \mathbb{N}_0$ ein Element $x \in A$ zuordnet und der jedem $x \in A$ auch tatsächlich eine Zahl $i \in \mathbb{N}_0$ zuordnet. Dann berechnet T_f eine totale Aufzählungsfunktion f von A. Der in Abbildung 17 dargestellte Algorithmus T_f betrachtet eine Eingabe $i \in \mathbb{N}_0$ zunächst als Ergebnis der Anwendung der Cantorfunktion c_2, d.h. T_f berechnet aus i die Zahlen $x, k \in \mathbb{N}_0$ mit $c_2(x, k) = i$. Da c_2 bijektiv und berechenbar ist, sind diese beiden Umkehrungen $x = c_{21}^{-1}(i)$ und $k = c_{22}^{-1}(i)$ berechenbar, und das Paar (x, k) ist eindeutig für i. T_f führt nun den Algorithmus $T_{\chi'_A}$ k Schritte auf x aus. Falls $T_{\chi'_A}$ innerhalb dieser Schrittzahl anhält, dann gibt T_f x aus, falls nicht, dann gibt T_f a aus.

Wir verifizieren noch, dass der Algorithmus T_f alle Anforderungen erfüllt, damit die davon berechnete Funktion eine rekursive aufzählung von A berechnet: Für die Eingabe i sei also $x, k \in \mathbb{N}_0$ so, dass $c_2(x, k) = i$ gilt. Ist $x \notin A$, dann stoppt $T_{\chi'_A}$ bei Eingabe x nicht, also auch nicht in k Schritten. Der Zahl i wird also das vorher fest gewählte Element $a \in A$ zugeordnet. Ist $x \in A$ und stoppt $T_{\chi'_A}$ bei Eingabe x innerhalb von k Schritten, dann wird i dem Element x zugeordnet, andernfalls wird i dem Element a zugeordnet. Wegen der Bijektivität von c_2 gibt es zu jedem $x \in A$ ein k und ein i mit $c_2(x, k) = i$. Damit ist gesichert, dass jedes Element $x \in A$ nummeriert wird. Insgesamt folgt, dass der Algorithmus T_f eine rekursive Aufzählung f von A berechnet. □

```
read(i);
 x := c_21^{-1}(i);
 k := c_22^{-1}(i);
 if T_{χ'_A} stoppt bei Eingabe x in ≤ k Schritten
    then f := x
    else f := a
 endif;
write(f)
```

Abb. 17: Algorithmus T_f zur Berechnung der Aufzählung f von A bei gegebenem χ'_A

4.5.3 Reduzierbarkeit von Mengen

Wenn wir im nächsten Abschnitt konkrete Mengen auf Entscheidbarkeit oder Semi-Entscheidbarkeit untersuchen, kann es hilfreich sein, wenn man die Frage nach der (Semi-) Entscheidbarkeit einer Menge A auf die (Semi-) Entscheidbarkeit einer anderen Menge B zurückführen kann, und zwar in folgendem Sinne: Ist die Frage für B geklärt, lässt sich daraus die Frage für A beantworten.

Reduktion

Definition 4.6 Seien $A, B \subseteq \mathbb{N}_0$. A heißt *reduzierbar* auf B genau dann, wenn es eine totale berechenbare Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ gibt mit

$$x \in A \text{ genau dann, wenn } f(x) \in B \text{ für alle } x \in \mathbb{N}_0$$

Ist A reduzierbar auf B (mittels der totalen berechenbaren Funktion f), so schreiben wir $A \leq B$ (oder $A \leq_f B$, falls die Funktion, mit der die Reduktion vorgenommen wird, genannt werden soll). □

Gilt $A \leq B$, dann können wir die Entscheidbarkeit von A auf B (algorithmisch) transformieren: Ist B entscheidbar, d.h. gibt es ein Entscheidungsverfahren für B, dann ist auch A entscheidbar. Um zu entscheiden, ob $w \in A$ ist, berechnen wir $f(w)$ und wenden auf das Ergebnis das Entscheidungsverfahren für B an. Ist $f(w) \in B$, dann ist $w \in A$, ist $f(w) \notin B$, dann ist $w \notin A$. Der folgende Satz fasst diese Überlegung zusammen.

Satz 4.9 Es seien $A, B \subseteq \mathbb{N}_0$. Ist $A \leq B$ und ist B entscheidbar oder semi-entscheidbar, dann ist auch A entscheidbar bzw. semi-entscheidbar.

Beweis Wir führen den Beweis nur für den Fall der Entscheidbarkeit, der Beweis für den Fall der Semi-Entscheidbarkeit erfolgt analog. Es gelte also: $A \leq_f B$, f ist total berechenbar, und B ist entscheidbar. Da B entscheidbar ist, ist χ_B, die charakteristische Funktion von B, berechenbar. Wir definieren die Funktion $\chi_A : \mathbb{N}_0 \to \{0, 1\}$ durch

$$\chi_A(x) = \chi_B(f(x))$$

Es gilt:

- χ_A ist die charakteristische Funktion von A, denn es ist

$$\chi_B(f(x)) = \left\{ \begin{array}{ll} 1, & f(x) \in B \\ 0, & f(x) \notin B \end{array} \right\} = \left\{ \begin{array}{ll} 1, & x \in A \\ 0, & x \notin A \end{array} \right\} = \chi_A(x)$$

- χ_A ist berechenbar, denn f und χ_B sind berechenbar.

Mit der Reduktion f und der charakteristischen Funktion von B können wir also eine berechenbare charakteristische Funktion für A angeben. A ist also entscheidbar. □

Wir werden den Satz 4.9 im nächsten Abschnitt im „negativen Sinne“ verwenden: Wenn wir gezeigt haben, dass eine Sprache A nicht entscheidbar ist, und wenn wir zeigen können, dass $A \leq B$ für die Sprache B gilt, wissen wir, dass auch B unentscheidbar ist. Es gilt also die

Folgerung 4.3 Es seien $A, B \subseteq \mathbb{N}_0$. Ist $A \leq B$ und ist A nicht entscheidbar (nicht semi-entscheidbar), dann ist auch B nicht entscheidbar (nicht semi-entscheidbar). □

4.6 Unentscheidbare Mengen

Im Beispiel 3.5 b) auf Seite 138 haben wir gezeigt, dass die Menge der totalen Funktionen von $\mathbb{N}$ in sich überabzählbar ist. Aus Abschnitt 4.4.1 wissen wir, dass die Menge der berechenbaren Funktionen abzählbar ist, denn wir konnten für diese Menge eine Nummerierung angeben. Es folgt, dass es prinzipiell Probleme geben muss, die nicht berechenbar sind. In diesem Abschnitt und in den folgenden Abschnitten werden wir konkrete nicht berechenbare Probleme angeben. Dabei werden wir diese Probleme als Sprachen formulieren und zeigen, dass diese Sprachen nicht entscheidbar sind. Für die entsprechenden Betrachtungen legen wir die im Abschnitt 4.4.1 eingeführte Nummerierung $(\mathbb{N}_0, \mathcal{P}, \varphi)$ zugrunde.

4.6.1 Das Halteproblem

Selbstanwendbarkeitsproblem

Wir betrachten zunächst ein spezielles Halteproblem, auch *Selbstanwendbarkeitsproblem* genannt, und formulieren es als Menge:

$$K = \{\, i \mid i \in \mathit{Def}(\varphi_i) \,\}$$

K enthält die Nummern aller berechenbaren Funktionen, die – auf sich selbst angewendet – ein Ergebnis liefern (deswegen Selbstanwendbarkeitsproblem).

Nicht als Entscheidungsproblem der Menge K, sondern als Problemspezifikation formuliert, kann das Selbstanwendbarkeitsproblem formuliert werden mithilfe der charakteristischen Funktion von K:

$$\chi_K : \mathbb{N}_0 \to \{0, 1\}$$

definiert durch

$$\chi_K(i) = \begin{cases} 1, & \text{falls } \varphi \text{ auf } i \text{ definiert ist} \\ 0, & \text{sonst} \end{cases}$$

Die Frage ist dann, ob χ_K berechenbar ist.

Da wir die Nummern i auch als Programme betrachten, lautet diese Fragestellung entsprechend: Gibt es einen Algorithmus, der für alle Programme testen kann, ob diese bei Eingabe ihres eigenen Programmcodes anhalten oder nicht. Der folgende Satz verneint diese Frage.

Satz 4.10 K ist nicht entscheidbar.

Beweis Wir nehmen an, K sei entscheidbar. Dann ist χ_K berechenbar, d.h. es gibt eine Nummer p mit $\chi_K = \varphi_p$. Wir definieren die Funktion g durch

$$g(j) = \begin{cases} 1, & \text{falls } \varphi_p(j) = 0 \\ \bot, & \text{falls } \varphi_p(j) = 1 \end{cases}$$

g ist berechenbar, weil φ_p berechenbar ist, also gibt es ein q mit $g = \varphi_q$. Damit gilt:

$$\begin{aligned} \varphi_q(q) = 1 \quad & \text{genau dann, wenn} \quad g(q) = 1 \\ & \text{genau dann, wenn} \quad \varphi_p(q) = 0 \\ & \text{genau dann, wenn} \quad \chi_K(q) = 0 \\ & \text{genau dann, wenn} \quad q \notin K \\ & \text{genau dann, wenn} \quad q \notin \mathit{Def}(\varphi_q) \\ & \text{genau dann, wenn} \quad \varphi_q(q) = \bot \end{aligned}$$

Damit haben wir die Aussage

$$\varphi_q(q) = 1 \text{ genau dann, wenn } \varphi_q(q) = \bot$$

hergeleitet, die offensichtlich einen Widerspruch darstellt. Unsere Annahme, dass K entscheidbar ist, muss also falsch sein. □

Möglicherweise erscheint das Selbstanwendbarkeitsproblem auf den ersten Blick als ein künstliches Problem. Hinsichtlich Programmen, die man auf sich selbst anwenden kann, denke man z.B. an ein Programm, das die Anzahl der Buchstaben in einer Zeichenkette zählt. Es kann durchaus Sinn machen, das Programm auf sich selbst anzuwenden, um festzustellen, wie lang es ist.

Das Selbstanwendbarkeitsproblem ist also nicht entscheidbar, es ist aber semientscheidbar, d.h. die berechenbaren Funktionen (alle Programme) können rekursiv aufgezählt werden.

Satz 4.11 K ist rekursiv-aufzählbar.

Beweis Wir zeigen, dass K semi-entscheidbar ist, woraus mit Satz 4.8 folgt, dass K rekursiv-aufzählbar ist. K ist semi-entscheidbar, wenn χ'_K berechenbar ist. Wir setzen

$$\chi'_K(i) = \mathit{sign}(u_\varphi(i,i) + 1)$$

Diese Funktion ist berechenbar, denn u_φ, die universelle Funktion von $(\mathbb{N}_0, \mathcal{P}, \varphi)$, ist berechenbar, und die Signumfunktion ist berechenbar (siehe Beispiel 4.1 j). Falls $i \in \mathit{Def}(\varphi_i)$ ist, ist $u_\varphi(i,i) + 1 \geq 1$ und damit $\mathit{sign}(u_\varphi(i,i) + 1) = 1$. Falls $i \notin \mathit{Def}(\varphi_i)$ ist, dann ist $u_\varphi(i,i)$ und damit auch $\mathit{sign}(u_\varphi(i,i) + 1)$ nicht definiert. $\mathit{sign}(u_\varphi(i,i) + 1)$ implementiert also $\chi'_K(i)$. K ist also semi-entscheidbar und damit rekursiv-aufzählbar. □

Da K rekursiv-aufzählbar, aber nicht entscheidbar sind, folgt wegen Satz 4.7 b) die Gültigkeit des nächsten Satzes.

Satz 4.12 Das Komplement von K

$$\overline{K} = \{\, i \mid i \notin \mathit{Def}(\varphi_i) \,\}$$

ist nicht rekursiv-aufzählbar. □

Wir wollen nun das (allgemeine) Halteproblem betrachten. Es ist definiert durch die Menge **Halteproblem**

$$H = \{\, (i,j) \mid j \notin \mathit{Def}(\varphi_i) \,\}$$

Satz 4.13 H ist nicht entscheidbar.

Beweis Wir beweisen den Satz, indem wir K auf H reduzieren; K ist ein Spezialfall von H: Wäre der allgemeine Fall H entscheidbar, dann müsste auch der Spezialfall K entscheidbar sein (siehe Satz 4.9 bzw. Folgerung 4.3). Für die Funktion

$$f : \mathbb{N}_0 \to \mathbb{N}_0 \text{ definiert durch } f(i) = (i,i)$$

gilt: f ist total berechenbar, und es gilt $i \in K$ genau dann, wenn $(i,i) \in H$, d.h. genau dann, wenn $f(i) \in H$ gilt. Also ist $K \leq_f H$, woraus die Unentscheidbarkeit von H folgt. □

Übungsaufgaben

4.4 Zeigen Sie: H ist rekursiv aufzählbar! □

Wir setzen $\chi'_K(i,j) = \mathit{sign}(u_\varphi(i,j) + 1)$, und der Beweis erfolgt vollkommen analog zum Beweis von Satz 4.11.

4.6.2 Der Satz von Rice

Syntaktische Eigenschaften von Programmen, wie z.B. die Länge eines Programms, die Anzahl der Variablen oder die Anzahl der Zuweisungen, in denen eine Variable vorkommt, sind sicher entscheidbar. Wie steht es um die Entscheidbarkeit von semantischen Eigenschaften? Wir wissen schon, dass das Halteproblem nicht entscheidbar ist. Vielleicht gibt es aber Eigenschaften, die für alle Programme beweisbar sind, wie z.B.

(1) $E_1 = \{ f \in \mathcal{P} \mid f(x) = 3, \text{ für alle } x \in \mathbb{N}_0 \}$ enthält alle berechenbaren Funktionen, die für jede Eingabe den Wert 3 ausgeben („Konstante 3“).

(2) $E_2 = \{ f \in \mathcal{P} \mid f(4) = 17 \}$ ist die Menge aller berechenbaren Funktionen, die für die Eingabe 4 den Wert 17 ausgeben.

(3) $E_3 = \{ f \in \mathcal{P} \mid f(x) = x, \text{ für alle } x \in \mathbb{N}_0 \} = \{ f \in \mathcal{P} \mid f \equiv id \}$ enthält alle berechenbaren Funktionen, die die Identität darstellen.

(4) $E_4 = \{ f \in \mathcal{P} \mid \mathit{Def}(f) = \mathbb{N}_0 \}$ ist die Menge aller totalen berechenbaren Funktionen.

(5) $E_5 = \{ f \in \mathcal{P} \mid f \equiv g \}$, $g \in \mathcal{P}$, enthält alle berechenbaren zu g identischen Funktionen (siehe Korrektheitsproblem).

Der Satz von Rice,[35] den wir im Folgenden vorstellen, besagt, dass alle semantischen Eigenschaften von Programmen außer den beiden trivialen nicht entscheidbar sind.

Definition 4.7 Es sei $(\mathbb{N}_0, \mathcal{P}, \varphi)$ die in Abschnitt 4.4.1 eingeführte Nummerierung.

Indexmenge

a) Sei $E \subseteq \mathcal{P}$, dann ist $A_E = \{ i \mid \varphi_i \in E \}$ die *Indexmenge* von E.

Triviale Indexmenge

b) Die Indexmengen $A_E = \emptyset$ und $A_E = \mathbb{N}_0$ der Mengen $E = \emptyset$ bzw. $E = \mathcal{P}$ heißen *trivial*. □

Satz von Rice

Satz 4.14 Jede nicht triviale Indexmenge ist nicht entscheidbar.

Beweis Sei $E \subseteq \mathcal{P}$ mit $E \neq \emptyset$ und $E \neq \mathcal{P}$. Da $E \neq \emptyset$ ist, gibt es mindestens ein p mit $\varphi_p \in E$, und da $E \neq \mathcal{P}$ ist, gibt es mindestens ein q mit $\varphi_q \notin E$. Es ist also $p \in A_E$ und $q \notin A_E$. Wir nehmen an, dass A_E entscheidbar ist. Dann ist die charakteristische Funktion χ_{A_E} von A_E berechenbar, und damit ist die Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch

$$f(x) = \begin{cases} q, \chi_{A_E}(x) = 1 \\ p, \chi_{A_E}(x) = 0 \end{cases}$$

total berechenbar. f erfüllt somit die Voraussetzungen des Rekursionssatzes 4.5, also gibt es ein $n \in \mathbb{N}_0$ mit $\varphi_n = \varphi_{f(n)}$. Für dieses n gilt

$$\varphi_n \in E \text{ genau dann, wenn } \varphi_{f(n)} \in E \tag{4.25}$$

35 Henry G. Rice (geb. 1920) ist ein amerikanischer Logiker und Mathematiker, der diesen Beweis in seiner Disseration führte.

gilt. Andererseits gilt wegen der Definition von f: $x \in A_E$ genau dann, wenn $f(x) = q \notin A_E$, d.h. $\varphi_x \in E$ genau dann, wenn $\varphi_{f(x)} \notin E$ ist. Dieses gilt natürlich auch für $x = n$, was einen Widerspruch zu (4.25) bedeutet. Damit muss unsere Annahme, dass A_E entscheidbar ist, falsch sein, d.h. für $E \neq \emptyset$ und $E \neq \mathcal{P}$ ist $A_E = \{ i \mid \varphi_i \in E \}$ nicht entscheidbar. □

4.6.3 Das Korrektheitsproblem

Für das Programmieren wäre es von großer praktischer Bedeutung, wenn man die Korrektheit von beliebigen Programmen mithilfe eines automatischen Programmbeweisers nachweisen könnte. Wenn also eine Funktion f gegeben ist und man ein Programm konstruiert hat, das f berechnen soll, wäre es nützlich, wenn ein universeller Programmbeweiser zur Verfügung stünde, in den man die Problemspezifikation f und das zu ihrer Berechnung konstruierte Programm eingibt, und der Programmbeweiser „ja“ ausgibt, falls das Programm f berechnet, und anderenfalls „nein“ ausgibt. Der folgende Satz besagt, dass es einen solchen universellen Programmbeweiser nicht geben kann.

Satz 4.15 Sei $f \in \mathcal{P}$, dann ist die Indexmenge

$$P_f = \{ i \mid \varphi_i = f \}$$

nicht entscheidbar.

Beweis Wenn wir $E = \{ f \}$ wählen, ist die Voraussetzung von Satz 4.14 erfüllt. Es gilt also $A_E = \{ i \mid \varphi_i = f \} = \{ i \mid \varphi_i \in E \}$, und mit Satz 4.14 ist A_E nicht entscheidbar. □

P_f ist nicht entscheidbar, d.h. gegeben ein Programm i, dann ist es nicht entscheidbar, ob i die Funktion f berechent, das Programm i also korrekt ist, oder ob $i \notin P_f$ ist, d.h. ob f nicht von i berechnet wird, das Programm i also nicht korrekt ist.

Da also *im Nachhinein* ein (automatischer) Korrektheitsbeweis nicht möglich ist, und da das Testen von Programmen dem Finden von Fehlern dient und nicht die Korrektheit von Programmen garantiert, müssen Programmiererinnen und Programmierer *während* der Konstruktion von Programmen für Korrektheit sorgen. Mit Methoden und Verfahren zur Konstruktion von korrekten Programmen beschäftigt sich die *Programmverifikation*.

4.6.4 Das Äquivalenzproblem

Ein weiteres Problem mit praktischer Bedeutung ist das Äquivalenzproblem. Angenommen, es stehen zwei Programme zur Verfügung, z.B. unterschiedliche Releases einer Software, die ein Problem lösen sollen. Eine interessante Frage ist die nach der Äquivalenz der Programme, d.h. ob sie dasselbe Problem berechnen.

Gäbe es einen Äquivalenzbeweiser, dann könnte man ihn zu Hilfe nehmen, um diese Frage zu entscheiden. Im positiven Falle könnte man dann aufgrund weiterer Qualitätskriterien (u.a. Effizienz, Benutzerfreundlichkeit) eines der äquivalenten Programme auswählen. Der folgende Satz besagt, dass es keinen universellen Äquivalenzbeweiser geben kann.

Satz 4.16 Die Menge

$$A = \{ (i,j) \mid \varphi_i = \varphi_j \}$$

ist nicht entscheidbar.

Beweis Wir reduzieren für ein $f \in \mathcal{P}$ die bereits in Satz 4.15 als unentscheidbar gezeigte Indexmenge P_f auf die Indexmenge A, woraus mit Folgerung 4.3 die Behauptung folgt (P_f ist ein Spezialfall von A).

Da $f \in \mathcal{P}$ ist, existiert ein $i_f \in \mathbb{N}_0$ mit $\varphi_{i_f} = f$. Wir definieren die Funktion $g : \mathbb{N}_0 \to \mathbb{N}_0$ durch $g(j) = (j, i_f)$. g ist offensichtlich total berechenbar. Es gilt

$$\begin{aligned} j \in \mathcal{P}_f &\text{ genau dann, wenn } \varphi_j = f \\ &\text{genau dann, wenn } \varphi_j = \varphi_{i_f} \\ &\text{genau dann, wenn } (j, i_f) \in A \\ &\text{genau dann, wenn } g(j) \in A \end{aligned}$$

woraus $P_f \leq_g A$ und damit die Behauptung folgt. □

4.7 Zusammenfassung

Auf der Basis der rekursiven Definition der natürlichen Zahlen kann der Begriff der Berechenbarkeit einer Funktion $f : \mathbb{N}_0^k \to \mathbb{N}_0$ definiert werden. Als berechenbar werden sehr einfache, elementare Funktionen (Grundfunktionen) angesehen wie die Nullfunktion, die Projektionen und die Nachfolgerfunktion. Mithilfe von Operatoren wie Komposition, primitive Rekursion und μ-Rekursion werden aus berechenbaren Funktionen neue berechenbare Funktionen konstruiert. Eine Funktion $f : \mathbb{N}_0^k \to \mathbb{N}_0$ wird als berechenbar bezeichnet, falls es eine μ-rekursive Funktion gibt, die f berechnet.

Mithilfe dieser funktionalen Programmierung kann man sich schrittweise eine Sammlung von berechenbaren Funktionen (eine „Programmbibliothek“) schaffen und bei Bedarf erweitern. Die vorhandenen Programme können wie Bausteine zu neuen Bausteinen zusammengesetzt werden.

Neben den μ-rekursiven Funktionen gibt es weitere, davon und untereinander verschiedene Ansätze, den Begriff der Berechenbarkeit zu definieren

(z.B. Turingmaschinen, Universelle Registermaschinen, Goto-Programme, While-Programme, Markov-Algorithmen, λ-Kalkül). Man kann zeigen, dass alle diese Konzepte äquivalent sind, d.h. sie beschreiben alle dieselbe Klasse von Funktionen, nämlich die Klasse $\mathcal{P}$ der partiell berechenbaren Funktionen. Dies ist die Begründung für die Churchsche These, die besagt, dass $\mathcal{P}$ mit der Klasse der Funktionen übereinstimmt, die man intuitiv als berechenbar ansieht.

Die Klasse $\mathcal{PR}$ der primitiv-rekursiven Funktionen besteht aus den Funktionen, die sich aus den Grundfunktionen und der Anwendung von Komposition und primitver Rekursion ergibt. Diese Funktionen sind alle total definiert. Es gibt Funktionen, wie die Ackermannfunktion, die total berechenbar sind, aber nicht primitv-rekursiv. Wenn $\mathcal{R}$ die Klasse der total berechenbaren Funktionen ist, dann gilt $\mathcal{PR} \subset \mathcal{R} \subset \mathcal{P}$.

Die μ-rekursiven Funktionen erfüllen zwei wesentliche Anforderungen, die an Berechenbarkeitskonzepte sinnvollerweise gestellt werden können: die Existenz einer berechenbaren universellen Funktion und die Möglichkeit der effektiven Programmierung.

Eine berechenbare universelle Funktionen kann alle anderen berechenbaren Funktionen berechnen, sie ist die Grundlage für die Existenz universeller Rechner, welche alle Programme ausführen können.

Effektives Programmieren bedeutet unter anderem die Existenz eines universellen Binders, d.h. eines Generators der existierende Programme zu neuen Programmen zusammensetzt.

Eine Menge natürlicher Zahlen heißt entscheidbar, wenn ihre charakteristische Funktion berechenbar ist, sie heißt semi-entscheidbar, wenn ihre semi-charakteristische Funktion berechenbar ist, und sie heißt aufzählbar, wenn sie der Wertebereich einer total berechenbaren Funktion ist. Eine Menge ist semi-entscheidbar genau dann, wenn sie rekursiv aufzählbar ist. Mithilfe einer Reduktion kann die Frage nach der (Semi-) Entscheidbarkeit einer Menge auf die (Semi-) Entscheidbarkeit einer anderen Menge übertragen werden.

Beispiele für nicht entscheidbare Mengen sind das Selbstanwendbarkeits-, das Halte-, das Korrektheits- und das Äquivalenzproblem. Diese Beispiele sind nicht nur von theoretischer, sondern auch von großer praktischer Bedeutung. Der Satz von Rice besagt, dass es für keine nicht triviale semantische Eigenschaft von Programmen einen universellen Beweiser geben kann.

Lösungen zu den Aufgaben

Aufgabe 1.1

Zu a): (1) $\emptyset \notin \emptyset$, da die leere Menge kein Element enthält. (2) $\emptyset \in \{\emptyset\}$ ist offensichtlich. (3) $M \in \{\{1,2\}, M\}$ offensichtlich. (4) $1 \notin \{\{1,2\}, M\}$ offensichtlich. (5) $3 \in A = \{3, 4, \{5,6\}\}$ offensichtlich. (6) $\{4\} \notin A$, denn A enthält zwar das Element 4, aber nicht die Menge $\{4\}$ als Element. (7) $\{5,6\} \in A$ offensichtlich. (8) $\{B\} \notin A$, denn A enthält nicht die Menge $\{B\} = \{\{5,6\}\}$ als Element. (9) ist die gleiche Aussage wie (7). (10) $6 \notin A$ offensichtlich. (11) $6 \in B$ offensichtlich. (12) $\{5,6\} \notin B$, denn B enthält nicht die Menge $\{5,6\}$ als Element, sondern die beiden Elemente 5 und 6.

Zu (b): (1) falsch, da die leere Menge kein Element enthält. (2) offensichtlich wahr. (3) wahr, denn die rechte Menge enthält die Menge $\{a,b\}$ als Element. (4) falsch, den die rechte Menge enthält die Menge $\{a,b,c\}$ nicht als Element. (5) und (6) sind offensichtlich wahr.

Aufgabe 1.2

(1) $M_1 = \{0,1,8,27,64\}$. (2) $M_2 = \{15,18,21,27,30,33,39,42,45\}$.
(3) $M_3 = \{\{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$.
(4) $M_4 = \{0,1,\ldots,10\}$. (5) $M_5 = \{1,2,3,4,6,8,12,24\}$.
(6) $M_6 = \emptyset$. (7) $M_7 = \{1,2,3\}$.

Aufgabe 1.3

(1) $M_1 = \{x \mid x = 2^k,\ 0 \leq k \leq 6\}$.
(2) $M_2 = \{x \mid x = k^2,\ 2 \leq k \leq 11 \text{ und } k \text{ ist prim}\}$.
(3) $M_3 = \{x \mid x = k^3,\ 1 \leq k \leq 4\}$.
(4) $M_4 = \{x \mid x = 2k+1,\ k \text{ ist natürliche Zahl größer gleich } 0\}$.

Aufgabe 1.6

Zu (2): Es sei p die Ausssage „Der Hahn kräht auf dem Mist.“, und q sei die Aussage „Das Wetter ändert sich.“. Dann wird die umgangssprachliche Aussage durch die aussagenlogische Formel $p \rightarrow (q \vee \neg q)$ repräsentiert. Da $q \rightarrow \neg q$ eine Tautologie ist, also immer den Wert 1 hat, folgt, dass auch $p \rightarrow 1$ immer wahr ist. Somit ist die Aussage auch insgesamt eine Tautologie.

Aufgabe 1.7

a) Die Behauptung folgt unmittelbar aus der Tatsache, dass alle Modelle für $\mathcal{F} \cup \{\beta\}$ auch Modelle von $\mathcal{F}$ sind. Da wegen der Voraussetzung $\mathcal{F} \models \alpha$ alle Modelle von $\mathcal{F}$ auch Modelle von α sind, sind damit die Modelle von $\mathcal{F} \cup \{\beta\}$ auch Modelle von α, und das ist zu zeigen.

b) Falls $\beta \notin \mathcal{F}$ ist, dann ist nichts zu zeigen. Sei also $\beta \in \mathcal{F}$. Da β allgemeingültig ist, sind alle Modelle von $\mathcal{F}$ auch Modelle von $\mathcal{F} - \{\beta\}$ (die Formel

K.-U. Witt, *Mathematische Grundlagen für die Informatik*,
DOI 10.1007/978-3-658-03079-7_5, © Springer Fachmedien Wiesbaden 2013

β ist unerheblich für die Erfüllbarkeit von $\mathcal{F}$), woraus unmittelbar die Behauptung folgt.

Aufgabe 1.9

Aus der Voraussetzung $\alpha \equiv \beta$ folgt $\mathcal{I}^*(\alpha) = \mathcal{I}^*(\beta)$ für alle Belegungen $\mathcal{I}$ von α und β. Aus der Definition der Bijunktion folgt, dass dann auch $\mathcal{I}^*(\alpha \leftrightarrow \beta) = 1$ für alle Belegungen gilt, woraus folgt, dass $\alpha \leftrightarrow \beta$ eine Tautologie ist, also $\alpha \Leftrightarrow \beta$ gilt.

Aus der Voraussetzung $\alpha \Leftrightarrow \beta$ folgt, dass $\alpha \leftrightarrow \beta$ eine Tautologie ist, d.h. $\mathcal{I}^*(\alpha \leftrightarrow \beta) = 1$ für alle Belegungen $\mathcal{I}$ gilt, woraus wegen der Definition der Bijunktion folgt, dass $\mathcal{I}^*(\alpha) = \mathcal{I}^*(\beta)$ für alle Belegungen $\mathcal{I}$ und damit $\alpha \equiv \beta$ ist.

Aufgabe 1.11

(1) Mithilfe der Idempotenz von $\wedge$ und der Definition von $\uparrow$ ergibt sich:

$$\neg\alpha \equiv \neg(\alpha \wedge \alpha) \equiv \alpha \uparrow \alpha$$

(2) Mithilfe doppelter Negation, Definition von $\uparrow$ und a) ergibt sich:

$$\alpha \wedge \beta \equiv \neg\neg(\alpha \wedge \beta) =\equiv \neg(\alpha \uparrow \beta) \equiv (\alpha \uparrow \beta) \uparrow (\alpha \uparrow \beta)$$

(3) Mithilfe doppelter Negation, Definition von $\downarrow$ und (1.11) ergibt sich:

$$\alpha \vee \beta \equiv \neg\neg(\alpha \vee \beta) \equiv \neg(\alpha \downarrow \beta) \equiv (\alpha \downarrow \beta) \downarrow (\alpha \downarrow \beta)$$

(4) Mithilfe doppelter Negation, de Morgan-Regel und a) ergibt sich:

$$\alpha \vee \beta \equiv \neg\neg(\alpha \vee \beta) \equiv \neg(\neg\alpha \wedge \neg\beta) \equiv \neg\alpha \uparrow \neg\beta \equiv (\alpha \uparrow \alpha) \uparrow (\beta \uparrow \beta)$$

Aufgabe 1.12

Wir zeigen, dass die Menge $\{\neg, \vee, \wedge\}$, also die Boolesche Basis, eine aussagenlogische Basis ist. Mithilfe der de Morgan-Regeln folgt dann zum einen

$$\alpha \wedge \beta \equiv \neg\neg(\alpha \wedge \beta) \equiv \neg(\neg\alpha \vee \neg\beta)$$

und zum anderen

$$\alpha \vee \beta \equiv \neg\neg(\alpha \vee \beta) \equiv \neg(\neg\alpha \wedge \neg\beta)$$

und damit, dass $\{\neg, \vee\}$ bzw. $\{\neg, \wedge\}$ ebenfalls aussagenlogische Basen sind. Des Weiteren wissen wir, dass $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$ und damit $\alpha \vee \beta \equiv \neg\alpha \rightarrow \beta$ gilt. Es folgt damit unmittelbar, dass auch $\{\neg, \rightarrow\}$ eine aussagenlogische Basis ist. Da wir in vorherigen Beispielen und Übungsaufgaben gezeigt haben, dass die

Verknüpfugnen $\neg, \vee, \wedge$ sowohl durch $\downarrow$ als auch durch $\uparrow$ definierbar sind, folgt, dass sowohl $\{\downarrow\}$ als auch $\{\uparrow\}$ boolesche Basen sind.

Es bleibt zu zeigen, dass $\{\neg, \vee, \wedge\}$, d.h. die Menge der aussagenlogischen Verknüpfungen, mit der wir die Sprache der Aussagenlogik definiert haben, eine aussagenlogische Basis ist. Wir zeigen das „zu Fuß“, d.h. indem wir zeigen, dass jede der 16 Verknüpfungen durch $\{\neg, \vee, \wedge\}$ definierbar ist. Dazu betrachten wir die Wahrheitstabellen für alle 16 Verknüpfungen:

α	β	$*_0$	$*_1$	$*_2$	$*_3$	$*_4$	$*_5$	$*_6$	$*_7$	$*_8$	$*_9$	$*_{10}$	$*_{11}$	$*_{12}$	$*_{13}$	$*_{14}$	$*_{15}$
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Die Verknüpfung $*_0$ entspricht der Konstanten $\underline{0}$, diese ist äquivalent zu $\alpha \wedge \neg\alpha$. $*_1$ ist die *NOR*-Verknüpfung $\downarrow$, und es gilt $\alpha \downarrow \beta \equiv \neg(\alpha \vee \beta)$. $*_2$ ist die Negation von $*_{13}$. $*_3$ ist die Negation von $*_{12}$. $*_4$ ist die Negation der Subjunktion $*_{11}$. $*_5$ ist die Negation von $*_{10}$. $*_6$ ist das exklusive Oder $\oplus$ äquivalent zur Negation der Bijunktion $*_9$. $*_7$ ist die *NAND*-Verknüpfung: $\alpha \uparrow \beta \equiv \neg(\alpha \wedge \beta)$. $*_8$ ist die Konjunktion. $*_9$ ist die Bijunktion, und es gilt $\alpha \leftrightarrow \beta \equiv (\neg\alpha \vee \beta) \wedge (\alpha \vee \neg\beta)$. $*_{10}$ ist die Projektion auf β, äquivalent zu $\beta \vee (\alpha \vee \neg\alpha)$. $*_{11}$ ist die Subjunktion, und wir wissen, dass $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$ gilt. $*_{12}$ ist die Projektion auf α, darstellbar durch $\alpha \vee (\beta \vee \neg\beta)$. $*_{13}$ ist die Subjunktion $\beta \rightarrow \alpha$ definierbar durch $\alpha \vee \neg\beta$. $*_{14}$ ist die Disjunktion $\vee$. $*_{15}$ entspricht der Konstanten $\underline{1}$, welche definierbar ist durch $\underline{1} \equiv \alpha \vee \neg\alpha$.

Aufgabe 1.13

(3) Es gilt wegen (2) sowie mithilfe bekannter Äquivalenzen aus Tabelle **??**

$$\mathit{ifte}(\alpha, 0, 1) \equiv (\alpha \rightarrow 0) \wedge (\neg\alpha \rightarrow 1) \equiv \neg\alpha \wedge 1 \equiv \neg\alpha$$

womit gezeigt ist, dass die Negation mit der Verknüpfung ifte definierbar ist.

Des Weiteren gilt ebenfalls mit (2) und Tabelle **??**

$$\mathit{ifte}(\alpha, \beta, 1) \equiv (\alpha \rightarrow \beta) \wedge (\neg\alpha \rightarrow 1) \equiv (\alpha \rightarrow \beta) \wedge 1 \equiv \alpha \rightarrow \beta$$

womit gezeigt ist, dass die Subjunktion mit der Verknüpfung ifte definierbar ist.

Da $\{\neg, \rightarrow\}$ eine aussagenlogische Basis bildet (Frege-Basis), und deren beide Elemente durch die Verknüpfung ifte definierbar sind, folgt die Behauptung.

Aufgabe 1.17

Es ist $M_\alpha = \{\{p, \neg q\}, \{p, q, \neg r\}, \{p, \neg q, r\}\}$.

Aufgabe 1.18

(1) $M_\alpha = \{\{p, \neg q, r\}, \{q, r\}, \{\neg p, r\}, \{q, \neg r\}, \{\neg r\}\}$

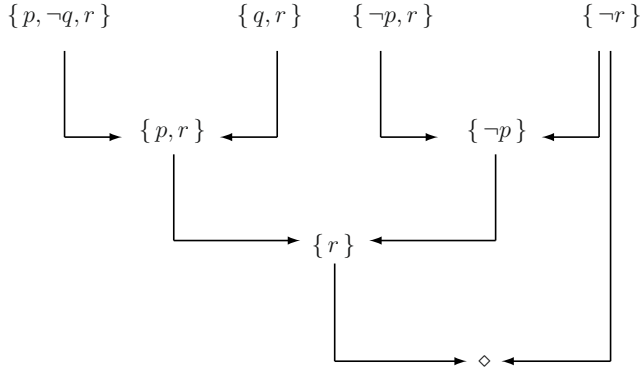
(2) Wir rechnen und lassen triviale Klauseln weg:

$$\begin{aligned}
Res(M_\alpha) = \{\, &\{p, \neg q, r\}, \{q, r\}, \{\neg p, r\}, \{q, \neg r\}, \{\neg r\}, \\
&\{p, r\}, \{\neg q, r\}, \{p, \neg q\}, \{q\}, \{\neg p, q\}, \{\neg p\} \,\} \\
Res^2(M_\alpha) = \{\, &\{p, \neg q, r\}, \{q, r\}, \{\neg p, r\}, \{q, \neg r\}, \{\neg r\}, \\
&\{p, r\}, \{\neg q, r\}, \{p, \neg q\}, \{q\}, \{\neg p, q\}, \{\neg p\} \\
&\{r\} \,\} \\
Res^3(M_\alpha) = \{\, &\{p, \neg q, r\}, \{q, r\}, \{\neg p, r\}, \{q, \neg r\}, \{\neg r\}, \\
&\{p, r\}, \{\neg q, r\}, \{p, \neg q\}, \{q\}, \{\neg p, q\}, \{\neg p\} \\
&\{r\}, \diamond \,\}
\end{aligned}$$

Es gilt $Res^4(M_\alpha) = Res^3(M_\alpha)$ und damit $Res^*(M_\alpha) = Res^3(M_\alpha)$.

(3) Da $\diamond \in Res^*(M_\alpha)$ ist, folgt, dass α unerfüllbar ist.

(4) Folgender Resolutionsgraph deduziert die leere Klausel:



Aufgabe 1.19

Zu (2):

a) $P \rightarrow (\neg W \rightarrow S)$

b) $P \rightarrow \neg W$

c) $P \rightarrow S$

Zu (3):

a) $\neg P \vee W \vee S$

b) $\neg P \vee \neg W$

c) $\neg P \vee S$

Zu (4): Die Formelmenge der Aussagen a) und b) ist gegeben durch:

$$\mathcal{F} = \{\, \neg P \vee W \vee S, \neg P \vee \neg W \,\}$$

Präsident I vermutet

$$\mathcal{F} \models \neg P \vee S$$

Diese Implikation trifft zu, falls die Formelmenge

$$\mathcal{F}' = \{\, \neg P \vee W \vee S, \neg P \vee \neg W, \neg(\neg P \vee S) \,\}$$

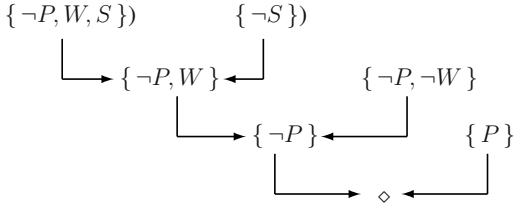
d.h.

$$\mathcal{F}' = \{\, \neg P \vee W \vee S, \neg P \vee \neg W, P \wedge \neg S) \,\}$$

unerfüllbar ist. Die zugehörige Klauselmenge ist

$$\mathcal{M}_{\mathcal{F}'} = \{\, \{\, \neg P, W, S \,\}, \{\, \neg P, \neg W \,\}, \{\, P \,\}, \{\, \neg S \,\} \,\}$$

Zu (5): Folgender Resolutionsgraph zeigt, dass $\mathcal{M}_{\mathcal{F}'}$ unerfüllbar ist. Präsident I hat also Recht mit seiner Vermutung.



Aufgabe 1.20

Zu (1): Umwandlung der Klauseln in Subjunktionen:

$$\begin{aligned} p \wedge q &\rightarrow r \\ s &\rightarrow t \\ p \wedge t &\rightarrow 0 \\ 1 &\rightarrow p \\ p &\rightarrow q \end{aligned} \tag{5.1}$$

Wegen der Klausel (5.1) erhalten wir mit Verfahrensschritt (1) folgende roten Markierungen:

$$\begin{aligned} p \wedge q &\rightarrow r \\ s &\rightarrow t \\ p \wedge t &\rightarrow 0 \\ 1 &\rightarrow p \\ p &\rightarrow q \end{aligned} \tag{5.2}$$

Anwendung von Verfahrensschritt (2i) führt wegen der Klausel (5.2) zur folgenden blauen Markierung:

$$\begin{aligned} p \wedge q &\rightarrow r \\ s &\rightarrow t \\ p \wedge t &\rightarrow 0 \\ 1 &\rightarrow p \\ p &\rightarrow q \end{aligned} \tag{5.3}$$

Anwendung von Verfahrensschritt (2i) führt wegen der Klausel (5.3) zur folgenden grünen Markierung:

$$\begin{aligned} p \wedge q &\rightarrow r \\ s &\rightarrow t \\ p \wedge t &\rightarrow 0 \\ 1 &\rightarrow p \\ p &\rightarrow q \end{aligned}$$

Es ist kein Verfahrensschritt mehr anwendbar, die Formel ist erfüllbar, und die Belegung $\mathcal{I}(p) = \mathcal{I}(q) = \mathcal{I}(r) = 1$ sowie $\mathcal{I}(s) = \mathcal{I}(t) = 0$ ist ein Modell für α.

Zu (2): Umwandlung der Klauseln in Subjunktionen:

$$\begin{aligned} 1 &\rightarrow p \\ p &\rightarrow q \\ q \wedge s &\rightarrow r \\ p \wedge r &\rightarrow 0 \\ 1 &\rightarrow s \end{aligned} \tag{5.4}$$

$$\tag{5.5}$$

Wegen der Klauseln (5.4) und (5.5) erhalten wir mit Verfahrensschritt (1) folgende roten Markierungen:

$$\begin{aligned} 1 &\rightarrow p \\ p &\rightarrow q \\ q \wedge s &\rightarrow r \\ p \wedge r &\rightarrow 0 \\ 1 &\rightarrow s \end{aligned} \tag{5.6}$$

Anwendung von Verfahrensschritt (2i) führt wegen der Klausel (5.6) zur folgenden blauen Markierung:

$$\begin{aligned} 1 &\to p \\ p &\to q \\ q \wedge s &\to r \\ p \wedge r &\to 0 \\ 1 &\to s \end{aligned} \tag{5.7}$$

Anwendung von Verfahrensschritt (2i) führt wegen der Klausel (5.7) zur folgenden grünen Markierung:

$$\begin{aligned} 1 &\to p \\ p &\to q \\ q \wedge s &\to r \\ p \wedge r &\to 0 \\ 1 &\to s \end{aligned} \tag{5.8}$$

Wegen Klausel (5.8) besagt Verfahrensschritt (2ii), dass die Formel β unerfüllbar ist.

Aufgabe 1.21

Direkter Beweis: Es gilt $\left(\sqrt{a} - \sqrt{b}\right)^2 \geq 0$. Daraus folgt $a - 2\sqrt{a \cdot b} + b \geq 0$ und daraus die Behauptung.

Aufgabe 1.22

(1) Wir nehmen an, dass

$$\frac{a+b}{2} < \sqrt{a \cdot b}$$

gilt. Dann folgt $a - 2\sqrt{a \cdot b} + b < 0$ und daraus $\left(\sqrt{a} - \sqrt{b}\right)^2 < 0$, was ein Widerspruch gegen die Tatsache ist, das Quadratzahlen nicht negativ sind.

(2) Direkter Beweis: Da $x, y \in \mathbb{U}_+$ ist, gibt es $m, n \in \mathbb{N}_0$ mit $x = 2m + 1$ und $y = 2n + 1$. Es gilt $x \cdot y = 4mn + 2m + 2n + 1 = 2(mn + m + n) + 1 \in \mathbb{U}_+$.

Widerspruchsbeweis: Da $x, y \in \mathbb{U}_+$ ist, gibt es $m, n \in \mathbb{N}_0$ mit $x = 2m + 1$ und $y = 2n + 1$. Wir nehmen an, dass $xy \in \mathbb{G}_+$ ist. Dann muss $(2m + 1)(2n + 1) \in \mathbb{G}_+$, also $2(mn + m + n) + 1 \in \mathbb{G}_+$ sein, was einen Widerspruch dazu bedeutet, dass sich alle $z \in \mathbb{G}_+$ als $z = 2k$ für ein geeignetes $k \in \mathbb{N}$ darstellen lassen.

Aufgabe 1.23

(1) Nach Definition 1.21 ist zu zeigen: $x \in A \Rightarrow x \in A$, d.h. wir müssen zeigen, dass die Subjunktion $x \in A \to x \in A$ immer wahr ist.

Das Prädikat $x \in A$ kann wahr oder falsch sein. Folgende Wahrheitstafel zeigt die beiden möglichen Belegungen für die Subjunktion:

$x \in A$	$x \in A$	$x \in A \to x \in A$
1	1	1
0	0	1

$x \in A \to x \in A$ ist also immer wahr (es handelt sich um eine Tautologie der Art $\alpha \to \alpha$), d.h. es gilt $x \in A \Rightarrow x \in A$ und damit die Behauptung $A \subseteq A$.

Aufgabe 1.24

(1) Es ist $\mathcal{P}(\emptyset) = \{\emptyset\}$ sowie $\mathcal{P}(\mathcal{P}(\emptyset)) = \mathcal{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$ (beides folgt z.B. umittelbar aus Folgerung 1.14).

(2) Die einzige Menge mit dieser Eigenschaft ist die leere Menge $M = \emptyset$. In (1) sieht man, dass $\emptyset \subseteq \mathcal{P}(\emptyset)$ gilt (es gilt ja $\emptyset \subseteq A$ für jede Menge A, siehe Folgerung 1.13 a). Ist $M \neq \emptyset$, also z.B. $a \in M$, und würde $M \subseteq \mathcal{P}(M)$ sein, dann wäre $a \in \mathcal{P}(M)$. Die Elemente von $\mathcal{P}(M)$ enthalten aber – außer der leeren Menge – nur Mengen, die die Elemente von M enthalten, aber nicht die Elemente als solche.

Aufgabe 1.26

Ein Beispiel ist die Aussagenlogik. Weitere Beispiele sind die Potenzmengenalgebren einelementiger Mengen, wie z.B. $\mathcal{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$.

Aufgabe 2.2

(1) Es sei $p(x, y)$ das Prädikat xRy, $q(x, y)$ das Prädikat yRx und $r(x, y)$ das Prädikat $x = y$. Die Definition der Antisymmetrie wird repräsentiert durch (wir lassen aus schreibtechnischen Gründen die Variablen weg):

$$p \wedge q \to r$$

Die Aussage in der Aufgabe wird repräsentiert durch

$$p \wedge \neg r \to \neg q$$

Wir zeigen, dass diese beiden Ausdrücke äquivalent sind:

$$p \wedge q \to r \equiv \neg(p \wedge q) \vee r \equiv \neg p \vee r \vee \neg q \equiv \neg(p \wedge \neg r) \vee \neg q \equiv p \wedge \neg r \to \neg q$$

(2) Die logische Darstellung der Definition der Injektivität und die logische Darstellung der Aufgabe haben die gleiche Struktur wie die Aussagen in Aufgabe (1): Wenn wir p für $x_1 R y_1 \wedge x_2 R y_2$, q für $x_1 \neq x_2$ und r für $y_1 \neq y_2$ setzen, dann ist die Definiton der Injektivität gegeben durch

$$p \wedge q \to r$$

und die Behauptung durch

$$p \wedge \neg r \rightarrow \neg q$$

Die Äquivalenz dieser beiden Ausdrücke haben wir bereits in (1) gezeigt.

(3) lässt sich vollkommen analog zu (2) beweisen – führen Sie bitte den Beweis selber!

(4), (5) Diese Behauptungen gelten offensichtlich.

Aufgabe 2.6

a), c) und d) folgen unmittelbar aus Satz 2.1.

b) Sei I eine Indexmenge und $\{A_i\}_{i \in I}$ eine Partition von A. Wir definieren $R \subseteq A \times A$ durch: xRy gilt genau dann, wenn $x, y \in A_i$ für ein $i \in I$ ist, d.h. zwei Elemente $x, y \in A$ gehören zur Relation R genau dann, wenn sie zur selben Klasse A_i von A gehören. Es ist offensichtlich, dass R reflexiv und symmetrisch ist. Gilt xRy und yRz, dann müssen x, y, z zur selben Klasse von A gehören, dann gilt aber auch xRz. R ist also auch transitiv. R ist also eine Äquivalenzrelation auf A, womit die Behauptung gezeigt ist.

Aufgabe 2.7

(1) Der Beweis erfolgt analog zum Beweis in Beispiel 2.8 b), wir müssen nur 3 durch m ersetzen. Die ÄquivalenzKlassen (Restklassen) sind

$$[\,k\,]_m = \{q \cdot m + k \mid q \in \mathbb{Z}\}$$

für $0 \leq k \leq m-1$. Der Index von $\equiv_m$ ist m.

(2) Wenn m ein Teiler von n ist, dann gibt es eine Zahl $t \in \mathbb{Z}$ mit $n = t \cdot m$. Wir müssen zeigen, dass es zu k mit $0 \leq k \leq n-1$ ein k' mit $0 \leq k' \leq m-1$ gibt, so dass $[\,k\,]_n \subseteq [\,k'\,]_m$ gilt. Das geeignete k' erhalten wir, wenn wir k durch m teilen:

$$k = q_k \cdot m + k' \text{ mit } 0 \leq k' \leq m-1$$

$q_k \in \mathbb{Z}$ ist der Quotient und r ist der kleinste positive Rest, die bei Division von k durch m entstehen. Es gilt nun

$$\begin{aligned}
x \in [\,k\,]_n \text{ genau dann, wenn } & x = q \cdot n + k \text{ für ein } q \in \mathbb{Z} \\
\text{genau dann, wenn } & x = q \cdot n + q_k \cdot m + k' \text{ für } q, q_k \in \mathbb{Z} \\
\text{genau dann, wenn } & x = q \cdot t \cdot m + q_k \cdot m + k' \text{ für } q, q_k, t \in \mathbb{Z} \\
\text{genau dann, wenn } & x = (q \cdot t + q_k) \cdot m + k' \text{ für } q, q_k, t \in \mathbb{Z} \\
\text{genau dann, wenn } & x = q' \cdot m + k' \text{ für } q' = q + q_k + t \in \mathbb{Z} \\
\text{genau dann, wenn } & x \in [\,k'\,]_m
\end{aligned}$$

womit wir wie angestrebt $[\,k\,]_n \subseteq [\,k'\,]_m$ gezeigt haben.

Aufgabe 2.10

a) folgt unmittelbar aus der Definition 2.10.

b) folgt unmittelbar aus c).

c) folgt unmittelbar aus der Symmetrie von R.

d) folgt unmittelbar aus e).

e) Die Behauptung folgt wegen der Symmetrie von R: Es gilt

$$xRy \text{ gdw. } yRx \text{ gedw. } xR^{-1}y$$

Aufgabe 3.1

Wir nehmen an, dass $\varphi(0_1) = y \neq 0_2$ ist. Da $y \neq 0_2$ ist, hat y einen Vorgänger $z \in \mathcal{S}_2$: $s_2(z) = y$. Sei $x \in \mathcal{S}_1$ das Urbild von z: $\varphi(x) = z$. Wir erhalten mit diesen Gleichungen und der Strukturgleichung (3.1)

$$\varphi(0_1) = y = s_2(z) = s_2(\varphi(x)) = \varphi(s_1(x))$$

und damit gilt $0_1 = s_1(x)$, da φ injektiv ist. Damit wäre 0_1 Nachfolger von x, was ein Widerspruch zum Axiom P1 darstellt. Damit ist unsere Annahme widerlegt und die Behauptung gezeigt.

Aufgabe 3.3

Die Gültigkeit der Induktionsanfänge ist bei allen Aufgaben leicht zu verifizieren, deshalb zeigen wir nur die Induktionsschritte.

(1) $2(n+1)+1 = 2n+1+2 \leq 2^{n-1}+2 \leq 2^{n-1}+2^{n-1} = 2 \cdot 2^{n-1} = 2^n$

(2) $\sum_{i=1}^{n+1} 2(i-1) = \sum_{i=1}^{n} 2(i-1) + 2((n+1)-1) = n^2+2n+1 = (n+1)^2$

(3) $\sum_{i=0}^{n+1} q^i = \sum_{i=0}^{n} q^i + q^{n+1} = \frac{1-q^{n+1}}{1-q} + q^{n+1} = \frac{1-q^{n+1}+q^{n+1}-q^{n+2}}{1-q} = \frac{1-q^{n+2}}{1-q}$

(4) $\frac{x^{n+1}-y^{n+1}}{x-y} = \frac{x(x^n-y^n)+y^n(x-y)}{x-y} = x \cdot \frac{x^n-y^n}{x-y} + y^n \in \mathbb{Z}$

(5) $\sum_{i=1}^{n+1} \frac{1}{i(i+1)} = \sum_{i=1}^{n} \frac{1}{i(i+1)} + \frac{1}{(n+1)(n+2)} = \frac{n}{n+1} + \frac{1}{(n+1)(n+2)} = \frac{n+1}{n+2}$

(6) Der Induktionsschritt reduziert sich darauf zu zeigen, dass $(-1)^{2n+1}(2n+1)^2 + (-1)^{2n+2}(2n+2)^2 = (2n+1)+(2n+2)$ ist, was durch Nachrechnen erledigt werden kann: Beide Seiten ergeben $4n+3$.

(7) Zeigen Sie zuerst, dass $\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$ gilt. Dann folgt die Behauptung mit der im Beispiel 3.3 a) (Seite 125) bewiesenen Gleichheit $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

(8) $\prod_{i=0}^{n} A_i = A_n \cdot \prod_{i=0}^{n-1} A_i = A_n(A_n-2) = (2^{2^n}+1)(2^{2^n}+1-2) = (2^{2^n}+1)(2^{2^n}-1) = 2^{2^n} \cdot 2^{2^n} - 1 = 2^{2^n+2^n} - 1 = 2^{2 \cdot 2^n} - 1 = 2^{2^{n+1}} - 1 = 2^{2^{n+1}} + 1 - 2 = A_{n+1} - 2$

(9) Diese Behauptung stiummt für $n \in \mathbb{N}_0$ mit $0 \leq n \leq 41$, aber nicht für $n = 41$: $41^2 - 41 + 41 = 41^2 \notin \mathbb{P}$.

Aufgabe 3.6

(iv) $F_{n+2}F_n - F_{n+1} = (F_{n+1} + F_n)F_n - (F_n + F_{n-1})^2 = F_{n+1}F_n + F_n^2 - F_n^2 - 2F_nF_{n-1} - F_{n-1}^2 = F_{n+1}F_n - F_nF_{n-1} - F_nF_{n-1} - F_{n-1}^2 = F_n(F_{n+1} - F_{n-1}) - F_{n-1}(F_n + F_{n-1}) = F_n^2 - F_{n-1}F_{n+1} = (-1)(F_{n+1}F_{n-1} - F_n^2) = (-1)\cdot(-1)^n = (-1)^{n+1}$

Aufgabe 3.9

Wir nehmen an, dass $\mathcal{P}(\mathbb{N})$ abzählbar ist, $M_1, M_2, M_3, \ldots$ sei eine solche Abzählung, und $x_1, x_2, s_3, \ldots$ sei eine Abzählung von $\mathbb{N}$. Damit stellen wir folgende (boolesche) Matrix auf:

	x_1	x_2	x_3	$\ldots$	x_j	$\ldots$
M_1	b_{11}	b_{12}	b_{13}	$\ldots$	b_{1j}	$\ldots$
M_2	b_{21}	b_{22}	b_{23}	$\ldots$	b_{2j}	$\ldots$
M_3	b_{31}	b_{32}	b_{33}	$\ldots$	b_{3j}	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
M_i	b_{i1}	b_{i2}	b_{i3}	$\ldots$	b_{ij}	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Dabei ist

$$b_{ij} = \begin{cases} 1, & x_j \in M_i \\ 0, & x_i \notin M_i \end{cases}$$

Mithilfe der Diagonalen der Matrix definieren die Menge M_D durch:

$$x_k \in M_D \text{ genau dann, wenn } x_k \notin M_k \text{ (d.h. genau dann, wenn } b_{kk} = 0) \quad (5.9)$$

M_D ist eine Menge natürlicher Zahlen, also ein Element von $\mathcal{P}(\mathbb{N})$. Diese Menge muss also in der Abzählung $M_1, M_2, M_3, \ldots$ vorkommen, d.h. es muss eine Nummer r geben, mit $M_D = M_r$. Für das Element x_r gibt es zwei Fälle: (1) $x_r \in M_D$ oder (2) $x_r \notin M_D$. Für diese gilt:

(1): Aus $x_r \in M_D$ folgt $x_r \in M_r$ wegen $M_D = M_r$ und daraus wegen (5.9) $x_r \notin M_D$

(2): Aus $x_r \notin M_D$ folgt $x_r \notin M_r$ wegen $M_D = M_r$ und daraus wegen (5.9) $x_r \in M_D$

Beide Fälle führen also zu einem Widerspruch, womit unsere Annahme widerlegt und die Behauptung gezeigt ist.

Aufgabe 3.11

(1) Durchschnitt und Vereinigung sind kommutative und assoziative Verjnüpfungen, also sind beide Strukturen kommutative Monoide. In der Struktur (i) ist die leere Menge das Einselement, denn es gilt $A \cup \emptyset = \emptyset \cup A = A$ für alle $A \in \mathcal{P}(M)$. In der Struktur (ii) ist die Menge M das Einselement, denn es gilt $A \cap M = M \cap A = A$ für alle $A \in \mathcal{P}(M)$. Abgesehen von dem Fall $M = \emptyset$

```
read(x);
 ω := 0;
 while  x + ω + 1 ≠ 0  do
     ω := ω + 1
 endwhile;
write(ω)
```

Abb. 18: Berechnung der nirgends definierten Funktion ω durch μ-Rekursion

bilden beide Strukturen keine Gruppen, denn in der Struktur (i) gibt es zu keiner Menge $A \in \mathcal{P}(M)$ mit $A \neq \emptyset$ eine Menge $B \in \mathcal{P}(M)$ mit $A \cup B = \emptyset$, und in der Struktur (ii) gebt es zu keiner Menge $A \in \mathcal{P}(M)$ mit $A \neq M$ eine Menge $B \in \mathcal{P}(M)$ mit $A \cap B = M$.

(2) Es sei $\Pi(M) = \{f : M \to M \mid f \text{ bijektiv}\}$ die Menge der bijektiven Funktionen von der Menge M in die Menge M; damit betrachten wir die Rechenstruktur $(\Pi(M), \circ)$. Die Struktur ist ein Monoid, denn Kompositon von Funktionen ist assoziativ (siehe Satz 2.4 (Seite 102. Das Einselement ist die identische Abbildung id_M (siehe Satz 2.2 a) und b) auf Seite 102). Das Inverse zur Bijektion f ist die Umkehrfunktion f^{-1}. Aus Übung 2.11 (2) und wegen Beispiel 2.16 auf Seite 107 wissen wir, dass die Komposition von Funktionen im Allgemeinen nicht kommutativ ist.

Aufgabe 4.3

(1) Idee: Die Eingabe x wird um1 erhöht, damit $x + 1$ auch für $x = 0$ größer als 0 ist. $x + 1$ wird dann permanent um 1 erhöht, womit der Wert niemals gleich 0 wird, d.h. die Schleife terminiert für keine Eingabe x (siehe Abbildung 18). Entsprechend unserem Schema ergibt sich als μ-rekursive Funktion ω:

$$\omega = \mu\left[\mathcal{C}\left[\nu; \mathcal{C}\left[add; \pi^2_1, \pi^2_2\right]\right]\right]$$

(2) Falls $y = 0$ ist, ist div nicht definiert, für $x = y$ ist das Ergebnis 1, sonst wird q beginnend bei 0 hochgezählt, bis $g(x, y, q) = x - y(q + 1) = 0$ ist:

$$div(x, y) = equal(y, 0) \cdot \omega(x) + equal(x, y) + \mu\left[g(x, y, d)\right]$$

(3) Der Rest ergibt sich durch

$$mod(x, y) = x - y \cdot div(x, y)$$

(4) Wir passen Beispiel 4.2 b) (Seite 174) im Exponenten an:

$$\sqrt[y]{} = \mu\left[\mathcal{C}\left[sub; \pi^2_1, \mathcal{C}\left[exp; \pi^3_3, \pi^3_2\right]\right]\right]$$

```
read(x, n);
 y := 0;
 while  x ⊖ exp(y, n) ≠ 0  do
     y := y + 1
 endwhile;
write(y)
```

Abb. 19: Berechnung der Funktion $\sqrt[n]{\ }$ durch μ-Rekursion

Abbildung 19 zeigt eine Implementierung dieses Verfahrens.

(5), (6) Wir überlegen uns zunächst diese Funktionen in Pseudocode:

$$min(x,y) = sign(x-y)\cdot y + sign(y-x)\cdot x + equal(x,y)\cdot x$$
$$max(x,y) = sign(x-y)\cdot x + sign(y-x)\cdot y + equal(x,y)\cdot x$$

sowie zur kürzeren und übersichtlicheren Darstellung die Hilfsfunktionen

$$s(a,b,c) = sign(a-b)\cdot c$$
$$s = \mathcal{C}\left[mult; \mathcal{C}\left[sign; \mathcal{C}\left[minus; \pi^3_1, \pi^3_2\right], \pi^3_3\right]\right]$$
$$eq(d,e,f) = equal(d,e)\cdot f$$
$$eq = \mathcal{C}\left[mult; \mathcal{C}\left[equal; \pi^3_1, \pi^3_2\right], \pi^3_3\right]$$

Damit erhalten wir

$$min(x,y) = s(x,y,y) + s(y,x,x) + eq(x,y,x)$$

$$min = \mathcal{C}\left[add; \mathcal{C}\left[add; \mathcal{C}\left[s; \pi^2_1, \pi^2_2, \pi^2_2\right], \mathcal{C}\left[s; \pi^2_2, \pi^2_1, \pi^2_1\right]\right], \mathcal{C}\left[eq; \pi^2_1, \pi^2_2, \pi^2_1\right]\right]$$

$$max(x,y) = s(x,y,x) + s(y,x,y) + eq(x,y,x)$$

$$max = \mathcal{C}\left[add; \mathcal{C}\left[add; \mathcal{C}\left[s; \pi^2_1, \pi^2_2, \pi^2_1\right], \mathcal{C}\left[s; \pi^2_2, \pi^2_1, \pi^2_2\right]\right], \mathcal{C}\left[eq; \pi^2_1, \pi^2_2, \pi^2_1\right]\right]$$

(7) Rechnen Sie nach, dass

$$geq(x,y) = max(sign(x-y), equal(x,y))$$

gilt!

(8) $binom(n,k) = div(fak(n), mult(fak(k), fak(n-k)))$

Literatur

Appell, K., Appell, J.: *Mengen – Zahlen – Zahlbereiche, Eine elementare Einführung in die Mathematik;* Spektrum Akademischer Verlag, München, 2005

Ben-Ari,M.: *Mathematical Logic for Computer Science, Third Edition;* Springer, London, 2012

Dassow, J.: *Logik für Informatiker;* Teubner, Wiesbaden, 2005

Haggarty, R.: *Diskrete Mathematik für Informatiker;* Pearson Studium, München, 2004

Hoffmann, D.: *Grenzen der Mathematik, Eine Reise durch die Kerngebiete der mathematischen Logik;* Spektrum Akademischer Verlag, Heidelberg, 2011

Kelly, J.: *Logik im Klartext;* Pearson Studium, München, 2003

Kramer, J., von Pippich, A.-M.: *Von den natürlichen Zahlen zu den Quaternionen;* Springer Spektrum, Wiesbaden, 2013

Meinel, C., Mundhenk, M.: *Mathematische Grundlagen der Informatik;* Teubner, Stuttgart, 2002

Rogers, H.: *Theory of Recursive Functions and Effective Computability;* The MIT Press, Cambridge, MA, 1987

Schubert, M.: *Mathematik für Informatiker;* Vieweg+Teubner, Wiesbaden, 2009

Siefkes, D.: *Formalisieren und Beweisen;* Vieweg, Wiesbaden, 1990

Vossen, G., Witt, K.-U.: *Grundkurs Theoretische Informatik, 5. Auflage;* Vieweg + Teubner, Wiesbaden, 2011

Waismann, E.: *Einführung in das mathematische Denken;* Wissenschaftliche Buchgesellschaft, Darmstadt, 1996

K.-U. Witt, *Mathematische Grundlagen für die Informatik*,
DOI 10.1007/978-3-658-03079-7, © Springer Fachmedien Wiesbaden 2013

Stichwortverzeichnis