

# ALGORITHMEN UND DATENSTRUKTUREN

## ÜBUNG 2: SYNTAXDIAGRAMME & EBNF

---

Eric Kunze

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

TU Dresden, 06.11.2020

## Prof. Dr. Markus Krötzsch:

- ▶ <https://youtu.be/Lma6jaPnD-I>

## Tutorials für C:

- ▶ freeCodeCamp.org  
[https://www.youtube.com/watch?v=KJgsSF0SQv0&ab\\_channel=freeCodeCamp.org](https://www.youtube.com/watch?v=KJgsSF0SQv0&ab_channel=freeCodeCamp.org)
- ▶ SEPL Goethe University Frankfurt  
[https://www.youtube.com/watch?v=CeEfTlRFEA0&t=113s&ab\\_channel=SEPLGoetheUniversityFrankfurt](https://www.youtube.com/watch?v=CeEfTlRFEA0&t=113s&ab_channel=SEPLGoetheUniversityFrankfurt)
- ▶ Caleb Curry:  
[https://www.youtube.com/watch?v=Bz4MxDeEM6k&list=PL\\_c9BZZLwBRKKq0c9TJz1pP0ASrxLMtp2&ab\\_channel=CalebCurry](https://www.youtube.com/watch?v=Bz4MxDeEM6k&list=PL_c9BZZLwBRKKq0c9TJz1pP0ASrxLMtp2&ab_channel=CalebCurry)

# Syntaxdiagramme

---

- ▶ syntaktische Variable = Nichtterminalsymbol = Name eines Syntaxdiagramms
- ▶ Jedes Kästchen ist mit dem Namen eines Syntaxdiagramms beschriftet.
- ▶ Jedes Oval ist mit einem Terminalsymbol beschriftet.

## **Rücksprungalgorithmus**

- ▶ jedes Kästchen bekommt eindeutige Marke (Rücksprungadresse)
- ▶ beim Betreten eines Syntaxdiagramms wird eine Marke auf den Keller gelegt
- ▶ Nachweis von Zugehörigkeit eines Wortes zu einer Sprache

# AUFGABE 1

- ▶ Teil (a) — z.B.  $\varepsilon, a, c, caa, aaaa, \dots$
- ▶ Teil (b) — z.B.  $aaac, abacac, abbaccac, \dots$
- ▶ Teil (c) — z.B.  $\varepsilon, ab, abab, ac, aabcab, \dots$

## AUFGABE 2 — TEIL (A)

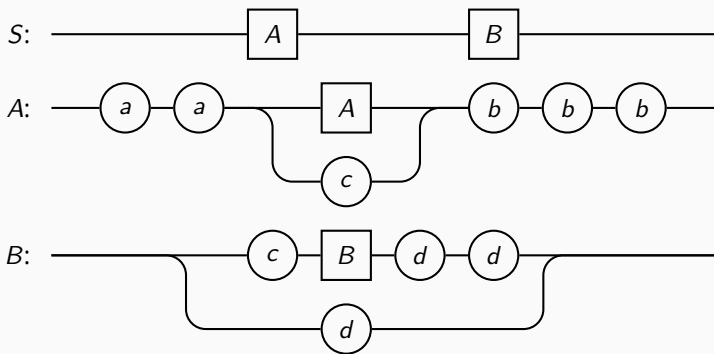
### Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶  $\mathcal{Z}$  = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	$\mathcal{Z}$ 2131
aaaaccb	$\mathcal{Z}$ 131
aaaaccbd	$\mathcal{Z}$ 131
aaaaccbdb	$\mathcal{Z}$ 1
aaaaccbdb	$\mathcal{Z}$
aaaaccbdbb	–

## AUFGABE 2 — TEIL (B)

$$\begin{aligned} L &= \{a^{2i}cb^{3i}c^kd^{2k+1} \mid i > 0, k \geq 0\} \\ &= \{a^{2i}cb^{3i} \mid i > 0\} \cdot \{c^kd^{2k+1} \mid k \geq 0\} \end{aligned}$$



# Extended Backus-Naur-Form

---



# EBNF-DEFINITION

- ▶ EBNF-Definition besteht aus endlicher Menge von EBNF-Regeln.
- ▶ Jede EBNF-Regel besteht aus einer linken und einer rechten Seite, die rechte Seite ist ein EBNF-Term.

## Definition: EBNF-Term

Seien  $V$  eine endliche Menge (syntaktische Variablen) und  $\Sigma$  eine endliche Menge (Terminalsymbole) mit  $V \cap \Sigma = \emptyset$ . Die Menge der EBNF-Terme über  $V$  und  $\Sigma$  (notiere:  $T(\Sigma, V)$ ), ist die *kleinste* Menge  $T \subseteq \left( V \cup \Sigma \cup \left\{ \hat{\{ \}}, \hat{\} \}, \hat{[ \]}, \hat{[ \}}, \hat{( \)}, \hat{( \}} \right\} \right)$  mit  $V \subseteq T, \Sigma \subseteq T$  und

- ▶ Wenn  $\alpha \in T$ , so auch  $\hat{(\alpha)} \in T, \hat{\{\alpha\}} \in T, \hat{[\alpha]} \in T$ .
- ▶ Wenn  $\alpha_1, \alpha_2 \in T$ , so auch  $\hat{(\alpha_1 \mid \alpha_2)} \in T, \alpha_1 \alpha_2 \in T$

## AUFGABE 3 — TEIL (A)

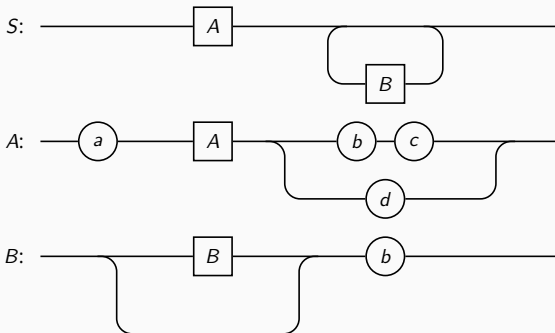
EBNF-Definition  $\mathcal{E} = (V, \Sigma, S, R)$  mit  $\Sigma = \{a, b, c, d\}$ ,

$$V = \{S, A, B\} \quad \text{und} \quad R = \left\{ S ::= A \hat{\ } B \hat{\ }, \right.$$

$$A ::= aA \hat{\ } (bc \mid d) \hat{\ },$$

$$B ::= \hat{\ } B \hat{\ } b \left. \right\}$$

Übersetzung in Syntaxdiagrammsystem:



## AUFGABE 3 — TEIL (B)

Gegeben sei die Sprache

$$L = \left\{ (ab)^n c^{m+1} d^k b^{n+m} : n, m \geq 0, k \geq 1 \right\}$$

Gesucht ist eine zugehörige EBNF-Definition.

$$L = \left\{ (ab)^n c^{m+1} d^k b^m b^n : n, m \geq 0, k \geq 1 \right\}$$

**EBNF-Definition:**  $\mathcal{E} = (V, \Sigma, S, R)$  mit  $\Sigma = \{a, b, c, d\}$ ,

$$\begin{aligned} V = \{S, A\} \quad \text{und} \quad R = \left\{ S ::= \hat{(} abSb \hat{)} A \hat{)}, \right. \\ \left. A ::= \hat{(} cAb \hat{)} cd \{ d \} \hat{)} \right\} \end{aligned}$$