

4 WÖRTER

4.1 WÖRTER

Jeder weiß, was ein Wort ist: Ein *Wort über einem Alphabet A* ist eine Folge von Zeichen aus A . Aber gerade weil jeder weiß, was das ist, werden wir uns im folgenden eine Möglichkeit ansehen, eine formale Definition des Begriffes „Wort“ zu geben. Sinn der Übung ist aber nicht, eine einfache Sache möglichst kompliziert darzustellen, sondern an einem Beispiel, das niemandem ein Problem bereitet, Dinge zu üben, die in späteren Kapiteln noch wichtig werden.

Wort über einem Alphabet A

Vorher aber noch kurz eine Bemerkung zu einem Punkt, an dem sich der Sprachgebrauch in dieser Vorlesung (und allgemeiner in der Theorie der formalen Sprachen) vom umgangssprachlichen unterscheidet: Das *Leerzeichen*. Übrigens benutzt man es heutzutage (jedenfalls z. B. in europäischen Schriften) zwar ständig — früher aber nicht! Aber wie der Name sagt, fassen wir auch das Leerzeichen als ein Zeichen auf. Damit man es sieht, schreiben wir manchmal explizit \square statt einfach nur Platz zu lassen.

Konsequenz der Tatsache, dass wir das Leerzeichen, wenn wir es denn überhaupt benutzen, als ein ganz normales Symbol auffassen, ist jedenfalls, dass z. B. *Hallo \square Welt* eine Folge von Zeichen, also nur *ein* Wort ist (und nicht zwei).

Was man für eine mögliche technische Definition von „Wort“ braucht, ist im wesentlichen eine Formalisierung von „Liste“ (von Symbolen). Für eine bequeme Notation definieren wir zunächst: Für jede natürliche Zahl $n \geq 0$ sei $\mathbb{Z}_n = \{i \mid i \in \mathbb{N}_0 \text{ und } 0 \leq i \text{ und } i < n\}$ die Menge der n kleinsten nichtnegativen ganzen Zahlen. Zum Beispiel ist $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, $\mathbb{Z}_1 = \{0\}$ und $\mathbb{Z}_0 = \{\}$.

Dann wollen wir jede *surjektive* Abbildung $w : \mathbb{Z}_n \rightarrow A$ als ein *Wort* auffassen. Die Zahl n heiße auch die *Länge* des Wortes, für die man manchmal kurz $|w|$ schreibt. (Es ist in Ordnung, wenn Sie im Moment nur an Längen $n \geq 1$ denken. Auf den Fall des sogenannten leeren Wortes ε mit Länge $n = 0$ kommen wir im *nachfolgenden Abschnitt* gleich noch zu sprechen.)

Wort, formalistisch definiert
Länge eines Wortes

Das Wort (im umgangssprachlichen Sinne) $w = \text{hallo}$ ist dann also formal die Abbildung $w : \mathbb{Z}_5 \rightarrow \{a, h, l, o\}$ mit $w(0) = h$, $w(1) = a$, $w(2) = l$, $w(3) = l$ und $w(4) = o$.

Im folgenden werden wir uns erlauben, manchmal diese formalistische Sicht auf Wörter zu haben, und manchmal die vertraute von Zeichenfolgen. Dann ist insbesondere jedes einzelne Zeichen auch schon ein Wort. Formalismus, vertraute Sichtweise und das Hin- und Herwechseln zwischen beidem ermöglichen dreierlei:

- präzise Argumentationen, wenn andernfalls nur vages Händewedeln möglich wäre,

- leichteres Vertrautwerden mit Begriffen und Vorgehensweisen bei Wörtern und formalen Sprachen, und
- das langsame Vertrautwerden mit Formalismen.

Menge aller Wörter
 A^*

Ganz häufig ist man in der Situation, dass man ein Alphabet A gegeben hat und über die *Menge aller Wörter* reden möchte, in denen höchstens die Zeichen aus A vorkommen. Dafür schreibt man A^* . Ganz formalistisch gesehen ist das also Menge aller surjektiven Abbildungen $w : \mathbb{Z}_n \rightarrow B$ mit $n \in \mathbb{N}_0$ und $B \subseteq A$. Es ist aber völlig in Ordnung, wenn Sie sich einfach Zeichenketten vorstellen.

4.2 DAS LEERE WORT

Beim Zählen ist es erst einmal natürlich, dass man mit eins beginnt: 1, 2, 3, 4, ... Bei Kindern ist das so, und geschichtlich gesehen war es bei Erwachsenen lange Zeit auch so. Irgendwann stellte sich jedoch die Erkenntnis ein, dass die Null auch ganz praktisch ist. Daran hat sich jeder gewöhnt, wobei vermutlich eine gewisse Abstraktion hilfreich war; oder stellen Sie sich gerade vor, dass vor Ihnen auf dem Tisch 0 Elefanten stehen?

Ebenso umfasst unsere eben getroffene Definition von „Wort“ den Spezialfall der Wortlänge $n = 0$. Auch ein Wort der Länge 0 verlangt zugegebenermaßen ein bisschen Abstraktionsvermögen. Es besteht aus 0 Symbolen. Deshalb sieht man es so schlecht.

Wenn es Ihnen hilft, können Sie sich die formalistische Definition ansehen: Es ist $\mathbb{Z}_0 = \{\}$ die leere Menge; und ein Wort der Länge 0 enthält keine Zeichen. Formalisiert als surjektive Abbildung ergibt das dann ein $w : \{\} \rightarrow \{\}$.

Wichtig:

- Wundern Sie sich nicht, wenn Sie sich über $w : \{\} \rightarrow \{\}$ erst einmal wundern. Sie werden sich an solche Dinge schnell gewöhnen.
- Vielleicht haben Sie ein Problem damit, dass der Definitionsbereich oder/und der Zielbereich von $w : \{\} \rightarrow \{\}$ die leere Menge ist. Das löst sich aber, wenn man daran denkt, dass Abbildungen besondere Relationen sind.
- Es gibt nur *eine* Relation $R \subseteq \{\} \times \{\} = \{\}$, nämlich $R = \{\}$. Als Menge von Paaren aufgefasst ist dieses R aber linkstotal und rechtseindeutig, also tatsächlich eine Abbildung; und die ist sogar rechtstotal. Also ist es richtig von *dem leeren Wort* zu sprechen.

leeres Wort

Nun gibt es ähnlich wie schon beim Leerzeichen ein ganz praktisches Problem: Da das leere Wort aus 0 Symbolen besteht, „sieht man es nicht“. Das führt leicht zu Verwirrungen. Man will aber gelegentlich ganz explizit darüber sprechen und schreiben. Deswegen vereinbaren wir, dass wir für das leere Wort ε schreiben.

Beachten Sie, dass wir in unseren Beispielen Symbole unseres Alphabetes immer blau darstellen, ε aber nicht. Es ist nie Symbol das gerade untersuchten Alphabetes. Wir benutzen dieses Zeichen aus dem Griechischen als etwas, das Sie immer interpretieren (siehe Kapitel 2) müssen, nämlich als das leere Wort.

4.3 MEHR ZU WÖRTERN

Für die Menge aller Wörter einer festen Länge n über einem Alphabet A schreiben wir auch A^n . Wenn zum Beispiel das zu Grunde liegende Alphabet $A = \{\mathbf{a}, \mathbf{b}\}$ ist, dann ist:

$$\begin{aligned} A^0 &= \{\varepsilon\} \\ A^1 &= \{\mathbf{a}, \mathbf{b}\} \\ A^2 &= \{\mathbf{aa}, \mathbf{ab}, \mathbf{ba}, \mathbf{bb}\} \\ A^3 &= \{\mathbf{aaa}, \mathbf{aab}, \mathbf{aba}, \mathbf{abb}, \mathbf{baa}, \mathbf{bab}, \mathbf{bba}, \mathbf{bbb}\} \end{aligned}$$

Vielleicht haben nun manche die Idee, dass man auch erst die A^n hätte definieren können, und dann festlegen:

$$A^* = A^0 \cup A^1 \cup A^2 \cup A^3 \cup \dots$$

Das unschöne daran sind die „...“: Im vorliegenden Fall mag ja noch klar sein, was gemeint ist. Da wir aber darauf achten wollen, dass Sie sich nichts angewöhnen, was im allgemeinen zu Problemen führen könnte (und dafür sind Pünktchen, bei denen man darauf baut, dass der Leser schon die passende Interpretation haben möge, prädestiniert), wollen wir lieber das „große Vereinigungszeichen“ benutzen, das wir in Kapitel 3 kennengelernt haben:

$$A^* = \bigcup_{i \in \mathbb{N}_0} A^i$$

4.4 KONKATENATION VON WÖRTERN

Zahlen kann man zum Beispiel addieren oder multiplizieren. Man spricht auch davon, dass die Addition und Multiplikation zweistellige oder binäre Operationen sind.

Für Wörter definieren wir nun auch eine ganz einfache aber wichtige binäre Operation: die sogenannte *Konkatenation* von Wörtern. Das ist einfach die Hin-

Konkatenation

tereinanderschreibung zweier Wörter. Als Operationssymbol verwendet man üblicherweise wie bei der Multiplikation von Zahlen den Punkt „·“. Also zum Beispiel:

$$\text{SCHRANK} \cdot \text{SCHLÜSSEL} = \text{SCHRANKSCHLÜSSEL}$$

oder

$$\text{SCHLÜSSEL} \cdot \text{SCHRANK} = \text{SCHLÜSSELSCHRANK}$$

Oft lässt man wie bei der Multiplikation auch den Konkatenationspunkt weg.

Wie man sieht, kommt es (im Gegensatz zur Multiplikation von Zahlen) auf die Reihenfolge an: Ein **SCHRANKSCHLÜSSEL** ist etwas anderes als ein **SCHLÜSSELSCHRANK**.

Konkatenation zweier
Wörter formal

Nun wollen wir die *Konkatenation zweier Wörter formal* definieren.

4.1 Definition. Es seien zwei beliebige Wörter $w_1 : \mathbb{Z}_m \rightarrow A_1$ und $w_2 : \mathbb{Z}_n \rightarrow A_2$ gegeben. Dann ist

$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A_1 \cup A_2$$

$$i \mapsto \begin{cases} w_1(i) & \text{falls } 0 \leq i < m \\ w_2(i-m) & \text{falls } m \leq i < m+n \end{cases}$$

Ist das eine sinnvolle Definition? Oder vielmehr: Ist das überhaupt eine Definition? Und wird hier ein Wort definiert?

- Als erstes hat man sich zu überlegen, dass die Ausdrücke $w_1(i)$ für $0 \leq i < m$ und $w_2(i-m)$ für $m \leq i < m+n$ stets definiert sind. Das ist so.
- Zweitens stammen die in der Fallunterscheidung vorgeschriebenen Funktionswerte tatsächlich aus dem Bereich $A_1 \cup A_2$: denn $w_1(i)$ ist stets aus A_1 und $w_2(i-m)$ ist stets aus A_2 .
- Drittens muss man sich klar machen, dass die Fallunterscheidung von der Art ist, dass für jedes $i \in \mathbb{Z}_{m+n}$ nur genau *ein* Funktionswert festgelegt wird und nicht mehrere verschiedene.
- Und schließlich muss man sich noch klar machen, dass wieder ein *Wort* definiert wird: Dafür muss die Abbildung $w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A_1 \cup A_2$ surjektiv sein. Das ist sie auch. Denn für jedes $a \in A_1 \cup A_2$ gilt (mindestens) eine der folgenden Möglichkeiten:
 - $a \in A_1$: Dann gibt es aber, da w_1 ein Wort ist, also eine surjektive Abbildung, ein $i_1 \in \mathbb{Z}_m$ mit $w_1(i_1) = a$. Also ist $(w_1 w_2)(i_1) = w_1(i_1) = a$.
 - $a \in A_2$: Dann gibt es aber, da w_2 ein Wort ist, also eine surjektive Abbildung, ein $i_2 \in \mathbb{Z}_n$ mit $w_2(i_2) = a$. Also ist $(w_1 w_2)(m + i_2) = w_2(i_2) = a$.

Als letztes sei noch angemerkt, dass man an der Definition sofort sieht:

4.2 Lemma. Für jedes Alphabet A gilt:

für jedes $w_1 \in A^*$ und jedes $w_2 \in A^*$ ist $|w_1 w_2| = |w_1| + |w_2|$.

4.4.1 Konkatenation mit dem leeren Wort

Gefragt, was das Besondere an der Zahl Null ist, antworten zumindest manche Leute, dass es die Eigenschaft hat:

für jedes $x \in \mathbb{N}_0$ ist $x + 0 = x$ und $0 + x = x$.

Man sagt auch, die Null sei das *neutrale Element* bezüglich der Addition.

neutrales Element

Etwas Ähnliches wie die Null für natürliche Zahlen gibt es bei Wörtern: Das leere Wort ist das neutrale Element bezüglich Konkatenation.

4.3 Lemma. Für jedes Alphabet A gilt:

für jedes $w \in A^*$ ist $w \cdot \varepsilon = w$ und $\varepsilon \cdot w = w$.

Anschaulich ist das wohl klar: Wenn man ein Wort w nimmt und hinten dran der Reihe nach noch alle Symbole des leeren Wortes „klebt“, dann „ändert sich an w nichts“.

Aber da wir auch eine formale Definition von Wörtern haben, können wir das auch präzise beweisen ohne auf Anführungszeichen und „ist doch wohl klar“ zurückgreifen zu müssen. Wie weiter vorne schon einmal erwähnt: Wir machen das nicht, um Einfaches besonders schwierig darzustellen (so etwas hat angeblich Herr Gauß manchmal gemacht . . .), sondern um an einem einfachen Beispiel etwas zu üben, was Sie durch Ihr ganzes Studium begleiten wird: Beweisen.

4.4 Beweis. Die erste Frage, die sich stellt, ist: Wie beweist man das für alle denkbaren(?) Alphabete A ? Eine Möglichkeit ist: Man geht von einem wie man sagt „beliebigen aber festen“ Alphabet A aus, über das man *keinerlei weitere* Annahmen macht und zeigt, dass die Aussage für dieses A gilt.

Tun wir das: Sei im folgenden A ein beliebiges aber festes Alphabet.

Damit stellt sich die zweite Frage: Wie beweist man, dass die Behauptung für alle $w \in A^*$ gilt? Im vorliegenden Fall funktioniert das gleiche Vorgehen wieder: Man geht von einem beliebigen Wort w aus, über das man keinerlei Annahmen macht.

Sei also im folgenden w ein beliebiges aber festes Wort aus A^* , d. h. eine surjektive Abbildung $w : \mathbb{Z}_m \rightarrow B$ mit $B \subseteq A$.

Außerdem wissen wir, was das leere Wort ist: $\varepsilon : \mathbb{Z}_0 \rightarrow \{\}$.

Um herauszufinden, was $w' = w \cdot \varepsilon$ ist, können wir nun einfach losrechnen: Wir nehmen die formale Definition der Konkatenation und setzen unsere „konkreten“ Werte ein. Dann ist also w' eine Abbildung $w' : \mathbb{Z}_{m+0} \rightarrow B \cup \{\}$, also schlicht $w' : \mathbb{Z}_m \rightarrow B$. Und für alle $i \in \mathbb{Z}_m$ gilt für die Funktionswerte laut der formalen Definition von Konkatenation für alle $i \in \mathbb{Z}_m$:

$$\begin{aligned} w'(i) &= \begin{cases} w_1(i) & \text{falls } 0 \leq i < m \\ w_2(i-m) & \text{falls } m \leq i < m+n \end{cases} \\ &= \begin{cases} w(i) & \text{falls } 0 \leq i < m \\ \varepsilon(i-m) & \text{falls } m \leq i < m+0 \end{cases} \\ &= w(i) \end{aligned}$$

Also haben w und w' die gleichen Definitions- und Zielbereiche und für alle Argumente die gleichen Funktionswerte, d. h. an allen Stellen die gleichen Symbole. Also ist $w' = w$. ■

4.4.2 Eigenschaften der Konkatenation

Wenn man eine neue binäre Operation definiert, stellt sich immer die Frage nach möglichen Rechenregeln. Weiter oben haben wir schon darauf hingewiesen, dass man bei der Konkatenation von Wörtern nicht einfach die Reihenfolge vertauschen darf. (Man sagt auch, die Konkatenation sei *nicht kommutativ*.)

Was ist, wenn man mehrere Wörter konkateniert? Ist für jedes Alphabet A und alle Wörter w_1, w_2 und w_3 aus A^* stets

$$(w_1 \cdot w_2) \cdot w_3 = w_1 \cdot (w_2 \cdot w_3) ?$$

Die Antwort ist: Ja. Auch das kann man stur nachrechnen.

Das bedeutet, dass man bei der Konkatenation mehrerer Wörter keine Klammern setzen muss. Man sagt auch, die Konkatenation sei eine *assoziative Operation* (siehe Unterabschnitt 4.6).

4.4.3 Beispiel: Aufbau von E-Mails

RFC
Request For Comments

Die Struktur von E-Mails ist in einem sogenannten RFC festgelegt. RFC ist die Abkürzung für *Request For Comments*. Man findet alle RFCs zum Beispiel unter <http://tools.ietf.org/html/>.

RFC 5322

Die aktuelle Fassung der Spezifikation von E-Mails findet man in RFC 5322

(<http://tools.ietf.org/html/rfc5322>, 24.10.14). Wir zitieren und komentieren¹ einige Ausschnitte aus der Einleitung von Abschnitt 2.1 dieses RFC, wobei die Nummerierung/Strukturierung von uns stammt:

1.
 - „This document specifies that messages are made up of characters in the US-ASCII range of 1 through 127.“
 - Das Alphabet, aus dem die Zeichen stammen müssen, die in einer E-Mail vorkommen, ist der US-ASCII-Zeichensatz mit Ausnahme des Zeichens mit der Nummer 0.
2.
 - „Messages are divided into lines of characters. A line is a series of characters that is delimited with the two characters carriage-return and line-feed; that is, the carriage return (CR) character (ASCII value 13) followed immediately by the line feed (LF) character (ASCII value 10). (The carriage-return/line-feed pair is usually written in this document as "CRLF".)“
 - Eine Zeile (*line*) ist eine Folge von Zeichen, also ein Wort, das mit den beiden „nicht druckbaren“ Symbolen CR LF endet.
An anderer Stelle wird im RFC übrigens spezifiziert, dass als Zeile im Sinne des Standards nicht beliebige Wörter zulässig sind, sondern nur solche, deren Länge kleiner oder gleich 998 ist.
3.
 - A message consists of
 - [...] the header section of the message [...] followed,
 - optionally, by a body.“
 - Eine E-Mail (*message*) ist die Konkatenation von Kopf (*header*) der E-Mail und Rumpf (*body*) der E-Mail. Dass der Rumpf optional ist, also sozusagen fehlen darf, bedeutet nichts anderes, als dass der Rumpf auch das leere Wort sein darf. (Aus dem Rest des RFC ergibt sich, dass der Kopf nie das leere Wort sein kann.)
Aber das ist noch nicht ganz vollständig. Gleich anschließend wird der RFC genauer:
4.
 - – „The header section is a sequence of lines of characters with special syntax as defined in this specification.
– The body is simply a sequence of characters that follows the header and
– is separated from the header section by an empty line (i.e., a line with nothing preceding the CRLF). [...]“
 - Es gilt also:
 - Der Kopf einer E-Mail ist die Konkatenation (de facto mehrerer) Zeilen.

¹Wir erlauben uns kleine Ungenauigkeiten, weil wir nicht den ganzen RFC zitieren wollen.

- Der Rumpf einer E-Mail ist (wie man an anderer Stelle im RFC nachlesen kann) ebenfalls die Konkatenation von Zeilen. Es können aber auch 0 Zeilen oder 1 Zeile sein.
- Eine Leerzeile (*empty line*) ist das Wort `CR LF`.
- Eine Nachricht ist die Konkatenation von Kopf der E-Mail, einer Leerzeile und Rumpf der E-Mail.

4.4.4 Iterierte Konkatenation

Von den Zahlen kennen Sie die Potenzschreibweise x^3 für $x \cdot x \cdot x$ usw. Das wollen wir nun auch für die Konkatenation von Wörtern einführen. Die Idee ist so etwas wie

$$w^k = \underbrace{w \cdot w \cdots w}_{k \text{ mal}} .$$

Aber da stehen wieder diese Pünktchen ... Wie kann man die vermeiden? Was ist mit $k = 1$ (immerhin stehen da ja drei w auf der rechten Seite)? Und was soll man sich für $k = 0$ vorstellen?

Dafür benutzen wir wieder eine induktive Definition wie schon beim kartesischen Produkt mehrerer Mengen. Für *Potenzen von Wörtern* beginnen wir mit dem Fall $n = 0$:

$$w^0 = \varepsilon$$

für jedes $k \in \mathbb{N}_0 : w^{k+1} = w^k \cdot w$

Man kann nun ausrechnen, was w^1 ist:

$$w^1 = w^{0+1} = w^0 \cdot w = \varepsilon \cdot w = w .$$

Und dann:

$$w^2 = w^{1+1} = w^1 \cdot w = w \cdot w .$$

Und so weiter.

4.5 FORMALE SPRACHEN

Eine natürliche Sprache umfasst mehrere Aspekte, z. B. Aussprache und Stil, also z. B. Wortwahl und Satzbau. Dafür ist es auch notwendig zu wissen, welche Formulierungen syntaktisch korrekt sind. Neben den anderen genannten und ungenannten Punkten spielt *syntaktische Korrektheit* auch in der Informatik an vielen Stellen eine Rolle.

Bei der Formulierung von Programmen ist das jedem klar. Aber auch der Text, der beim Senden einer Email über das Netz transportiert wird oder der Quelltext einer HTML-Seite müssen bestimmten Anforderungen genügen. Praktisch immer, wenn ein Programm Eingaben liest, sei es aus einer Datei oder direkt vom Benutzer, müssen diese Eingaben gewissen Regeln genügen, sofern sie weiterverarbeitet werden können sollen. Wird z. B. vom Programm die Darstellung einer Zahl benötigt, dann ist vermutlich „101“ in Ordnung, aber „a*&W“ nicht. Aber natürlich (?) sind es bei jeder Anwendung andere Richtlinien, die eingehalten werden müssen.

Es ist daher nicht verwunderlich, wenn

- syntaktische Korrektheit,
- Möglichkeiten zu spezifizieren, was korrekt ist und was nicht, und
- Möglichkeiten, syntaktische Korrektheit von Texten zu überprüfen,

von großer Bedeutung in der Informatik sind.

Man definiert: Eine *formale Sprache* (über einem Alphabet A) ist eine Teilmenge $L \subseteq A^*$. *formale Sprache*

Immer, wenn es um syntaktische Korrektheit geht, bilden die syntaktisch korrekten Gebilde eine formale Sprache L , während die syntaktisch falschen Gebilde eben *nicht* zu L gehören.

Beispiele:

- Es sei $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -\}$. Die formale Sprache der Dezimaldarstellungen ganzer Zahlen enthält zum Beispiel die Wörter „1“, „-22“ und „192837465“, aber nicht „2-3--41“.
- Die formale Sprache der syntaktisch korrekten Java-Programme über dem Unicode-Alphabet enthält zum Beispiel nicht das Wort „[2] class int)“ (aber eben alle Java-Programme).

4.6 BINÄRE OPERATIONEN

Unter einer binären Operation auf einer Menge M versteht man eine Abbildung $f : M \times M \rightarrow M$. Üblicherweise benutzt man aber ein „Operationssymbol“ wie das Pluszeichen oder den Multiplikationspunkt und setzt ihn zwischen die Argumente: Statt $+(3, 8) = 11$ schreibt man normalerweise $3 + 8 = 11$.

Allgemein heißt eine binäre Operation $\diamond : M \times M \rightarrow M$ genau dann *kommutativ*, *kommutative Operation* wenn gilt:

$$\text{für jedes } x \in M \text{ und für jedes } y \in M \text{ ist } x \diamond y = y \diamond x .$$

Abgesehen davon, dass wir bald eine kompaktere Notation für „für jedes“ kennenlernen werden, könnte man etwas platzsparender auch schreiben

$$\text{für alle } x \in M \text{ und } y \in M \text{ ist } x \diamond y = y \diamond x .$$

Man beachte, dass dabei x und y durchaus auch für den gleichen Wert stehen dürfen. Die sich ergebende Forderung ist allerdings trivialerweise erfüllt.

Eine binäre Operation $\diamond : M \times M \rightarrow M$ nennt man genau dann *assoziativ*, wenn gilt:

$$\text{für alle } x \in M, y \in M \text{ und } z \in M \text{ ist } (x \diamond y) \diamond z = x \diamond (y \diamond z) .$$

Wir haben gesehen, dass die Konkatenation von Wörtern eine assoziative Operation ist, die aber *nicht* kommutativ ist.

ZUSAMMENFASSUNG UND AUSBLICK

In diesem Kapitel wurde eingeführt, was wir unter einem *Wort* verstehen wollen, und wie *Konkatenation* und *Potenzen* von Wörtern definiert sind.

Als wichtiges technisches Hilfsmittel haben wir erstmals eine *induktive Definition* gesehen. So etwas wird im weiteren Verlauf der Vorlesung immer wieder vorkommen.

Bisher haben wir an vielen Stellen Aussagen durch die Wörter „oder“ und „und“ miteinander verbunden. Da das auf Dauer etwas mühsam ist, werden wir uns kommenden Kapitel ein erstes Mal ausführlich mit Aussagenlogik beschäftigen und dabei auch Abkürzungen einführen, die kürzere, besser lesbare Formeln erlauben.