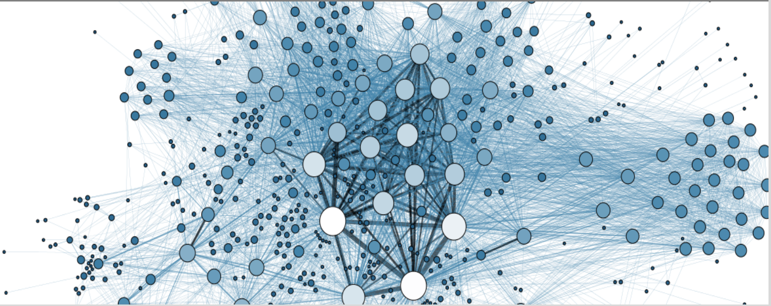
Karlsruher Institut für Technologie

lukasbach

Grundbegriffe der Informatik
Tutorium 33

Lukas Bach, lukas.bach@student.kit.edu | 4.11.2016



Lukas Bach, lukas.bach@student.kit.edu

Gliederung

Lukas Bach, lukas.bach@student.kit.edu

Wiederholung

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

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

Lukas Bach, lukas.bach@student.kit.edu

Wörter

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 $a \cdot b = ab$

Lukas Bach, lukas.bach@student.kit.edu

Wörter

Karlsruher Institut für Technologie

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

Lukas Bach, lukas.bach@student.kit.edu

Wörter: Abstraktere Definition

Ein Wort w über dem Alphabet A ist definiert als surjektive Abbildung $w : \mathbb{Z}_n \to A$. 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
  $w(0) = a, w(1) = b, w(2) = d, \ldots$
  Damit sieht man auch: $|w| = |\{(0, a), (1, b), (2, d), (3, e), (4, c)\}| = 5$.

Lukas Bach, lukas.bach@student.kit.edu

Wörter

- Wort der Kardinalität 0?

Das leere Wort

Das leere Wort ε 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)
- $|\varepsilon| = 0$.

Lukas Bach, lukas.bach@student.kit.edu

Mehr über Wörter

Karlsruher Institut für Technologie

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:

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

Lukas Bach, lukas.bach@student.kit.edu

Mehr über Wörter

Konkatenation von Wörtern:

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

Konkatenation von Wörtern.

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

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

- 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 $m + n$.

Lukas Bach, lukas.bach@student.kit.edu

Mehr über Wörter

Karlsruher Institut für Technologie

Konkatenation von Wörtern.

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

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

$|w_1| = 5 := m$ $\quad$ $|w_2| = 8 := n$

$w_1(1)$ $w_1(2)$ $w_1(3)$ $w_1(4)$ $w_1(5)$ $\quad$ $w_2(1)$ $w_2(2)$ $w_2(3)$ $w_2(4)$ $w_2(5)$ $w_2(6)$ $w_2(7)$ $w_2(8)$

grund · begriffe

Wert von i: 0 1 2 3 4 $\quad$ 5 6 7 8 9 10 11 12

Fall 1: $0 \leq i < m = 5$

Fall 2: $m \leq i < m+n = 13$

Lukas Bach, lukas.bach@student.kit.edu

Mehr über Wörter

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

Lukas Bach, lukas.bach@student.kit.edu

Mehr über Wörter

Karlsruher Institut für Technologie

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).

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

Lukas Bach, lukas.bach@student.kit.edu

Übung zu Wörter

Karlsruher Institut für Technologie

Sei A ein Alphabet.

Übung zu Wörter

1. Finde Abbildung $f : A^* \to A^*$, sodass für alle $w \in A^*$ gilt:
   $2 \cdot |w| = |f(w)|$.
2. Finde Abbildung $g : A^* \to A^*$, sodass für alle $w \in A^*$ gilt:
   $|w| + 1 = |g(w)|$.
3. Finde Abbildung $h : A^* \to A^*$, 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. $f : A^* \to A^*, w \mapsto w \cdot w$.
2. $g : A^* \to A^*, w \mapsto w \cdot x, x \in A$.
3. $h : A^* \to A^*, w \mapsto \hat{w}$ mit $\hat{w}_i = \begin{cases} w_i & \text{wenn } i \leq \lfloor \frac{|w|}{2} \rfloor \\ \varepsilon & \text{sonst} \end{cases}$ und $i \in \mathbb{Z}_{|w|}$.

Lukas Bach, lukas.bach@student.kit.edu

Übung zu Wörter

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 ε ab.
3. $h : A^* \rightarrow A^*, w \mapsto \hat{w}$ mit $\hat{w}_i = \left\{ \begin{array}{ll} w_i & \text{wenn } i \leq \lfloor \frac{|w|}{2} \rfloor \\ \varepsilon & \text{sonst} \end{array} \right\}$ 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 $h(\hat{w}) = w$.

Lukas Bach, lukas.bach@student.kit.edu

Formale Sprache

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

Formale Sprache

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

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

Lukas Bach, lukas.bach@student.kit.edu

Übung zu formalen Sprachen

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

- 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.
  - $L = \{a, b\}^* \setminus \{w_1 \cdot ab \cdot w_2 : w_1, w_2 \in \{a, b\}^*\}$

Lukas Bach, lukas.bach@student.kit.edu

Übung zu formalen 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. $L_1 = \{w = w_1 b w_2 b w_3 b w_4 : w_1, w_2, w_3, w_4 \in A^*\}$
2. $L_2 = \{w = (w_1 a w_2 a w_3)^* : w_1, w_2, w_3 \in \{b\}^*\}$ (Ist da ε drin?)
3. $L_3 = \{w = w \cdot 0 : w \in B^*\}$

Informationen

Lukas Bach, lukas.bach@student.kit.edu

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