

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 2: SYNTAXDIAGRAMME & EBNF

Eric Kunze

`eric.kunze@tu-dresden.de`

TU Dresden, 21. Oktober 2021

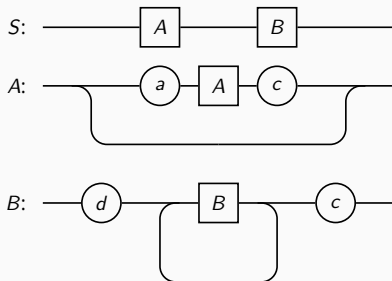
Prof. Dr. Markus Krötzsch hat im vergangenen Wintersemester 2020/21 die Vorlesung „Formale Systeme“ (3. Semester) in Form von YouTube-Videos gehalten. Diese Vorlesung beschäftigt sich vertieft mit formalen Sprachen. Die Einleitung entspricht ungefähr dem Inhalt der ersten Übung:

- ▶ <https://youtu.be/Lma6jaPnD-I>

Syntaxdiagramme

SYNTAXDIAGRAMME

Beispiel eines Syntaxdiagrammsystems mit Startdiagramm S:



□ ... Nichtterminalsymbol = syntaktische Variable

○ ... Terminalsymbol

Rücksprungalgorithmus

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

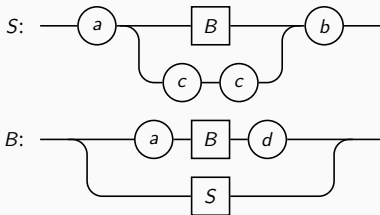
Hauptaugenmerk:

Protokollierung von Wortentstehung & Markenkeller

- ▶ jede Zeile entspricht dem Aufenthalt in einem Syntaxdiagramm
- ▶ jede Zeile führt eine Operation auf dem Markenkeller durch

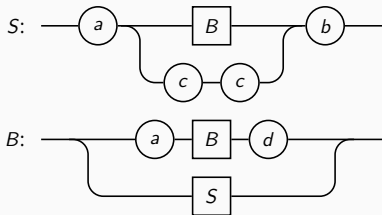
AUFGABE 1 — TEIL (A)

Gegeben sei das folgende Syntaxdiagrammsystem \mathcal{U} mit Startdiagramm S :



AUFGABE 1 — TEIL (A)

Gegeben sei das folgende Syntaxdiagrammsystem \mathcal{U} mit Startdiagramm S :



Beispiele für Wörter, die das System \mathcal{U} erzeugt:

- ▶ $a accb b$
- ▶ $aa accb bb$
- ▶ $aa accb db$
- ▶ $aaa accb ddb$
- ▶ $aa a accb bdb$

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Wort	Markenkeller
------	--------------

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Wort	Markenkeller
a	1

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Wort	Markenkeller
a	1
a	31

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	β 2131

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	β 2131
aaaaccb	β 131

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	β 2131
aaaaccb	β 131
aaaaccbd	β 131

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	β 2131
aaaaccb	β 131
aaaaccbd	β 131
aaaaccbdb	β 1

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	3 2131
aaaaccb	2 1 31
aaaaccbd	1 3 1
aaaaccbdb	3 1
aaaaccbdb	1 3

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

Protokollierungszeitpunkte:

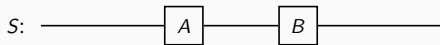
- ▶ jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- ▶ jede Zeile führt eine Operation auf dem Markenkeller aus
- ▶ β = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	3 2131
aaaaccb	2131
aaaaccbd	131
aaaaccbdb	3 1
aaaaccbdb	1
aaaaccbdbb	-

$$L = L_A \cdot L_B$$

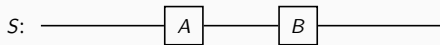
GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$$L = L_A \cdot L_B$$



GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

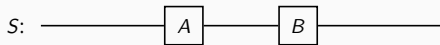
$$L = L_A \cdot L_B$$



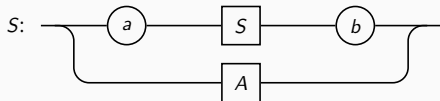
$$L = \{a^n L_A b^n : n \geq 0\}$$

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$$L = L_A \cdot L_B$$

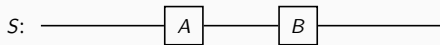


$$L = \{a^n L_A b^n : n \geq 0\}$$

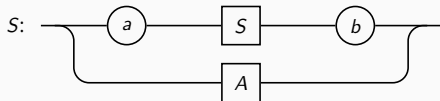


GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$$L = L_A \cdot L_B$$



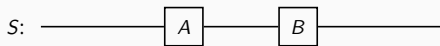
$$L = \{a^n L_A b^n : n \geq 0\}$$



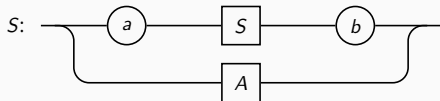
$$L = \{a^n L_A b^n : n > 0\}$$

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

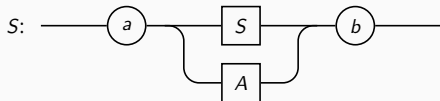
$$L = L_A \cdot L_B$$



$$L = \{a^n L_A b^n : n \geq 0\}$$

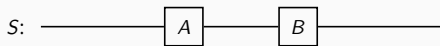


$$L = \{a^n L_A b^n : n > 0\}$$

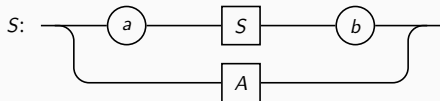


GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

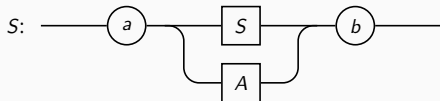
$$L = L_A \cdot L_B$$



$$L = \{a^n L_A b^n : n \geq 0\}$$



$$L = \{a^n L_A b^n : n > 0\}$$



kleine Tricks:

- ▶ $a^{2n} = (a^2)^n = (aa)^n$
- ▶ $a^{2n+1} = a a^{2n} = a (aa)^n$

AUFGABE 1 — TEIL (C)

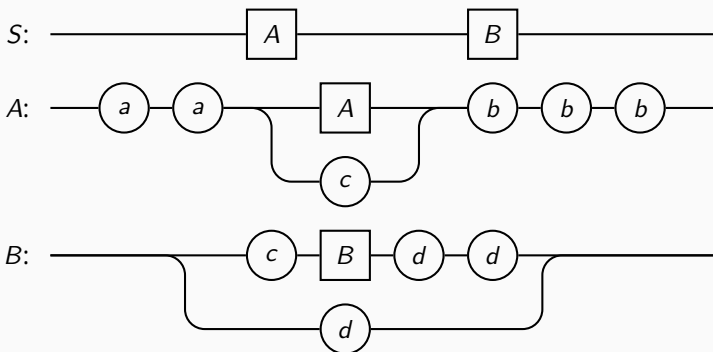
$$L = \{a^{2i}cb^{3i}c^kd^{2k+1} \mid i > 0, k \geq 0\}$$

AUFGABE 1 — TEIL (C)

$$\begin{aligned}L &= \{a^{2i}cb^{3i}c^k d^{2k+1} \mid i > 0, k \geq 0\} \\&= \{a^{2i}cb^{3i} \mid i > 0\} \cdot \{c^k d^{2k+1} \mid k \geq 0\} \\&= \{(aa)^i c (bbb)^i \mid i > 0\} \cdot \{c^k d (dd)^k \mid k \geq 0\}\end{aligned}$$

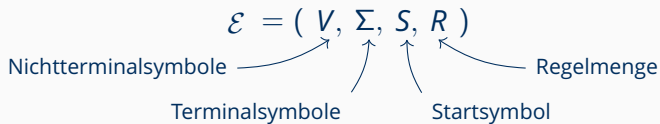
AUFGABE 1 — TEIL (C)

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



Extended Backus-Naur-Form

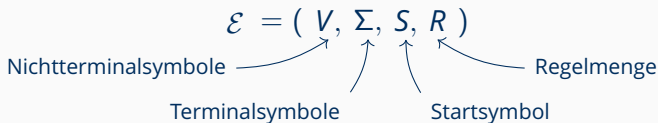
EBNF-DEFINITION



Jede **EBNF-Regel** besteht aus einer linken und einer rechten Seite, die rechte Seite ist ein **EBNF-Term**.

Nichtterminalsymbol ::= EBNF-Term

EBNF-DEFINITION



Jede **EBNF-Regel** besteht aus einer linken und einer rechten Seite, die rechte Seite ist ein **EBNF-Term**.

Nichtterminalsymbol ::= EBNF-Term

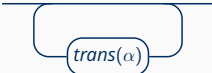
Definition (EBNF-Terme): Seien V (syntaktische Variablen) und Σ (Terminalsymbole) endliche Mengen mit $V \cap \Sigma = \emptyset$. Die Menge der EBNF-Terme über V und Σ (notiere: $T(\Sigma, V)$), ist die *kleinste* Menge $T \subseteq (V \cup \Sigma \cup \{ \{ \}, \{ \}, [\], (\), \hat{ \ } \})$ mit $V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

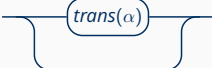
- ▶ Wenn $\alpha \in T$, so auch $\hat{\alpha} \in T$, $\{ \alpha \} \in T$, $[\alpha] \in T$.
- ▶ Wenn $\alpha_1, \alpha_2 \in T$, so auch $\hat{\alpha_1 \mid \alpha_2} \in T$, $\alpha_1 \alpha_2 \in T$.

ÜBERSETZUNG EBNF \leftrightarrow SYNTAXDIAGRAMME

Sei $v \in V$ und $w \in \Sigma$. $trans(v) = \boxed{v}$; $trans(w) = \bigcirc w$

Sei $\alpha \in T(\Sigma, V)$ ein EBNF-Term.


► $trans(\hat{\{\alpha\}}) =$  A horizontal line with a rounded rectangle labeled $trans(\alpha)$ connected to it by a U-shaped bracket above the line.

► $trans(\hat{[\alpha]}) =$  A horizontal line with a rounded rectangle labeled $trans(\alpha)$ connected to it by a U-shaped bracket below the line.

► $trans(\hat{(\alpha)}) = trans(\alpha)$

Seien $\alpha_1, \alpha_2 \in T(\Sigma, V)$ zwei EBNF-Terme.

► $trans(\alpha_1\alpha_2) =$  A horizontal line with two rounded rectangles labeled $trans(\alpha_1)$ and $trans(\alpha_2)$ connected in sequence.

► $trans(\hat{(\alpha_1 \mid \alpha_2)}) =$  A horizontal line with two rounded rectangles labeled $trans(\alpha_1)$ and $trans(\alpha_2)$ connected in parallel by a U-shaped bracket below the line.

AUFGABE 2 — TEIL (A)

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

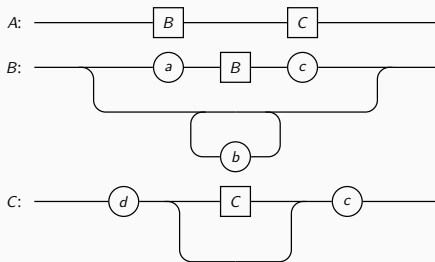
$$V = \{A, B, C\} \quad \text{und} \quad R = \left\{ \begin{array}{l} A ::= BC, \\ B ::= \hat{a} B c \hat{\{ b \}}, \\ C ::= d \hat{C} c \end{array} \right\}$$

AUFGABE 2 — TEIL (A)

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

$$V = \{A, B, C\} \quad \text{und} \quad R = \left\{ \begin{array}{l} A ::= BC, \\ B ::= (\hat{a}Bc \mid \hat{\{b\}}), \\ C ::= d \hat{[C]} c \end{array} \right\}$$

Übersetzung in ein Syntaxdiagrammsystem:



Startdiagramm: A

AUFGABE 2 — TEIL (B)

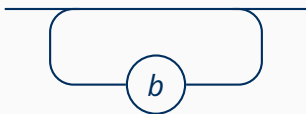
Wir wollen die von \mathcal{E} beschriebene Sprache L_A beschreiben und wenden dafür die Grundkonstruktionen „rückwärts“ an.

AUFGABE 2 — TEIL (B)

Wir wollen die von \mathcal{E} beschriebene Sprache L_A beschreiben und wenden dafür die Grundkonstruktionen „rückwärts“ an.

$$\begin{aligned}L_A &= L_B \cdot L_C \\ &= \{a^n w c^n : w \in \{b\}^*, n \geq 0\} \cdot \{d^m c^m : m \geq 1\}\end{aligned}$$

Der Teil $w \in \{b\}^*$ beschreibt dabei, dass wir ein beliebiges Wort w aus $\{b\}^*$ zwischen die a 's und c 's schreiben. Diese Sprache $\{b\}^*$ wird durch



beschrieben.

AUFGABE 2 — TEIL (C)

Gegeben sei die Sprache

$$L = \{ a^{n+\ell} cb^n (cd)^\ell : n, \ell \in \mathbb{N}, n \geq 1 \}$$

Gesucht ist eine zugehörige EBNF-Definition.

AUFGABE 2 — TEIL (C)

Gegeben sei die Sprache

$$L = \{ a^{n+\ell} c b^n (cd)^\ell : n, \ell \in \mathbb{N}, n \geq 1 \}$$

Gesucht ist eine zugehörige EBNF-Definition.

$$L = \{ a^\ell a^n c b^n (cd)^\ell : n, \ell \in \mathbb{N}, n \geq 1 \}$$

AUFGABE 2 — TEIL (C)

Gegeben sei die Sprache

$$L = \{ a^{n+\ell} c b^n (cd)^\ell : n, \ell \in \mathbb{N}, n \geq 1 \}$$

Gesucht ist eine zugehörige EBNF-Definition.

$$L = \{ a^\ell a^n c b^n (cd)^\ell : n, \ell \in \mathbb{N}, n \geq 1 \}$$

Lösungsweg: via Syntaxdiagrammsystem & Übersetzung

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

$$V = \{S, A\} \quad \text{und} \quad R = \left\{ \begin{array}{l} S ::= \hat{a} S c d \hat{A} \hat{A} \\ A ::= a \hat{A} c \hat{A} b \end{array} \right\}$$