

# ALGORITHMEN UND DATENSTRUKTUREN

## ÜBUNG 1: EINLEITUNG

---

Eric Kunze

`eric.kunze@mailbox.tu-dresden.de`

TU Dresden, 27. November 2019

*Wer bin ich?*

- ▶ Eric [Kunze]
- ▶ `eric.kunze@mailbox.tu-dresden.de`
- ▶ Fragen, Wünsche, Vorschläge, ...  
gern persönlich vor/nach der Übung  
oder per Mail



*Wer seid ihr?*

- ▶ <https://invote.tu-dresden.de/78190>



## Lehrveranstaltungswebsite:

<https://www.orchid.inf.tu-dresden.de/teaching/2019ws/aud/>  
(Google: "aud tu dresden")

- ▶ alle Informationen zur Lehrveranstaltung
- ▶ aktuelles Übungsblatt
- ▶ Link zu Lösungsvorschlägen
- ▶ Übungsverlegungen

## Skript & Aufgabensammlung: Copyshop "Die Kopie"

### Übungsverlegung für Donnerstag, 31. Oktober 2019

- |                   |                   |
|-------------------|-------------------|
| ▶ Montag, 4. DS   | ▶ Mittwoch, 2. DS |
| ▶ Dienstag, 6. DS | ▶ Mittwoch, 3. DS |
|                   | ▶ Mittwoch, 6. DS |

## Was ist eine Übung?

„Lehrveranstaltung an der Hochschule, in der etw., bes. das Anwenden von Grundkenntnissen, von den Studierenden geübt wird“ [Duden]

<b>Vorlesung</b>	<b>Übung</b>
Vermittlung von neuem Wissen	Üben und Festigen des Stoffes der VL
hohes Tempo	(selbst definierbares) langsames Tempo
wenig Interaktion	(sehr) viel Interaktion
≈ 50% Verständnis	> 80% Verständnis

# WAS WIRD IN DER ÜBUNG ERWARTET?

**Es wird keine reine Vorrechenübung werden!**

## Mein Input

- ▶ Zusammenfassung einiger Vorlesungsinhalte
- ▶ Zeit zur Rekapitulation
- ▶ beispielhafte Lösungsansätze und Lösungen
- ▶ Fragen, Fragen, Fragen

## Euer Input

- ▶ Grundverständnis aus der Vorlesung
- ▶ Vorbereitung der Übungsaufgaben
- ▶ aktive Mitarbeit und FRAGEN

**Slides werden mit Sourcecode auf Github zur Verfügung stehen.**

- ▶ `https://github.com/oakonerik/algorithmen-datenstrukturen-ws19`
- ▶ `github.com` → `oakonerik` → `algorithmen-datenstrukturen-ws19`
- ▶ evtl. zusätzliche Materialien (nach Bedarf)
- ▶ **kein Anspruch auf Vollständigkeit & Korrektheit**
- ▶ gefundene Fehler melden - am besten per Pull-Request

# Übungsblatt 1

---

# AUFGABE 1

Begriff	Erklärung
Syntax	Struktur (einer Sprache), erlaubte Zeichenketten
Semantik	Bedeutung der Zeichenketten
Objektsprache	(syntaktisch) zu beschreibende Sprache
Metasprache	Hilfssprache zur Beschreibung der Objektsprache
Alphabet $\Sigma$	nichtleere, endliche Menge von Terminalsymbolen, Zeichenvorrat
Wort	endliche Folge von Symbolen
Konkatenation	Verkettung von Wörtern



## AUFGABE 1 (FORTSETZUNG)

Begriff	Erklärung
Potenzmenge $\mathcal{P}$	Menge aller Teilmengen
$\Sigma^*$	Menge aller Wörter über $\Sigma$
$\mathcal{P}(\Sigma^*)$	Menge aller Sprachen über $\Sigma$
formale Sprache $L$	Menge von Wörtern über $\Sigma$ $L \in \mathcal{P}(\Sigma^*)$
Komplexprodukt "."	Verknüpfung von Sprachen $L_1 \cdot L_2 = \{uv \mid u \in L_1, v \in L_2\}$
$L^*$	Menge aller Konkatenationen von Wörtern aus $L$ $L^* = \bigcup_{n \geq 0} L^n$ mit $L^0 = \{\varepsilon\}$ und $L^{n+1} = L^n \cdot L$

## AUFGABE 2

Sei  $\Sigma = \{1, 2, a, b\}$ .

- **Wörter** ... entstehen durch Konkatenation von Symbolen

z.B.  $\varepsilon, 1, 2, a, b, 12, 1a, 1b, 21, 22, 2a, 2b, ab, abba, \dots$

- Symbole  $\xrightarrow{"\cdot"} \text{Wörter} \xrightarrow{\in} \underbrace{\Sigma^*}_{\text{Menge 1. Ordnung}} \xrightarrow{\in} \underbrace{\mathcal{P}(\Sigma^*)}_{\text{Menge 2. Ordnung}}$

- **Sprache**  $L$  ... Menge von Wörtern, d.h.  $L \subseteq \Sigma^*$  bzw.  
 $L \in \mathcal{P}(\Sigma^*)$ , z.B.

$$\begin{aligned} L &= \{1a, 1b, 1aa, 1bb, 1ab, 1aab, \dots\} = \{1a^n b^m : n, m \geq 0\} \\ &= \{1\} \cdot \{a\}^* \cdot \{b\}^* \end{aligned}$$

**Beachte:**  $\emptyset \in \mathcal{P}(\Sigma^*)$  und  $\varepsilon \in \Sigma^*$

## AUFGABE 3

Seien  $L_1 = \{a\}$ ,  $L_2 = \{b\}$ ,  $L_3 = \{a, ba\}$ .

- ▶  $L_1 \cdot L_2 \cdot L_3 = \{aba, abba\}$
- ▶  $L_1^* = \{a\}^* = \{\varepsilon, a, aa, aaa, \dots\} = \{a^n : n \geq 0\}$
- ▶  $L_3^* = \{\varepsilon, a, ba, aa, aba, baa, baba, \dots\} =$   
 $\{a^{m_1}(ba)^{n_1} \dots a^{m_k}(ba)^{n_k} : m_i, n_i \in \mathbb{N}, k \in \mathbb{N}^+, 1 \leq i \leq k\}$
- ▶  $L_2^* \cdot L_1 = \{a, ba, bba, bbba, \dots\} = \{b^n a : n \geq 0\}$
- ▶  $\mathcal{P}(L_1^*) =$   
 $\{\emptyset, \{\varepsilon\}, \{a\}, \{aa\}, \{aaa\}, \dots, \{\varepsilon, a\}, \{\varepsilon, aa\}, \{\varepsilon, aaa\}\} =$   
 $\{\{a^n : n \in I\} : I \subseteq \mathbb{N}\}$

## Euklid: Satz 4 in Buch II der "Elemente"

Wird eine Strecke in zwei geteilt, dann ist das Quadrat über der ganzen Strecke gleich den Quadraten über den Teilen und dem doppelten Rechteck, das die Teile ergeben, zusammen.

siehe <http://www.opera-platonis.de/euklid/Buch2.pdf>

## al-Khwarizmi in Al-jabr wa'l muqabalah'

What must be the amount of a square, which, when twenty-one dirhems are added to it, becomes equal to the equivalent of ten roots of that square?

**Solution:** Halve the number of the roots; the moiety is five. Multiply this by itself; the product is twenty-five. Subtract from this the twenty-one which are connected with the square; the remainder is four. Extract its root; it is two. Subtract this from the moiety of the root, which is five; the remainder is three. This is the root of the square which you required, and the square is nine. Or you may add the root of the moiety of the roots; the sum is seven; this is the root of the square which you sought for, and the square itself is forty nine.

## Definition (Kleene-Stern)

Für eine formale Sprache  $L$  definieren wir

$$L^* = \bigcup_{n \geq 0} L^n = \bigcup_{n=0}^{\infty} L^n$$

wobei  $L^0 = \{\varepsilon\}$  und  $L^{n+1} = L^n \cdot L$ .

Beachte:  $\{\varepsilon\}^* = \emptyset^* = \{\varepsilon\}$