

Formale Sprachen

Definition:

- Typ 0:** keine Bedingung, z.B. Linke Seite länger als rechte allgemeine Grammatik Wenn Startsymbol $S \rightarrow \epsilon$
- Typ 1:** $| \alpha | \leq | \beta |$, z.B. Links \leq rechts nichtverarbeitende Grammatik, $aBBA \Rightarrow abba$
 $aBa \Rightarrow abba$
- Typ 2:** Links nur Nichtterminale Strukturbäume (Bsp. nächste Seite)
kontextfreie Grammatik, kontextfrei, $A \rightarrow aBC, C \rightarrow aCC$
- Typ 3:** Links nur Nichtterminale, rechts nur ein Nichtterminal
(rechts-)lineare Grammatik, regular, $A \rightarrow aA, A \rightarrow abA, A \rightarrow a$

$$G = (N, T, S, P)$$

N \rightarrow Nichtterminale
T \rightarrow Terminate
S \rightarrow Startsymbol
P \rightarrow Produktion $\alpha \rightarrow \beta$

Sonderregel leeres Wort X 8
Snew $\rightarrow \epsilon$ | Salt wird erlaubt
Snew wird zum Startzustand

Normalformen	immer eindeutig			
	Type-3	Type-2	Type-1	Type-0
$A \rightarrow e$				X
$A \rightarrow t$	X	X	X	X
$A \rightarrow tB$	X			X
$A \rightarrow BC$		X	X	X
$AB \rightarrow CD$		X	X	X

Endaus - Normalform
Startsymbol: < S >
 $P = \{ < S > ::= a < S > | b < S > | < X >, < X > ::= aaa | bbb \}$

Linkslinear: $aEE \Rightarrow aBE$
rechtslinear: $aEE \Rightarrow aEB$
links- & rechtslinear: keine Linearität
Kettenregel: $A \rightarrow B$
mehrdeutige Terminalketten: $A \rightarrow xyza$
 $b \rightarrow ab$

Normalisierung Typ-2

$$G = ([A, C], [a, b, c], A, P), P = [A \rightarrow aA, A \rightarrow ab, A \rightarrow C, C \rightarrow c]$$

- ① Separation von Terminalen
 $P' = [T_a = a, T_b = b, T_c = c, A = AaA, A = Abab, A = C, C = c]$
- ② Elimination von Wörterketten längter als 2
 $P'' = [T_a = a, T_b = b, T_c = c, A = ANa, N_a = T_b A, A = T_b N_a, N_a = T_b T_c, A = C, C = c]$
- ③ Elimination von Kettenregeln
 $P''' = [T_a = a, T_b = b, T_c = c, A = ANa, N_a = T_b A, A = T_b N_a, N_a = T_b T_c, A = C, C = c]$

Normalisierung Typ-3

$$x = [a, b], L = [(ab)^n b^m | n, m \in \mathbb{N}], G = ([S, T], [a, b], S, P), P = [S \rightarrow ab, T \rightarrow bT | b]$$

- ① Elimination von mehrdeutigen Terminalketten

$$S \rightarrow abS \quad S \rightarrow A \quad A \rightarrow bS$$

$$S \rightarrow abT \quad S \rightarrow b \quad b \rightarrow bT$$

$$P_{nr} = [S \rightarrow aA, A \rightarrow bS, S \rightarrow b, b \rightarrow bT, T \rightarrow bT | b]$$

- ② Elimination der Kettenregeln nicht verwenden in diesem Beispiel

- ③ Gnr = ([S, T], A, B), [a, b], S, P_{nr})

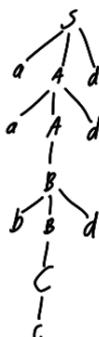
Akkordieren verhindern alle Terminalen die alleine stehen bekommen ein N2

$$P_{nr} = [S \rightarrow aA \quad A \rightarrow bS \quad S \rightarrow b \quad b \rightarrow bT \quad T \rightarrow bT | b \quad N_a = a, N_b = b]$$

Kommas hinzusetzen

Beispiel Strukturbäum:

a) $G_1 = (\{A, B, C, D\}, \{ab, bc, cd, A, B, P_1\})$
 $P_1 = \{ S \rightarrow aAd | bbd, A \rightarrow aAb | bB, B \rightarrow bBa | C, C \rightarrow cCc \}$



b) aabcd

Doppelte Striche!!!

$$\Rightarrow aAd \Rightarrow aaAdd \Rightarrow aabBdd \Rightarrow aabBddd \Rightarrow aabBddd \Rightarrow aabddd$$

Aufgabe 1:

Sei $L = \{(abc)^n d^m \mid n \in \mathbb{N}, m \in \mathbb{N}_0\}$

- Geben sie eine Typ-3 Grammatik an, die L erzeugt.
- Geben sie auf Basis der Grammatik von a) eine Ableitung des Wortes $abcabcd$ an.
- Normalisieren sie die Grammatik von a).
- Konstruieren sie den zugehörigen endlichen Automaten.

Aufgabe 2:

Sei $L = \{(ab)^n (cd)^m \mid n, m \in \mathbb{N}\}$

- Geben sie eine Typ-3 Grammatik an, die L erzeugt.
- Geben sie auf Basis der Grammatik von a) eine Ableitung des Wortes $abcdcdcd$ an.
- Normalisieren sie die Grammatik von a).
- Konstruieren sie den zugehörigen endlichen Automaten.

Aufgabe 3:

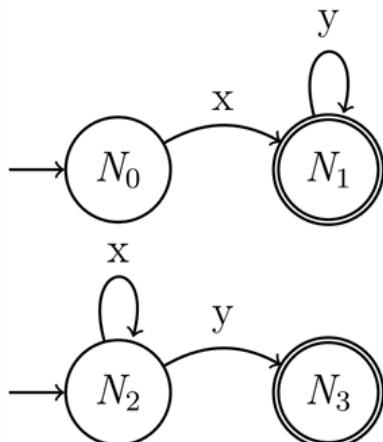
Sei $R = ((ba)^* \cup c)d^*$ und L die von R erzeugte Sprache.

- Geben sie eine Typ-3 Grammatik an, die L erzeugt.
- Normalisieren sie die Grammatik von a).

Aufgabe 4:

Geben sie die zugehörige Typ-3 Grammatiken der folgenden Automaten an.

a) A1



b) A2

