

# Grundbegriffe der Informatik

## Tutorium 33

Lukas Bach, [lukas.bach@student.kit.edu](mailto:lukas.bach@student.kit.edu) | 4.11.2016



Lukas Bach, lu-  
kas.bach@student.kit.edu

Wiederholung

Wörter

1 Wiederholung

Formale  
Sprachen

2 Wörter

3 Formale Sprachen

## Wiederholung

## Wörter

## Formale Sprachen

$$A := \{a, b, c\}, B := \{b, c, d\}, C := \{a, d\}$$

- $A \cap B = \{b, c\}$
- $A \cup B = \{a, b, c, d\}$
- $A \setminus B = \{a\}$
- $C^2 = C \times C = \{(a, a), (a, d), (d, a), (d, d)\}$
- $2^C = \{\emptyset, \{a, d\}, \{a\}, \{d\}\}$
- Unterschied zwischen  $\{a, b\}$  und  $(a, b)$ ?
- Definition von...
  - Alphabet?
  - Abbildung?

## Konkatenation

Durch Konkatenation werden einzelne Buchstaben aus einem Alphabet miteinander verbunden.

- Symbol:  $\cdot$ , also zwei Buchstaben  $a$  und  $b$  miteinander konkateniert:  
 $a \cdot b$ .
- Nicht kommutativ:  $a \cdot b \neq b \cdot a$
- Aber assoziativ:  $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Kurzschreibweise: Ohne Punkte, also  $a \cdot b = ab$

## Wörter: Intuitivere Definition

Ein Wort  $w$  entsteht durch die Konkatenation durch Buchstaben aus einem Alphabet.

Also Abfolge von Zeichen.

Sei  $A := \{a, b, c\}$ .

- Mögliche Worte:  $w_1 := a \cdot b$ ,  $w_2 = b \cdot c \cdot c$ ,  $w_3 = a \cdot c \cdot c \cdot b \cdot a$ .
- Keine möglichen Worte:  $d$ .
- Konkatenation nicht kommutativ: Wort  $abc$  ist ungleich dem Wort  $bca$ .

## Wörter: Abstraktere Definition

Ein Wort  $w$  über dem Alphabet  $A$  ist definiert als surjektive Abbildung  $w : \mathbb{Z}_n \rightarrow A$ . Dabei heißt  $n$  die Länge  $|w|$  des Wortes.

- $\mathbb{Z}_n = \{i \in \mathbb{N} : 0 \leq i < n\}$   
 $\mathbb{Z}_3 = \{0, 1, 2\}, \mathbb{Z}_2 = \{0, 1\}, \mathbb{Z}_0 = \emptyset.$
- Länge oder Kardinalität eines Wortes:  $|w|$ .  $|abcde| = 5.$
- Wort  $w = abdec$  als Relation aufgeschrieben:  
 $w = \{(0, a), (1, b), (2, d), (3, e), (4, c)\}$ . Also  
 $w(0) = a, w(1) = b, w(2) = d, \dots$   
Damit sieht man auch:  
 $|w| = |\{(0, a), (1, b), (2, d), (3, e), (4, c)\}| = 5.$

- Wort der Kardinalität 0?

## Das leere Wort

Das leere Wort  $\varepsilon$  ist definiert ein Wort mit Kardinalität 0, also mit 0 Zeichen.

- Leere Wort wird interpretiert als “nicht sichtbar” und kann überall platziert werden:  $aabc = a\varepsilon abc = \varepsilon\varepsilon a\varepsilon bc\varepsilon$ .
- $|\{\varepsilon\}| = 1$ , die Menge ist nicht leer! Das leere Wort ist nicht *nichts*! (Vergleiche leere Menge)
- $|\varepsilon| = 0$ .

$A^n$

Zu einem Alphabet  $A$  ist  $A^n$  definiert als die Menge aller Wörter der Länge  $n$  über dem Alphabet  $A$ .

- Nicht mit Mengenpotenz verwechseln!
- $A := \{a, b, c\}$ ,  $A^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$ .  
 $A^1 = A$ ,  $A^0 = \{\varepsilon\}$ .

Die Menge aller Wörter *beliebiger* Länge:

- $A^* := \bigcup_{i \in \mathbb{N}_0} A_i$
- $A := \{a, b, c\}$ .  $aa \in A^*$ ,  $abcabcabc \in A^*$ ,  $aaaa \in A^*$ ,  $\varepsilon \in A^*$ .



## Konkatenation von Wörtern:

- $lager \cdot regal = lagerregal$
- $lag \cdot erregal = lagerregal$

## Konkatenation von Wörtern.

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

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

- Warum  $\mathbb{Z}_{m+n}$ ? Wörter  $w_1$  und  $w_2$  mit  $|w_1| = m$  und  $|w_2| = n$  werden konkateniert, also neues Wort hat Länge  $m + n$ .

## Konkatenation von Wörtern.

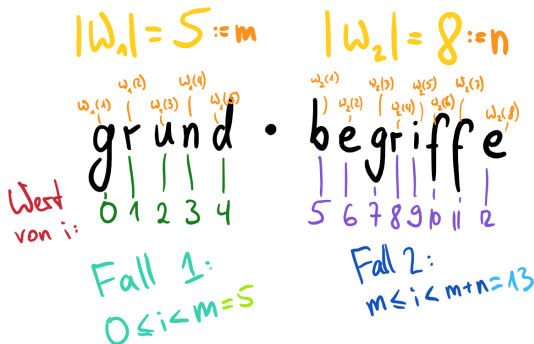
Wiederholung

Wörter

Formale  
Sprachen

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

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



## Wiederholung

## Wörter

## Formale Sprachen

- Immernoch: Reihenfolge ist wichtig!  
 $OTT \cdot O = OTTO \neq OOTT = O \cdot OTT$
- Auf wieviele Weisen kann man  $abc$  als Konkatination nichtleerer Wörter schreiben?  $abc$ ,  $a \cdot bc$ ,  $ab \cdot c$ ,  $a \cdot b \cdot c$ .
- Wortkonkatination mit dem leeren Wort:  $w \cdot \varepsilon = w = \varepsilon \cdot w$ .

## Wort Potenzen

Sich direkt wiederholende Teilworte kann man als Wortpotenz darstellen, daher  $w_i^n = w_i \cdot w_i \cdots w_i$  ( $n \times$  mal).

- $a^4 = aaaa$ ,  $b^3 = bbb$ ,  $c^0 = \varepsilon$ ,  $d^1 = d$ .
- $a^3 c^2 b^6 = aaaccbbbbb$ .
- $b \cdot a \cdot (n \cdot a)^2 = banana$ .
- $(a^3 b^2)^2 c (a^2 bcb^3)^3 dd = (aaabb)^2 c (aabcbbb)^3 dd$   
 $= aaabb \cdot aaabb \cdot c \cdot aabcbbb \cdot aabcbbb \cdot aabcbbb \cdot dd$ .

Sei  $A$  ein Alphabet.

## Übung zu Wörter

1. Finde Abbildung  $f : A^* \rightarrow A^*$ , sodass für alle  $w \in A^*$  gilt:  
 $2 \cdot |w| = |f(w)|$ .
2. Finde Abbildung  $g : A^* \rightarrow A^*$ , sodass für alle  $w \in A^*$  gilt:  
 $|w| + 1 = |g(w)|$ .
3. Finde Abbildung  $h : A^* \rightarrow A^*$ , sodass für alle  $w \in A^*$  gilt:  
 $\lfloor \frac{|w|}{2} \rfloor = |h(w)|$ . (Zusatz)
4. Sind  $f, g, h$  injektiv und/oder surjektiv?

1.  $f : A^* \rightarrow A^*, w \mapsto w \cdot w$ .

2.  $g : A^* \rightarrow A^*, w \mapsto w \cdot x, x \in A$ .

3.  $h : A^* \rightarrow A^*, w \mapsto \hat{w}$  mit  $\hat{w}_i = \begin{cases} w_i & \text{wenn } i \leq \lfloor \frac{|w|}{2} \rfloor \\ \varepsilon & \text{sonst} \end{cases}$  und  $i \in \mathbb{Z}_{|w|}$ .

## Wiederholung

## Wörter

## Formale Sprachen

1.  $f : A^* \rightarrow A^*, w \mapsto w \cdot w.$

- $f$  ist injektiv, denn jedes  $w$  aus der Bildmenge wird von maximal einem Wort abgebildet.
- $f$  ist nicht surjektiv, denn z.B. bildet nichts auf  $x \in A$  ab (oder auf andere Wörter mit ungerader Anzahl an Buchstaben).

2.  $g : A^* \rightarrow A^*, w \mapsto w \cdot x, x \in A.$

- $g$  ist injektiv.
- $g$  ist nicht surjektiv, denn z.B. bildet nichts auf  $\varepsilon$  ab.

3.  $h : A^* \rightarrow A^*, w \mapsto \hat{w}$  mit  $\hat{w}_i = \begin{cases} w_i & \text{wenn } i \leq \lfloor \frac{|w|}{2} \rfloor \\ \varepsilon & \text{sonst} \end{cases}$  und  $i \in \mathbb{Z}_{|w|}.$

- $h$  ist nicht injektiv, denn z.B.  $x = h(xy) = h(xz)$  mit  $x, y, z \in A.$
- $h$  ist surjektiv, denn für jedes  $w \in A^*$  existiert ein  $\hat{w} \in A^*$  mit  $\hat{w} = w \cdot w$  sodass  $h(\hat{w}) = w.$

- Was war nochmal  $A^*$ ? Menge aller Wörter *beliebiger* Länge über Alphabet  $A$ .

Wiederholung

## Formale Sprache

Eine Formale Sprache  $L$  über einem Alphabet  $A$  ist eine Teilmenge  $L \subseteq A^*$ .

Wörter

Formale  
Sprachen

- Zufälliges Beispiel:  $A := \{b, n, a\}$ .
  - $L_1 := \{ban, baan, nba, aa\}$  ist eine mögliche formale Sprache über  $A$ .
  - $L_2 := \{banana, bananana, banananana, \dots\}$   
 $= \{w : w = bana(na)^k, k \in \mathbb{N}\}$  auch.
  - $L_3 := \{ban, baan, baaan, \dots\}$  auch. Andere Schreibweise?  
 $L_3 = \{w : w = ba^k n, k \in \mathbb{N}\}$
- Formale Sprachen sind also nicht zwangsweise endliche Mengen.
- Praktischeres Beispiel:  $A := \{w : w \text{ ist ein ASCII Symbol}\}$ .
  - $L_4 := \{class, if, else, while, for, \dots\}$  ist eine formale Sprache über  $A$ .
  - $L_5 := \{w : w = a \cdot b \text{ mit } a \text{ als Großbuchstabe und } b \text{ als Groß- oder Kleinbuchstabe}\} \setminus L_4$  ist eine formale Sprache von korrekten Klassennamen in Java.

$A := \{a, b\}$

- Sprache  $L$  aller Wörter über  $A$ , die nicht das Teilwort  $ab$  enthalten?
  - Was passiert wenn ein solches Wort ein  $a$  enthält? Dann keine  $b$ 's mehr!
  - $L = \{w_1 \cdot w_2 : w_1 \in \{b\}^* \text{ und } w_2 \in \{a\}^*\}$
  - Andere Möglichkeit: Suche Wörter mit  $ab$  und nehme diese Weg.
  - $L = \{a, b\}^* \setminus \{w_1 \cdot ab \cdot w_2 : w_1, w_2 \in \{a, b\}^*\}$



Wiederholung

Wörter

Formale  
Sprachen

Sei  $A := \{a, b\}$ ,  $B := \{0, 1\}$ .

## Aufgabe zu formalen Sprachen

1. Sprache  $L_1 \subseteq A^*$  von Wörtern, die mindestens drei  $b$ 's enthalten.
2. Sprache  $L_2 \subseteq A^*$  von Wörtern, die gerade Zahl von  $a$ 's enthält.
3. Sprache  $L_3 \subseteq B^*$  von Wörtern, die, interpretiert als Binärzahl eine gerade Zahl sind.

1.  $L_1 = \{w = w_1bw_2bw_3bw_4 : w_1, w_2, w_3, w_4 \in A^*\}$
2.  $L_2 = \{w = (w_1aw_2aw_3)^* : w_1, w_2, w_3 \in \{b\}^*\}$  (Ist da  $\varepsilon$  drin?)
3.  $L_3 = \{w = w \cdot 0 : w \in B^*\}$

## Zum Tutorium

- Lukas Bach
- Tutorienfolien auf:
  - <http://gbi.lukasbach.com>
- Tutorium findet statt:
  - Donnerstags, 14:00 - 15:30
  - 50.34 Informatikbau, -107

## Mehr Material

- Ehemalige GBI Webseite:
  - <http://gbi.ira.uka.de>
  - Altklausuren!

## Zur Veranstaltung

- Grundbegriffe der Informatik
- Klausurtermin:
  - 06.03.2017, 11:00
  - Zwei Stunden Bearbeitungszeit
  - 6 ECTS für Informatiker und Informationswirte, 4 ECTS für Mathematiker und Physiker

## Zum Übungsschein

- Übungsblatt jede Woche
- Ab 50% insgesamt hat man den Übungsschein
- Keine Voraussetzung für die Klausur, aber für das Modul