TECHNISCHE UNIVERSITÄT DRESDEN

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 2: SYNTAXDIAGRAMME & EBNF

Eric Kunze

eric.kunze@tu-dresden.de

TU Dresden, 06.11.2020

VIDEOEMPFEHLUNG

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.

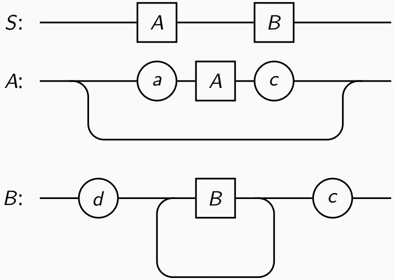
Die Einleitung entspricht ungefähr dem Inhalt der ersten Übung:

- `https://youtu.be/Lma6jaPnD-I`

Syntaxdiagramme

SYNTAXDIAGRAMME

Beispiel eines Syntaxdiagrammsystems mit Startdiagramm S:



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

ⓐ . . . Terminalsymbol

RÜCKSPRUNGALGORITHMUS

Rücksprungalgorithmus

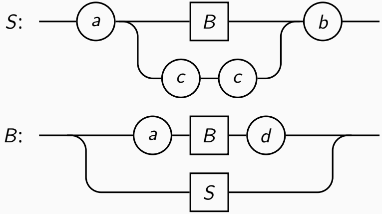
- Ziel: Nachweis von Zugehörigkeit eines Wortes zu einer Sprache
- jedes Kästchen bekommt eindeutige Marke (Rücksprungadresse)
- beim Betreten eines Syntaxdiagramms wird eine Marke auf den Keller gelegt

Hauptaugenmerk:
Protokollierung von Wortentstehung & Markenkeller

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

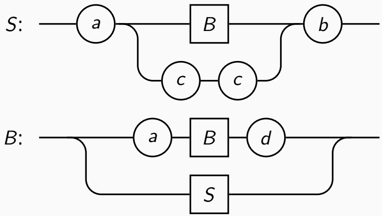
AUFGABE 1

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

AUFGABE 1 — TEIL (B)

Wort: `aaaaccbdbb`

Protokollierungszeitpunkte:

- jeder Aufenthalt in einem Syntaxdiagramm entspricht einer Zeile
- jede Zeile führt eine Operation auf dem Markenkeller aus
- $\cancel{3}$ = 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
- $\cancel{3}$ = 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
- $\cancel{3}$ = 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
- $\cancel{3}$ = 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
- $\cancel{3}$ = 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
- $\cancel{3}$ = 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
- $\cancel{3}$ = 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
- $\not 3$ = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	$\not 3$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
- $\cancel{3}$ = Rücksprung zu Marke 3

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	$\cancel{3}2131$
aaaaccb	$\cancel{2}131$

AUFGABE 1 — TEIL (B)

Wort: `aaaaccbdbb`

Protokollierungszeitpunkte:

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

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	$\cancel{3}2131$
aaaaccb	$\cancel{2}131$
aaaaccbd	$\cancel{1}31$

AUFGABE 1 — TEIL (B)

Wort: `aaaaccbdbb`

Protokollierungszeitpunkte:

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

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	$\cancel{3}2131$
aaaaccb	$\cancel{2}131$
aaaaccbd	$\cancel{1}31$
aaaaccbdb	$\cancel{3}1$

AUFGABE 1 — TEIL (B)

Wort: `aaaaccbdbb`

Protokollierungszeitpunkte:

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

Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	$\cancel{3}2131$
aaaaccb	$\cancel{2}131$
aaaaccbd	$\cancel{1}31$
aaaaccbdb	$\cancel{3}1$
aaaaccbdb	$\cancel{1}$

AUFGABE 1 — TEIL (B)

Wort: `aaaaccbdbb`

Protokollierungszeitpunkte:

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

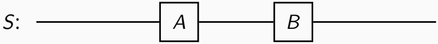
Wort	Markenkeller
a	1
a	31
aa	131
aaa	2131
aaa	32131
aaaaccb	$\cancel{3}2131$
aaaaccb	$\cancel{2}131$
aaaaccbd	$\cancel{1}31$
aaaaccbdb	$\cancel{3}1$
aaaaccbdb	$\cancel{1}$
aaaaccbdbb	–

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

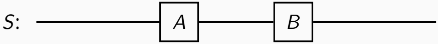
$L = L_A \cdot L_B$

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$L = L_A \cdot L_B$



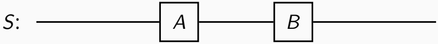
$L = L_A \cdot L_B$



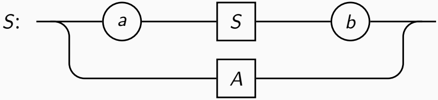
$L = \{a^n \; L_A \; b^n : n \geq 0\}$

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$L = L_A \cdot L_B$

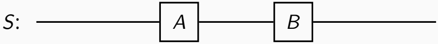


$L = \{a^n \, L_A \, b^n : n \geq 0\}$

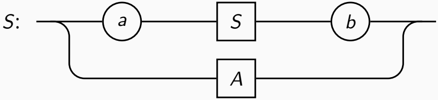


GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

$L = L_A \cdot L_B$



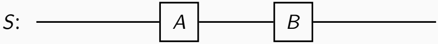
$L = \{a^n\ L_A\ b^n : n \geq 0\}$



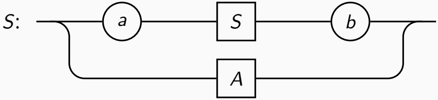
$L = \{a^n\ L_A\ b^n : n > 0\}$

GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

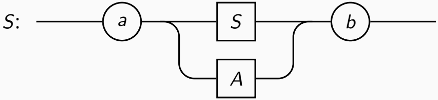
$L = L_A \cdot L_B$



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

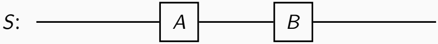


$L = \{a^n \; L_A \; b^n : n > 0\}$

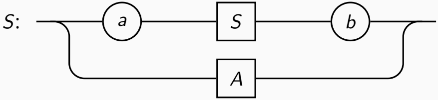


GRUNDKONSTRUKTION VON SYNTAXDIAGRAMMEN

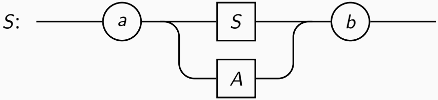
$L = L_A \cdot L_B$



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



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



kleine Tricks:

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

AUFGABE 1 — TEIL (C)

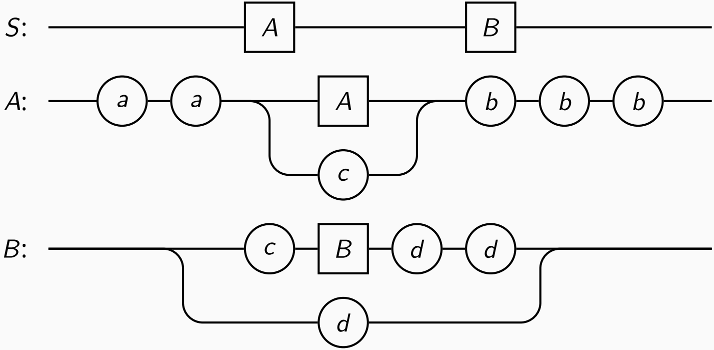
$$L = \left\{ a^{2i} c b^{3i} c^k d^{2k+1} \mid i > 0, k \geq 0 \right\}$$

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)

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



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.

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

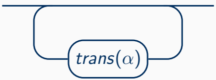
- Wenn $\alpha \in T$, so auch $\hat{(}\ \alpha\ \hat{)} \in T$, $\hat{\{}\ \alpha\ \hat{\}} \in T$, $\hat{[}\ \alpha\ \hat{]} \in T$.
- Wenn $\alpha_1, \alpha_2 \in T$, so auch $\hat{(}\ \alpha_1\ \hat{|}\ \alpha_2\ \hat{)} \in T$, $\alpha_1\alpha_2 \in T$

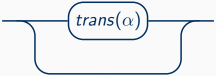
ÜBERSETZUNG EBNF $\leftrightarrow$ SYNTAXDIAGRAMME

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

; $\mathit{trans}(w) =$ 

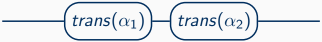
Sei $\alpha \in T(\Sigma, V)$ ein EBNF-Term.

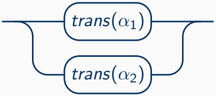
- $\mathit{trans}(\ \hat{\{}\ \alpha\ \hat{\}}\ ) =$ 

- $\mathit{trans}(\ \hat{[}\ \alpha\ \hat{]}\ ) =$ 

- $\mathit{trans}(\ \hat{(}\ \alpha\ \hat{)}\ ) = \mathit{trans}(\alpha)$

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

- $\mathit{trans}(\ \alpha_1\alpha_2\ ) =$ 

- $\mathit{trans}(\ \hat{(}\ \alpha_1\ \hat{|}\ \alpha_2\ \hat{)}\ ) =$ 

Auszug aus dem Foliensatz der Vorlesung

AUFGABE 2 — TEIL (A)

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

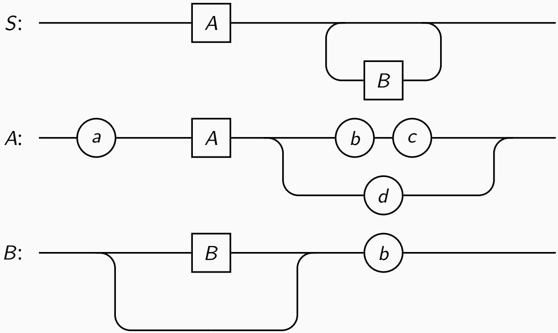
$$V = \{S, A, B\} \quad \text{und} \quad R = \left\{ \begin{aligned} &S ::= A \,\hat{\{}\, B \,\hat{\}}, \\ &A ::= aA \,\hat{(}\, bc \,\hat{|}\, d \,\hat{)}, \\ &B ::= \hat{[}\, B \,\hat{]}\, b \end{aligned} \right\}$$

AUFGABE 2 — TEIL (A)

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

$$V = \{S, A, B\} \quad \text{und} \quad R = \left\{ \begin{array}{l} S ::= A \,\hat{\{}\, B \,\hat{\}}, \\ A ::= aA \,\hat{(}\, bc \,\hat{|}\, d \,\hat{)}, \\ B ::= \hat{[}\, B \,\hat{]}\, b \end{array} \right\}$$

Übersetzung in ein Syntaxdiagrammsystem:



AUFGABE 2 — 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.

AUFGABE 2 — 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.

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

AUFGABE 2 — 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 $\Sigma = \{a, b, c, d\}$,

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