

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 2: SYNTAXDIAGRAMME & EBNF

Eric Kunze

`eric.kunze@tu-dresden.de`

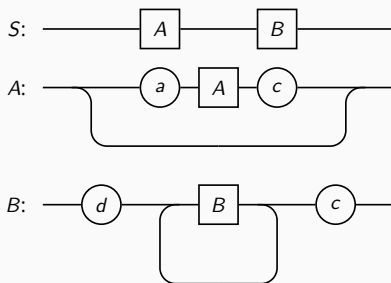
TU Dresden, 06.11.2020

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

Beispiel eines Syntaxdiagrammsystems mit Startdiagramm S :



\boxed{A} ... Nichtterminalsymbol = syntaktische Variable

$\odot a$... 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

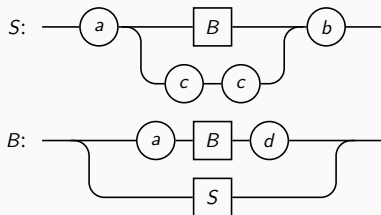
Hauptaugenmerk:

Protokollierung von Wortentstehung & Markenkeller

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

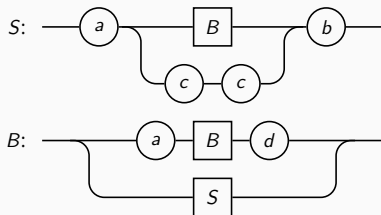
AUFGABE 1

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



AUFGABE 1

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



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

- ▶ $a accb b$
- ▶ $a a accb b b$
- ▶ $a a accb d b$
- ▶ $a a a accb d d b$
- ▶ $a a a accb b d b$

Wort: aaaaccbdbb

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: aaaaccbdbb

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

Protokollierungszeitpunkte:

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

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	\mathcal{Z} 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
- ▶ \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

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

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

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

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

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

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}

AUFGABE 1 — TEIL (B)

Wort: aaaaccbdbb

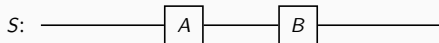
Protokollierungszeitpunkte:

- ▶ 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	-

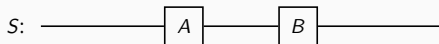
$$L = L_A \cdot L_B$$

$$L = L_A \cdot L_B$$



GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

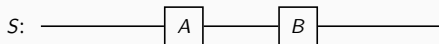
$$L = L_A \cdot L_B$$



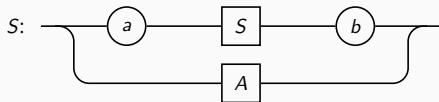
$$L = \{a^n L_A b^n : n \geq 0\}$$

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$$L = L_A \cdot L_B$$

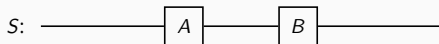


$$L = \{a^n L_A b^n : n \geq 0\}$$

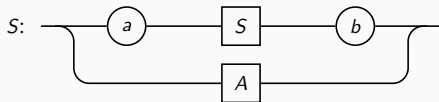


GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$$L = L_A \cdot L_B$$



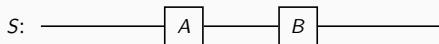
$$L = \{a^n L_A b^n : n \geq 0\}$$



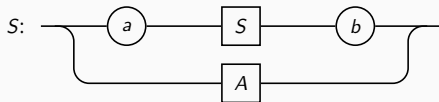
$$L = \{a^n L_A b^n : n > 0\}$$

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

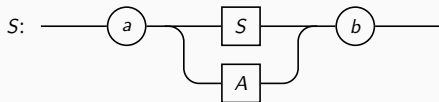
$$L = L_A \cdot L_B$$



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

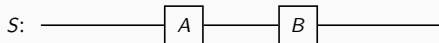


$$L = \{a^n L_A b^n : n > 0\}$$

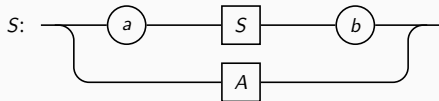


GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

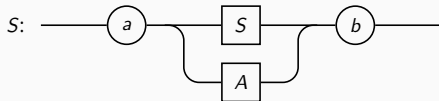
$$L = L_A \cdot L_B$$



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



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



kleine Tricks:

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

AUFGABE 1 — TEIL (C)

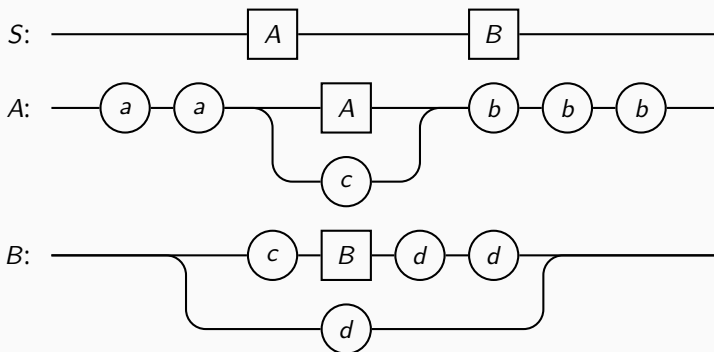
$$L = \left\{ a^{2i} c b^{3i} c^k d^{2k+1} \mid i > 0, k \geq 0 \right\}$$

AUFGABE 1 — TEIL (C)

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

AUFGABE 1 — TEIL (C)

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



Extended Backus-Naur-Form

EBNF-DEFINITION

- ▶ EBNF-Definition besteht aus endlicher Menge von EBNF-Regeln.

$$\mathcal{E} = (V, \Sigma, S, R)$$

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

$$left ::= EBNFTerm$$

EBNF-DEFINITION

- ▶ EBNF-Definition besteht aus endlicher Menge von EBNF-Regeln.

$$\mathcal{E} = (V, \Sigma, S, R)$$

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

$$left ::= EBNFTerm$$

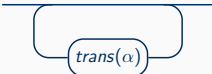
Definition (EBNF-Terme): Seien V (syntaktische Variablen) und Σ (Terminalsymbole) endliche Mengen mit $V \cap \Sigma = \emptyset$. Die Menge der EBNF-Terme über V und Σ (notiere: $T(\Sigma, V)$), ist die *kleinste* Menge $T \subseteq \left(V \cup \Sigma \cup \left\{ \hat{\{ \}}, \hat{[\]}, \hat{(\)},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

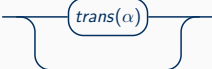
- ▶ Wenn $\alpha \in T$, so auch $\hat{(\alpha)} \in T, \hat{\{ \alpha \}} \in T, \hat{[\alpha]} \in T$.
- ▶ Wenn $\alpha_1, \alpha_2 \in T$, so auch $\hat{(\alpha_1 \hat{[\alpha_2]})} \in T, \alpha_1 \alpha_2 \in T$.

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.

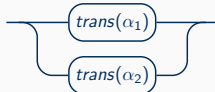
► $trans(\hat{\{\alpha\}}) =$ 

► $trans(\hat{[\alpha]}) =$ 

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

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

► $trans(\alpha_1 \alpha_2) =$ 

► $trans(\hat{(\alpha_1 \mid \alpha_2)}) =$ 

AUFGABE 2 — TEIL (A)

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

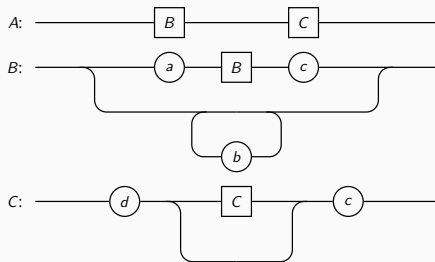
$$\begin{aligned} V = \{A, B, C\} \quad \text{und} \quad R = \Big\{ & A ::= BC, \\ & B ::= \hat{a} B c \mid \hat{\{ b \}}, \\ & C ::= d \hat{[C]} c \quad \Big\} \end{aligned}$$

AUFGABE 2 — TEIL (A)

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

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

Übersetzung in ein Syntaxdiagrammsystem:



Startdiagramm: A

AUFGABE 2 — TEIL (B)

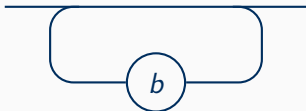
Wir wollen die von \mathcal{E} beschriebene Sprache L_A beschreiben und wenden dafür die Grundkonstruktionen „rückwärts“ an.

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 aus $\{b\}^*$ schreiben. Diese Sprache $\{b\}^*$ wird durch



beschrieben.

AUFGABE 2 — TEIL (C)

Gegeben sei die Sprache

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

Gesucht ist eine zugehörige EBNF-Definition.

AUFGABE 2 — TEIL (C)

Gegeben sei die Sprache

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

Gesucht ist eine zugehörige EBNF-Definition.

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

AUFGABE 2 — TEIL (C)

Gegeben sei die Sprache

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

Gesucht ist eine zugehörige EBNF-Definition.

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

Lösungsweg: via Syntaxdiagrammsystem & Übersetzung

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

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