

Grundbegriffe der Informatik

Kapitel 4: Wörter

Mattias Ulbrich

(basierend auf Folien von Thomas Worsch)

KIT · Institut für Theoretische Informatik

Wintersemester 2023/2024

Wörter kennt jeder
formale Definition dafür: etwas zum Üben

Konkatenation von Wörtern ist deren Hintereinanderschreibung in einem Wort
ihre Eigenschaften

die Spezifikation von

Emails dient als Beispiel für die Nutzung eingeführter Begriffe

iterierte Konkatenation sehr nützlich und wieder mit einer induktiven Definition

formale Sprachen sind ein ganz wichtiges Konzept

binäre Operationen kommen oft vor, häufig mit netten Eigenschaften

Ein Wort über einem Alphabet A ist
eine Folge von Zeichen aus A .

Apfelmus

Ein Wort über einem Alphabet A ist
eine Folge von Zeichen aus A .

Apfelmus

Milchreis

Symbole dürfen mehrfach vorkommen.

Wo sind wir?

Definition des Begriffs „Wort“

Konkatenation von Wörtern

Eigenschaften der Konkatenation

Beispiel: Aufbau von E-Mails

Iterierte Konkatenation

Formale Sprachen

Binäre Operationen

Wort — formal definiert als Liste von Zeichen

- Sinn der Übung
 - an harmlosem Beispiel Wichtiges üben
 - nicht: Einfaches möglichst kompliziert darstellen

Wort — formal definiert als Liste von Zeichen

- Sinn der Übung
 - an harmlosem Beispiel Wichtiges üben
 - nicht: Einfaches möglichst kompliziert darstellen
- wesentlich an einer „Folge“ oder „Liste“ (von Zeichen)?

- Sinn der Übung
 - an harmlosem Beispiel Wichtiges üben
 - nicht: Einfaches möglichst kompliziert darstellen
- wesentlich an einer „Folge“ oder „Liste“ (von Zeichen)?
- Reihenfolge, deutlich gemacht durch **Nummerierung**:

0	1	2	3	4	5	6	7	8
M	i	l	c	h	r	e	i	s

- definiere für $n \in \mathbb{N}_0$ Menge der n kleinsten nichtnegativen ganzen Zahlen
 - $\mathbb{Z}_n = \{i \in \mathbb{N}_0 \mid 0 \leq i \text{ und } i < n\}$
 - Beispiele: $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, $\mathbb{Z}_1 = \{0\}$ und $\mathbb{Z}_0 = \{\}$
- Ein *Wort* (über dem Alphabet A) ist eine *Abbildung* $w : \mathbb{Z}_n \rightarrow A$
- Erinnerung: $\{w \mid w : \mathbb{Z}_n \rightarrow A\} = A^n$ (n -faches Kartesisches Produkt)
- n heißt die *Länge eines Wortes*, geschrieben $|w|$

- definiere für $n \in \mathbb{N}_0$ Menge der n kleinsten nichtnegativen ganzen Zahlen
 - $\mathbb{Z}_n = \{i \in \mathbb{N}_0 \mid 0 \leq i \text{ und } i < n\}$
 - Beispiele: $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, $\mathbb{Z}_1 = \{0\}$ und $\mathbb{Z}_0 = \{\}$
- Ein *Wort* (über dem Alphabet A) ist eine *Abbildung* $w : \mathbb{Z}_n \rightarrow A$
- Erinnerung: $\{w \mid w : \mathbb{Z}_n \rightarrow A\} = A^n$ (n -faches Kartesisches Produkt)
- n heißt die *Länge eines Wortes*, geschrieben $|w|$
- Beispiel: $w = \text{hallo}$
 - formal $w : \mathbb{Z}_5 \rightarrow \{a, h, l, o\}$ mit
 $w(0) = h$, $w(1) = a$, $w(2) = l$, $w(3) = l$ und $w(4) = o$.
 - aber auch $(h, a, l, l, o) \in \{a, h, l, o\}^5$

- Wo fangen Zahlen an?
 - als Kinder: Zählen ab Eins
 - später: die Null ist auch nützlich (für Mächtigkeiten)

- Wo fangen Zahlen an?
 - als Kinder: Zählen ab Eins
 - später: die Null ist auch nützlich (für Mächtigkeiten)
- Analogon bei Wörtern: das *leere Wort*
 - Es besteht aus 0 Symbolen.
 - wir schreiben ϵ

- Wo fangen Zahlen an?
 - als Kinder: Zählen ab Eins
 - später: die Null ist auch nützlich (für Mächtigkeiten)
- Analogon bei Wörtern: das *leere Wort*
 - Es besteht aus 0 Symbolen.
 - wir schreiben ε
- hilft Formalismus?
$$\varepsilon : \mathbb{Z}_0 \rightarrow A \quad \text{also} \quad \varepsilon : \{\} \rightarrow A$$
 - nur eine Relation $R \subseteq \{\} \times A = \{\}$, also $R = \{\} \times A = \{\}$
 - linkstotal, rechtseindeutig
 - also ein Wort
 - *Das leere Wort ist eindeutig.*

- Das leere Wort ist „etwas“.
- Die Kardinalität der Menge $\{\varepsilon, \text{abaa}, \text{bbbababb}\}$ ist
$$|\{\varepsilon, \text{abaa}, \text{bbbababb}\}| = 3$$
- Die Kardinalität der Menge $\{\varepsilon\}$ ist $|\{\varepsilon\}| = 1$
Das ist **nicht** die leere Menge!
- Die Kardinalität der Menge $\{\}$ ist $|\{\}| = 0$
Das **ist** die leere Menge.

A^* — die Menge aller Wörter

- A^* : *Menge aller Wörter* über Alphabet A
 - alle Wörter, die nur Zeichen aus A enthalten
- Beispiel: $A = \{a, b\}$.
Dann enthält A^* die Wörter
 - a und b
 - aa, ab, ba und bb
 - $aaa, aab, aba, abb, baa, bab, bba$ und bbb
 - ...
 - und außerdem ε , das *leere Wort*
- unendlich viele Wörter, die alle *endliche* Länge haben!
- A^* formal: die Menge aller Abbildungen
 $w : \mathbb{Z}_n \rightarrow A$ mit $n \in \mathbb{N}_0$.

A^n — die Wörter der Länge n

- A^n : Menge aller Wörter der Länge $n \in \mathbb{N}_0$ über dem Alphabet A
- Beispiel $A = \{a, b\}$
 - $A^0 = \{\varepsilon\}$
 - $A^1 = \{a, b\}$
 - $A^2 = \{aa, ab, ba, bb\}$
 - $A^3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$
- Also ist

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

- bessere Schreibweise ohne Pünktchen:

$$A^* = \bigcup_{n \in \mathbb{N}_0} A^n = \bigcup_{n \in \mathbb{N}_0} A^{\mathbb{Z}_n}$$

Sie erinnern sich?

$$\bigcup_{i \in \mathbb{N}_0} A^i = \{w \mid \text{es gibt ein } i \in \mathbb{N}_0 \text{ mit: } w \in A^i\}$$

Wo sind wir?

Definition des Begriffs „Wort“

Konkatenation von Wörtern

Eigenschaften der Konkatenation

Beispiel: Aufbau von E-Mails

Iterierte Konkatenation

Formale Sprachen

Binäre Operationen

Konkatenation — anschaulich:

Wörter werden hintereinander geschrieben

- Operationssymbol üblicherweise der Punkt „·“,
 - wie bei Multiplikation manchmal weggelassen
- Beispiele

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

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

- Reihenfolge!

$\text{SCHRANKSCHLÜSSEL} \neq \text{SCHLÜSSELSCHRANK}$

Konkatenation — formal:
aus zwei surjektiven Abbildungen wird eine dritte

■ Beispiel

0	1	2	3	4		0	1	2
A	P	F	E	L	·	M	U	S
0	1	2	3	4	5	6	7	
=	A	P	F	E	L	M	U	S

Konkatenation — formal:

aus zwei surjektiven Abbildungen wird eine dritte

■ Beispiel

$$\begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & & 0 & 1 & 2 \\ A & P & F & E & L & \cdot & M & U & S \\ \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ = & A & P & F & E & L & M & U & S \end{array}$$

Definition

- für $w_1 : \mathbb{Z}_m \rightarrow A$ und $w_2 : \mathbb{Z}_n \rightarrow A$ ist

$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

- übrigens: $|w_1 \cdot w_2| = |w_1| + |w_2|$

Definition

Es seien $w_1 : \mathbb{Z}_m \rightarrow A$ und $w_2 : \mathbb{Z}_n \rightarrow A$ zwei Wörter.
Wir definieren

$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

Definition

Es seien $w_1 : \mathbb{Z}_m \rightarrow A$ und $w_2 : \mathbb{Z}_n \rightarrow A$ zwei Wörter.
Wir definieren

$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

- Was tun mit so etwas?
 - **Nicht abschrecken lassen!**
 - Abbildung: *überall* definiert?
 - Fallunterscheidungen: widerspruchsfrei?
 - Hat das Definierte die erforderlichen Eigenschaften?
 - Verstehen!



$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

■ Überprüfung:

- ? Fallunterscheidung vollständig
- ? Fallunterscheidung widerspruchsfrei
- ? $w_1(i)$ für $0 \leq i < m$ und $w_2(i - m)$ für $m \leq i < m + n$ definiert
- ? Funktionswerte aus dem Bereich A



$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

■ Überprüfung:

- ✓ Fallunterscheidung vollständig
- ? Fallunterscheidung widerspruchsfrei
- ? $w_1(i)$ für $0 \leq i < m$ und $w_2(i - m)$ für $m \leq i < m + n$ definiert
- ? Funktionswerte aus dem Bereich A



$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

■ Überprüfung:

- ✓ Fallunterscheidung vollständig
- ✓ Fallunterscheidung widerspruchsfrei
- ? $w_1(i)$ für $0 \leq i < m$ und $w_2(i - m)$ für $m \leq i < m + n$ definiert
- ? Funktionswerte aus dem Bereich A



$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

■ Überprüfung:

- ✓ Fallunterscheidung vollständig
- ✓ Fallunterscheidung widerspruchsfrei
- ✓ $w_1(i)$ für $0 \leq i < m$ und $w_2(i - m)$ für $m \leq i < m + n$ definiert
- ? Funktionswerte aus dem Bereich A



$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \rightarrow A$$

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

■ Überprüfung:

- ✓ Fallunterscheidung vollständig
- ✓ Fallunterscheidung widerspruchsfrei
- ✓ $w_1(i)$ für $0 \leq i < m$ und $w_2(i - m)$ für $m \leq i < m + n$ definiert
- ✓ Funktionswerte aus dem Bereich A

Das leere Wort — **neutrales Element** bezüglich Konkatination

- bei den Zahlen:

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

Die Null ist das **neutrale Element** bezüglich der Addition.

- Analog bei Wörtern:
Für jedes Alphabet A gilt:

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

- anschaulich: klar (?)
- formal: können wir das jetzt beweisen

Lemma

Das leere Wort — neutrales Element bezüglich Konkatination

- Wie beweist man das für jedes denkbare Alphabet A ?
- Z. B.: betrachte „beliebiges“ Alphabet A ,
über das keine Annahmen gemacht werden.

Das leere Wort — neutrales Element bezüglich Konkatenation

- Wie beweist man das für jedes denkbare Alphabet A ?
 - Z. B.: betrachte „beliebiges“ Alphabet A ,
über das keine Annahmen gemacht werden.
- Wie beweist man die Behauptung für jedes $w \in A^*$?
 - Z. B.: betrachte „beliebiges“ Wort w ,
über das keine Annahmen gemacht werden.

Das leere Wort — neutrales Element bezüglich Konkatenation

- Wie beweist man das für jedes denkbare Alphabet A ?
 - Z. B.: betrachte „beliebiges“ Alphabet A ,
über das keine Annahmen gemacht werden.
- Wie beweist man die Behauptung für jedes $w \in A^*$?
 - Z. B.: betrachte „beliebiges“ Wort w ,
über das keine Annahmen gemacht werden.
- Also:
 - Es sei A ein Alphabet und $w \in A^*$, d. h. eine Abbildung $w : \mathbb{Z}_m \rightarrow A$
 - Außerdem ist $\varepsilon : \mathbb{Z}_0 \rightarrow A$.
 - berechne $w' = w \cdot \varepsilon$ anhand der formalen Definition:
 - w' ist eine Abbildung $w' : \mathbb{Z}_{m+0} \rightarrow A$, also $w' : \mathbb{Z}_m \rightarrow A$.

Das leere Wort — neutrales Element bezüglich Konkatination (2)

- $w' : \mathbb{Z}_m \rightarrow A$

- für $i \in \mathbb{Z}_m$ gilt

allgemein: $(w_1 \cdot w_2)(i)$

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

$$w'(i) = (w \cdot \varepsilon)(i) =$$

Das leere Wort — neutrales Element bezüglich Konkatination (2)

- $w' : \mathbb{Z}_m \rightarrow A$

- für $i \in \mathbb{Z}_m$ gilt

allgemein: $(w_1 \cdot w_2)(i)$

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

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

Das leere Wort — neutrales Element bezüglich Konkatination (2)

- $w' : \mathbb{Z}_m \rightarrow A$

- für $i \in \mathbb{Z}_m$ gilt

allgemein: $(w_1 \cdot w_2)(i)$

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

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

- w und w' haben gleichen Definitionsbereich
- w und w' haben gleichen Zielbereich
- w und w' haben überall die gleichen Funktionswerte.
- also $w' = w$, also $w \cdot \varepsilon = w$.

Wo sind wir?

Definition des Begriffs „Wort“

Konkatenation von Wörtern

Eigenschaften der Konkatenation

Beispiel: Aufbau von E-Mails

Iterierte Konkatenation

Formale Sprachen

Binäre Operationen

Eigenschaften der Konkatination — nicht kommutativ aber assoziativ

- Konkatination ist *nicht kommutativ*
 $\text{SCHRANKSCHLÜSSEL} \neq \text{SCHLÜSSELSCHRANK}$
- Reihenfolge ist wichtig

Eigenschaften der Konkatination — nicht kommutativ aber assoziativ

- Konkatination ist *nicht kommutativ*
 $\text{SCHRANKSCHLÜSSEL} \neq \text{SCHLÜSSELSCHRANK}$
- Reihenfolge ist wichtig

- Bei Zahlen gilt: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

- Konkatination ist *assoziativ*:

Für jedes Alphabet A und alle $w_1, w_2, w_3 \in A^*$ gilt:

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

Beweis: nachrechnen

Lemma

Wo sind wir?

Definition des Begriffs „Wort“

Konkatenation von Wörtern

Eigenschaften der Konkatenation

Beispiel: Aufbau von E-Mails

Iterierte Konkatenation

Formale Sprachen

Binäre Operationen

- RFC ist die Abkürzung für *Request For Comment*.
- alle RFCs zum Beispiel unter <http://tools.ietf.org/html/>
- Spezifikationen der *Internet Engineering Task Force (IETF)*

- z. B. Struktur von E-Mails
- nicht mehr aktuelle Fassung in RFC 5322
<http://tools.ietf.org/html/rfc5322>
- im folgenden einige Zitate aus Abschnitt 2.1 des RFC 5322
und Kommentare dazu

- *„This document specifies that messages are made up of characters in the US-ASCII range of 1 through 127.“*
- *Alphabet* für Emails
ASCII-Zeichensatz mit Ausnahme des Zeichens mit Nr. 0

- *„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 [...] by the
line feed (LF) character (ASCII value 10).
(The carriage-return/line-feed pair is usually written [...] as “CRLF”).“*
- Eine Zeile (*line*) ist
 - eine Folge von Zeichen, also ein Wort,
 - das mit den „nicht druckbaren“ Symbolen CR LF endet.
 - an anderer Stelle:
 - als Zeile nicht beliebige Wörter zulässig,
 - sondern nur solche, deren Länge kleiner oder gleich 998 ist.

- *A message consists of*
 - *[...] header fields [...] called the header section [...] followed,*
 - *optionally, by a body.“*
- eine E-Mail (*message*) ist die Konkatenation von
 - Kopf (*header section*) der E-Mail und
 - Rumpf (*body*) der E-Mail.
- Rumpf optional,
 - darf also auch fehlen,
 - d. h. der Rumpf darf auch das leere Wort sein.
- Das ist noch nicht ganz vollständig: ...

- „*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 section and*
- *is separated from the header section by an empty line (i.e., a line with nothing preceding the CRLF). [...]“*

also:

- Kopf einer E-Mail: Konkatenation (mehrerer) Zeilen.
- Rumpf einer E-Mail: Konkatenation von Zeilen.
 - darf auch 0 Zeilen oder 1 Zeile sein
- Leerzeile (*empty line*) ist das Wort `CR` `LF`.
- Email ist Konkatenation von
 - Kopf, einer Leerzeile und Rumpf

Wo sind wir?

Definition des Begriffs „Wort“

Konkatenation von Wörtern

Eigenschaften der Konkatenation

Beispiel: Aufbau von E-Mails

Iterierte Konkatenation

Formale Sprachen

Binäre Operationen

- bei Zahlen: Potenzschreibweise x^3 für $x \cdot x \cdot x$ usw.
- Ziel: analog für Wörter so etwas wie

$$w^n = \underbrace{w \cdot w \cdot \dots \cdot w}_{n \text{ mal}}$$

- Pünktchen sollte man vermeiden, denn
 - was ist mit $n = 1$? $n = 2$?
 - $n = 0$?
- Möglichkeit: eine *induktive Definition*
- für *Potenzen von Wörtern* geht das so:

$$\begin{aligned} w^0 &= \varepsilon \\ \forall n \in \mathbb{N}_0 : w^{n+1} &= w^n \cdot w \end{aligned}$$

- Definition

$$w^0 = \varepsilon$$

$$\forall n \in \mathbb{N}_0 : w^{n+1} = w^n \cdot w$$

- Was ist w^1 ?

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

- Was ist w^2 ?

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

- Was ist w^3 ?

$$w^3 = w^{2+1} = w^2 \cdot w = (w \cdot w) \cdot w$$

Lemma

Für jedes Alphabet A , jedes $w \in A^*$ und jedes $n \in \mathbb{N}_0$ gilt:

$$|w^n| = n|w| .$$

- Wie kann man das beweisen?
- Wenn etwas induktiv Definiertes eine Rolle spielt, sollte man *vollständige Induktion* als Beweismethode erwägen.
 - Details im übernächsten Kapitel

Wo sind wir?

Definition des Begriffs „Wort“

Konkatenation von Wörtern

Eigenschaften der Konkatenation

Beispiel: Aufbau von E-Mails

Iterierte Konkatenation

Formale Sprachen

Binäre Operationen

- natürliche Sprache:
 - Aussprache, Stil, Wortwahl, Satzbau, ...
 - Welche Formulierungen sind syntaktisch korrekt?
 - Ist und syntaktische nicht welcher korrekt?
- Informatik: künstliche Sprachen
 - Programmiersprachen
 - Aufbau von Emails, WWW-Seiten, ...
 - Eingabedateien für ...
- **Syntax**
 - Wie spezifiziert man, was wohlgeformt ist?
 - Wie überprüft man, ob etwas wohlgeformt ist?
- **Semantik**
 - Wie definiert man Bedeutung syntaktisch wolgeformter Gebilde?

Formale Sprache über einem Alphabet A — eine Teilmenge $L \subseteq A^*$

- im Zusammenhang mit syntaktischer Korrektheit:
 - formale Sprache L der syntaktisch wohlgeformte Gebilde
 - syntaktisch nicht-wohlgeformte Gebilde gehören *nicht* zu L
- Beispiele:
 - $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -\}$.
Die formale Sprache der Dezimaldarstellungen ganzer Zahlen
 - enthält z.B. 1, -22 und 192837465,
 - aber nicht 2-3--41 (obwohl das in A^* enthalten ist).
 - formale Sprache der syntaktisch korrekten Java-Programme
 - enthält alle Java-Programme
 - enthält zum Beispiel *nicht*: `[2} class int)(`

Wo sind wir?

Definition des Begriffs „Wort“

Konkatenation von Wörtern

Eigenschaften der Konkatenation

Beispiel: Aufbau von E-Mails

Iterierte Konkatenation

Formale Sprachen

Binäre Operationen

- *binäre Operation* auf Menge M ist eine Abbildung $f : M \times M \rightarrow M$
- oft Infixschreibweise
 - statt $+(3, 8) = 11$ schreibt man $3 + 8 = 11$
- binäre Operation $\diamond : M \times M \rightarrow M$ genau dann *kommutativ*, wenn gilt:

für jedes $x \in M$ für jedes $y \in M : x \diamond y = y \diamond x$.

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

für jedes $x \in M, y \in M, z \in M : (x \diamond y) \diamond z = x \diamond (y \diamond z)$.

nur dann Weglassen von Klammern: $x \diamond y \diamond z$ in Ordnung

- *binäre Operation* auf Menge M ist eine Abbildung $f : M \times M \rightarrow M$
- oft Infixschreibweise
 - statt $+(3, 8) = 11$ schreibt man $3 + 8 = 11$
- binäre Operation $\diamond : M \times M \rightarrow M$ genau dann *kommutativ*, wenn gilt:

$$\forall x \in M \quad \forall y \in M : x \diamond y = y \diamond x .$$

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

$$\text{für jedes } x \in M, y \in M, z \in M : (x \diamond y) \diamond z = x \diamond (y \diamond z) .$$

nur dann Weglassen von Klammern: $x \diamond y \diamond z$ in Ordnung

- *binäre Operation* auf Menge M ist eine Abbildung $f : M \times M \rightarrow M$
- oft Infixschreibweise
 - statt $+(3, 8) = 11$ schreibt man $3 + 8 = 11$
- binäre Operation $\diamond : M \times M \rightarrow M$ genau dann *kommutativ*, wenn gilt:

$$\forall x \in M \quad \forall y \in M : x \diamond y = y \diamond x .$$

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

$$\forall x \in M \quad \forall y \in M \quad \forall z \in M : (x \diamond y) \diamond z = x \diamond (y \diamond z) .$$

nur dann Weglassen von Klammern: $x \diamond y \diamond z$ in Ordnung

Das sollten Sie mitnehmen:

- ein *Wort* ist eine Folge von Symbolen
- eine *formale Sprache* ist eine Menge von Wörtern
- *induktive Definitionen*
erlauben, Pünktchen zu vermeiden ...

Das sollten Sie üben:

- Umgang mit formalen Definitionen
- „Rechnen“ mit Wörtern