

# Automatentheorie und Formale Sprachen

für die Studiengänge

- Angewandte Informatik
- Informatik - Technische Systeme

## 02 Chomsky-Grammatiken

Prof. Dr. David Sabel  
Sommersemester 2025

Stand der Folien: 20. Mai 2025

- Sei  $\Sigma$  ein Alphabet.
- Eine **Sprache über  $\Sigma$**  ist eine Teilmenge von  $\Sigma^*$ .
- Z.B. für  $\Sigma = \{ (, ), +, -, *, /, a \}$  sei  
 $L_{ArEx}$  die Sprache aller korrekt geklammerten Ausdrücke  
  
Z.B.  $((a + a) - a) * a \in L_{ArEx}$  aber  $(a - ) + a \notin L_{ArEx}$
- Unsere bisherigen Operationen auf Sprachen (Mengen) können das nicht darstellen

**Benötigt:** Formalismus, um  $L_{ArEx}$  zu beschreiben

# Formale Sprachen darstellen (2)

Anforderungen:

- **Endliche** Beschreibung
- Sprache selbst muss aber auch unendlich viele Objekte erlauben

Zwei wesentliche solchen Formalismen sind

- Grammatiken
- Automaten

Grammatik für einen sehr kleinen Teil der deutschen Sprache:

<Satz> → <Subjekt><Prädikat><Objekt>

<Subjekt> → <Artikel><Attribut><Nomen>

<Objekt> → <Artikel><Attribut><Nomen>

<Artikel> →  $\epsilon$

<Artikel> → der

<Artikel> → das

<Attribut> → <Adjektiv>

<Attribut> → <Adjektiv><Attribut>

<Adjektiv> → kleine

<Adjektiv> → große

<Adjektiv> → nette

<Adjektiv> → blaue

<Nomen> → Mann

<Nomen> → Auto

<Prädikat> → fährt

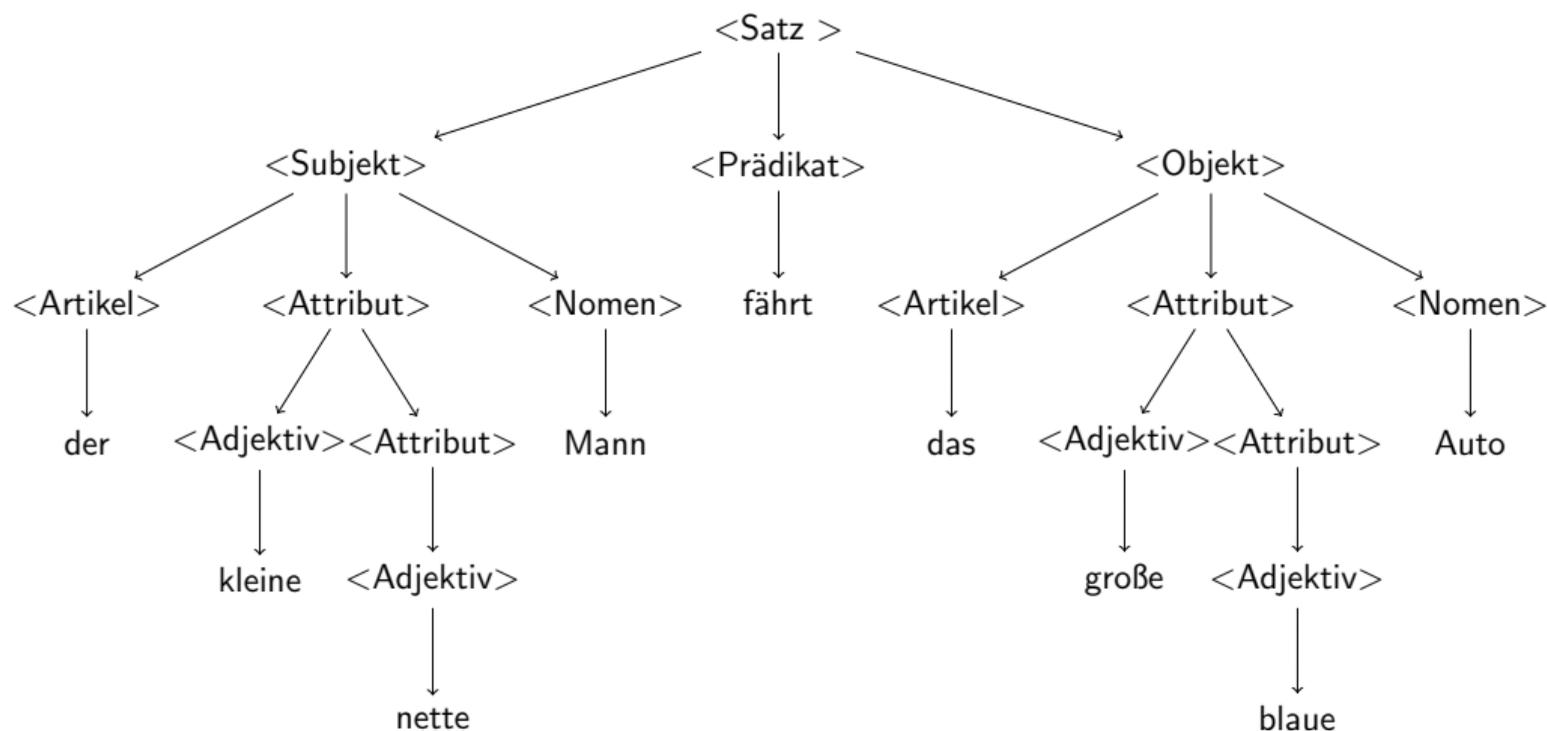
<Prädikat> → liebt

- Endliche Menge von Regeln „linke Seite  $\rightarrow$  rechte Seite“
- Symbole in spitzen Klammern wie  $\langle$ Artikel $\rangle$  sind **Variablen**, d.h. sie sind **Platzhalter**, die weiter **ersetzt** werden müssen.
- Ableiten: Ersetze Vorkommen von linken Seiten durch rechte Seiten
- Z.B. kann

„der kleine nette Mann fährt das große blaue Auto“

durch die obige Grammatik abgeleitet werden

# Syntaxbaum zum Beispiel



## Definition (Grammatik)

Eine **Grammatik** ist ein 4-Tupel  $G = (V, \Sigma, P, S)$  mit

- $V$  ist eine endliche Menge von **Variablen**  
(alternativ **Nichtterminale**, Nichtterminalsymbole)
- $\Sigma$  (mit  $V \cap \Sigma = \emptyset$ ) ist ein **Alphabet** von **Zeichen**  
(alternativ **Terminale**, Terminalsymbole)
- $P$  ist eine endliche Menge von **Produktionen** von der Form  $\ell \rightarrow r$  wobei  $\ell \in (V \cup \Sigma)^+$  und  $r \in (V \cup \Sigma)^*$   
(alternativ **Regeln**)
- $S \in V$  ist das **Startsymbol**  
(alternativ **Startvariable**)

Manchmal notieren wir nur  $P$  (wenn klar ist, was Variablen, Zeichen, Startsymbol sind)

# Beispiel für eine Grammatik

$G = (V, \Sigma, P, E)$  mit

$$V = \{E, M, Z\},$$

$$\Sigma = \{+, *, 1, 2, (, )\}$$

$$P = \{E \rightarrow M, \\ E \rightarrow E + M, \\ M \rightarrow Z, \\ M \rightarrow M * Z, \\ Z \rightarrow 1, \\ Z \rightarrow 2, \\ Z \rightarrow (E)\}$$

Sei  $G = (V, \Sigma, P, S)$  eine Grammatik.

## Ableitungsschritt $\Rightarrow_G$

Für Satzformen  $u, v$  (d.h. Wörter aus  $(V \cup \Sigma)^*$ ) sagen wir:

$u$  geht unter Grammatik  $G$  unmittelbar in  $v$  über,  $u \Rightarrow_G v$ , wenn

$$u = w_1 \ell w_2 \text{ und } w_1 r w_2 = v \text{ mit } (\ell \rightarrow r) \in P$$

- Wenn  $G$  klar ist, schreiben wir  $u \Rightarrow v$  statt  $u \Rightarrow_G v$
- $\Rightarrow_G^*$  sei die reflexiv-transitive Hülle von  $\Rightarrow_G$

## Ableitung

Eine Folge  $(w_0, w_1, \dots, w_n)$  mit  $w_0 = S$ ,  $w_n \in \Sigma^*$  und  $w_{i-1} \Rightarrow w_i$  für  $i = 1, \dots, n$  heißt **Ableitung von  $w_n$** . Statt  $(w_0, \dots, w_n)$  schreiben wir auch  $w_0 \Rightarrow \dots \Rightarrow w_n$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$E \Rightarrow M$$

# Beispiel

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$E \Rightarrow M \Rightarrow M * Z$$

# Beispiel

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z$$

# Beispiel

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E)$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M)$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2) \\ &\Rightarrow (M + Z) * (Z + 2) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2) \\ &\Rightarrow (M + Z) * (Z + 2) \Rightarrow (M + Z) * (2 + 2) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2) \\ &\Rightarrow (M + Z) * (Z + 2) \Rightarrow (M + Z) * (2 + 2) \\ &\Rightarrow (Z + Z) * (2 + 2) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{l} E \rightarrow M, \quad E \rightarrow E + M, \quad M \rightarrow Z, \quad M \rightarrow M * Z, \\ Z \rightarrow 1, \quad Z \rightarrow 2, \quad Z \rightarrow (E) \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2) \\ &\Rightarrow (M + Z) * (Z + 2) \Rightarrow (M + Z) * (2 + 2) \\ &\Rightarrow (Z + Z) * (2 + 2) \Rightarrow (2 + Z) * (2 + 2) \end{aligned}$$

$G = (V, \Sigma, P, E)$  mit  $V = \{E, M, Z\}$  und  $\Sigma = \{+, *, 1, 2, (, )\}$  und

$$P = \left\{ \begin{array}{llll} E \rightarrow M, & E \rightarrow E + M, & M \rightarrow Z, & M \rightarrow M * Z, \\ Z \rightarrow 1, & Z \rightarrow 2, & Z \rightarrow (E) & \end{array} \right\}$$

Eine Ableitung von  $(2+1) * (2+2)$ :

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2) \\ &\Rightarrow (M + Z) * (Z + 2) \Rightarrow (M + Z) * (2 + 2) \\ &\Rightarrow (Z + Z) * (2 + 2) \Rightarrow (2 + Z) * (2 + 2) \\ &\Rightarrow (2 + 1) * (2 + 2) \end{aligned}$$

## Beispiel: Ableitungen sind nicht eindeutig

Ableitung von letzter Folie (keine Linksableitung):

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\ &\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\ &\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z) \\ &\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2) \\ &\Rightarrow (M + Z) * (Z + 2) \Rightarrow (M + Z) * (2 + 2) \\ &\Rightarrow (Z + Z) * (2 + 2) \Rightarrow (2 + Z) * (2 + 2) \\ &\Rightarrow (2 + 1) * (2 + 2) \end{aligned}$$

Linksableitung: ersetzt immer das linkeste Nichtterminal

$$\begin{aligned} E &\Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow (E) * Z \\ &\Rightarrow (E + M) * Z \Rightarrow (M + M) * Z \Rightarrow (Z + M) * Z \\ &\Rightarrow (2 + M) * Z \Rightarrow (2 + Z) * Z \Rightarrow (2 + 1) * Z \Rightarrow (2 + 1) * (E) \\ &\Rightarrow (2 + 1) * (E + M) \Rightarrow (2 + 1) * (M + M) \Rightarrow (2 + 1) * (Z + M) \\ &\Rightarrow (2 + 1) * (2 + M) \Rightarrow (2 + 1) * (2 + Z) \\ &\Rightarrow (2 + 1) * (2 + 2) \end{aligned}$$

# Quiz 1

Sei  $G = (\{A, B, C\}, \{d, e\}, P, A)$  mit

$P = \{A \rightarrow BBCC, B \rightarrow d, C \rightarrow e, dC \rightarrow Cd, ed \rightarrow \varepsilon\}$

Welches Wort liegt in  $L(G)$ ?

- a)  $\varepsilon$
- b)  $BBdd$
- c)  $e$

arsnova.hs-rm.de  
6750 1376



Für eine Satzform  $u$  kann es verschiedene Satzformen  $v_i$  geben mit  $u \Rightarrow_G v_i$ .

Quellen des Nichtdeterminismus:

- Wähle, **welche Produktion**  $\ell \rightarrow r$  aus  $P$  angewendet wird
- Wähle die **Position des Teilworts**  $\ell$  in  $u$ , das durch  $r$  ersetzt wird.

Aber: Es gibt **nur endliche viele**  $v_i$  für jeden Schritt!

## Erzeugte Sprache einer Grammatik

Die von einer Grammatik  $G = (V, \Sigma, P, S)$  **erzeugte Sprache**  $L(G)$  ist

$$L(G) := \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}.$$

## Quiz 2

Sei  $G_1 = (\{S\}, \{a\}, \{S \rightarrow aS\}, S)$  und  
 $G_2 = (\{S'\}, \{a, b\}, \{S' \rightarrow aS', S' \rightarrow b\}, S')$ .

Welche Sprache erzeugt  $G_1$ ?

Welche Sprache erzeugt  $G_2$ ?

arsnova.hs-rm.de

6750 1376



$$G_1 = (\{S\}, \{a\}, \{S \rightarrow aS\}, S)$$

$$G_2 = (\{S'\}, \{a, b\}, \{S' \rightarrow aS', S' \rightarrow b\}, S')$$

$$G_1 = (\{S\}, \{a\}, \{S \rightarrow aS\}, S)$$

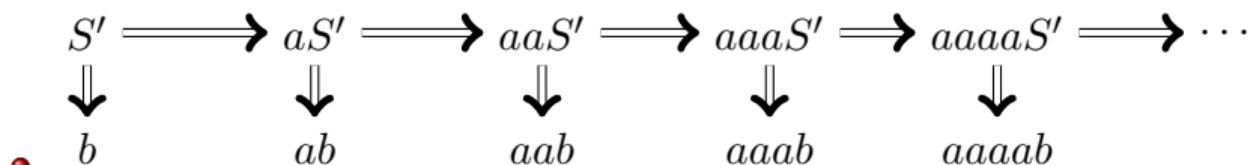
- $S \Rightarrow aS \Rightarrow aaS \Rightarrow \dots$  endet nie
- Andere Ableitungen gibt es nicht
- Daher sind keine Wörter aus  $\{a\}^*$  ableitbar

$$G_2 = (\{S'\}, \{a, b\}, \{S' \rightarrow aS', S' \rightarrow b\}, S')$$

$$G_1 = (\{S\}, \{a\}, \{S \rightarrow aS\}, S)$$

- $S \Rightarrow aS \Rightarrow aaS \Rightarrow \dots$  endet nie
- Andere Ableitungen gibt es nicht
- Daher sind keine Wörter aus  $\{a\}^*$  ableitbar

$$G_2 = (\{S'\}, \{a, b\}, \{S' \rightarrow aS', S' \rightarrow b\}, S')$$



- Für alle  $i \in \mathbb{N}_0$  gilt  $S \Rightarrow^i a^i S \Rightarrow a^i b$

# CHOMSKY-HIERARCHIE

- Grammatiken der Typen 0,1,2,3
- Typ-i Sprachen

# Die Chomsky-Hierarchie

Noam Chomsky teilte die Grammatiken in Typen 0 bis 3:

Sei  $G = (V, \Sigma, P, S)$  eine Grammatik.

# Die Chomsky-Hierarchie

Noam Chomsky teilte die Grammatiken in Typen 0 bis 3:

Sei  $G = (V, \Sigma, P, S)$  eine Grammatik.

## $G$ ist vom Typ 0

$G$  ist automatisch vom Typ 0 (alles erlaubt)

## $G$ ist vom Typ 1 (kontextsensitive Grammatik), wenn ...

für alle  $(\ell \rightarrow r) \in P$ :  $|\ell| \leq |r|$  (keine verkürzende Produktionen)

## $G$ ist vom Typ 2 (kontextfreie Grammatik), wenn ...

$G$  ist vom Typ 1 und für alle  $(\ell \rightarrow r) \in P$  gilt:  $\ell = A \in V$  (linke Seite = 1 Variable)

## $G$ ist vom Typ 3 (reguläre Grammatik), wenn ...

$G$  ist vom Typ 2 und für alle  $(A \rightarrow r) \in P$  gilt:  $r = a$  oder  $r = aA'$  für  $a \in \Sigma, A' \in V$   
(rechte Seite = Wort aus  $(\Sigma \cup (\Sigma V))$ )

# Die Chomsky-Hierarchie

Noam Chomsky teilte die Grammatiken in Typen 0 bis 3:

Sei  $G = (V, \Sigma, P, S)$  eine Grammatik.

## $G$ ist vom Typ 0

$G$  ist automatisch vom Typ 0 (alles erlaubt)

## $G$ ist vom Typ 1 (kontextsensitive Grammatik), wenn ...

für alle  $(\ell \rightarrow r) \in P$ :  $|\ell| \leq |r|$  (keine verkürzende Produktionen)

## $G$ ist vom Typ 2 (kontextfreie Grammatik), wenn ...

$G$  ist vom Typ 1 und für alle  $(\ell \rightarrow r) \in P$  gilt:  $\ell = A \in V$  (linke Seite = 1 Variable)

## $G$ ist vom Typ 3 (reguläre Grammatik), wenn ...

$G$  ist vom Typ 2 und für alle  $(A \rightarrow r) \in P$  gilt:  $r = a$  oder  $r = aA'$  für  $a \in \Sigma, A' \in V$   
(rechte Seite = Wort aus  $(\Sigma \cup (\Sigma V))$ )

# Die Chomsky-Hierarchie

Noam Chomsky teilte die Grammatiken in Typen 0 bis 3:

Sei  $G = (V, \Sigma, P, S)$  eine Grammatik.

## $G$ ist vom Typ 0

$G$  ist automatisch vom Typ 0 (alles erlaubt)

## $G$ ist vom Typ 1 (kontextsensitive Grammatik), wenn ...

für alle  $(\ell \rightarrow r) \in P$ :  $|\ell| \leq |r|$  (keine verkürzende Produktionen)

## $G$ ist vom Typ 2 (kontextfreie Grammatik), wenn ...

$G$  ist vom Typ 1 und für alle  $(\ell \rightarrow r) \in P$  gilt:  $\ell = A \in V$  (linke Seite = 1 Variable)

## $G$ ist vom Typ 3 (reguläre Grammatik), wenn ...

$G$  ist vom Typ 2 und für alle  $(A \rightarrow r) \in P$  gilt:  $r = a$  oder  $r = aA'$  für  $a \in \Sigma, A' \in V$   
(rechte Seite = Wort aus  $(\Sigma \cup (\Sigma V))$ )

# Die Chomsky-Hierarchie

Noam Chomsky teilte die Grammatiken in Typen 0 bis 3:

Sei  $G = (V, \Sigma, P, S)$  eine Grammatik.

## $G$ ist vom Typ 0

$G$  ist automatisch vom Typ 0 (alles erlaubt)

## $G$ ist vom Typ 1 (kontextsensitive Grammatik), wenn ...

für alle  $(\ell \rightarrow r) \in P$ :  $|\ell| \leq |r|$  (keine verkürzende Produktionen)

## $G$ ist vom Typ 2 (kontextfreie Grammatik), wenn ...

$G$  ist vom Typ 1 und für alle  $(\ell \rightarrow r) \in P$  gilt:  $\ell = A \in V$  (linke Seite = 1 Variable)

## $G$ ist vom Typ 3 (reguläre Grammatik), wenn ...

$G$  ist vom Typ 2 und für alle  $(A \rightarrow r) \in P$  gilt:  $r = a$  oder  $r = aA'$  für  $a \in \Sigma, A' \in V$   
(rechte Seite = Wort aus  $(\Sigma \cup (\Sigma V))$ )

## Definition

Für  $i = 0, 1, 2, 3$  nennt man eine formale **Sprache**  $L \subseteq \Sigma^*$  **vom Typ  $i$** , falls es **eine Typ  $i$ -Grammatik  $G$  gibt**, sodass  $L(G) = L$  gilt.

Spricht man von **dem Typ einer formalen Sprache**, so ist stets der größtmögliche Typ gemeint.

Bemerkung: Die Definition erlaubt Aussagen der Form:

*Typ  $i + k$ -Sprachen sind eine Teilmenge der Typ  $i$ -Sprachen, da jede Typ  $i + k$ -Grammatik auch eine Typ  $i$ -Grammatik ist.*

- $G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow b\}, S)$  ist regulär (Typ 3)
- $G_2 = (\{E, M, Z\}, \{+, *, 1, 2, (, )\}, P, E)$  mit  
 $P = \{E \rightarrow M, E \rightarrow E + M, M \rightarrow Z, M \rightarrow M * Z,$   
 $Z \rightarrow 1, Z \rightarrow 2, Z \rightarrow (E)\}$  ist kontextfrei (Typ 2)
- $G_3 = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab,$   
 $bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$  ist kontextsensitiv (Typ 1)
- $G_4 = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit  
 $P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$,$   
 $\$b \rightarrow b\$, Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB,$   
 $A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$  ist vom Typ 0

- $G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow b\}, S)$  ist regulär (Typ 3)
- $G_2 = (\{E, M, Z\}, \{+, *, 1, 2, (, )\}, P, E)$  mit  
 $P = \{E \rightarrow M, E \rightarrow E + M, M \rightarrow Z, M \rightarrow M * Z,$   
 $Z \rightarrow 1, Z \rightarrow 2, Z \rightarrow (E)\}$  ist kontextfrei (Typ 2)
- $G_3 = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab,$   
 $bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$  ist kontextsensitiv (Typ 1)
- $G_4 = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit  
 $P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$,$   
 $\$b \rightarrow b\$, Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB,$   
 $A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$  ist vom Typ 0

- $G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow b\}, S)$  ist regulär (Typ 3)
- $G_2 = (\{E, M, Z\}, \{+, *, 1, 2, (, )\}, P, E)$  mit  
 $P = \{E \rightarrow M, E \rightarrow E + M, M \rightarrow Z, M \rightarrow M * Z,$   
 $Z \rightarrow 1, Z \rightarrow 2, Z \rightarrow (E)\}$  ist kontextfrei (Typ 2)
- $G_3 = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab,$   
 $bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$  ist kontextsensitiv (Typ 1)
- $G_4 = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit  
 $P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$,$   
 $\$b \rightarrow b\$, Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB,$   
 $A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$  ist vom Typ 0

- $G_1 = (\{S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow b\}, S)$  ist regulär (Typ 3)
- $G_2 = (\{E, M, Z\}, \{+, *, 1, 2, (, )\}, P, E)$  mit  
 $P = \{E \rightarrow M, E \rightarrow E + M, M \rightarrow Z, M \rightarrow M * Z,$   
 $Z \rightarrow 1, Z \rightarrow 2, Z \rightarrow (E)\}$  ist kontextfrei (Typ 2)
- $G_3 = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab,$   
 $bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$  ist kontextsensitiv (Typ 1)
- $G_4 = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit  
 $P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$,$   
 $\$b \rightarrow b\$, Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB,$   
 $A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$  ist vom Typ 0

## Quiz 3

Sei  $G = (\{A, B, C\}, \{d, e\}, P, A)$  mit

$$P = \{A \rightarrow BBCC, B \rightarrow d, C \rightarrow e, dC \rightarrow Cd\}$$

Welchen Typ hat  $G$ ?

- a) 0
- b) 1
- c) 2
- d) 3

arsnova.hs-rm.de

6750 1376



## Quiz 4

Sei  $G = (\{A, B, C\}, \{d, e\}, P, A)$  mit

$$P = \{A \rightarrow BBCC, B \rightarrow d, C \rightarrow e, dC \rightarrow Cd\}$$

Welchen Typ hat  $L(G)$ ?

- a) 0
- b) 1
- c) 2
- d) 3

arsnova.hs-rm.de

6750 1376



## Quiz 4

Sei  $G = (\{A, B, C\}, \{d, e\}, P, A)$  mit

$$P = \{A \rightarrow BBCC, B \rightarrow d, C \rightarrow e, dC \rightarrow Cd\}$$

Welchen Typ hat  $L(G)$ ?

- a) 0
- b) 1
- c) 2
- d) 3

d) stimmt, da  $L(G) = \{ddee, dede, deed, edde, eded, eedd\}$  und es reguläre Grammatik gibt, die  $L(G)$  erzeugt.

arsnova.hs-rm.de  
6750 1376



## Beispiel (kontextsensitive Grammatik)

$G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit

$P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

Beispiel-Ableitung:

$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaSBCBCBC \Rightarrow aaaaBCBCBCBC$   
 $\Rightarrow aaaaabCBCBCBC \Rightarrow aaaaabBCBCBCBC \Rightarrow aaaaabbCCBCBC$   
 $\Rightarrow aaaaabbCBCBCBC \Rightarrow aaaaabbBCCBCBC \Rightarrow aaaaabbBCCBCC$   
 $\Rightarrow aaaaabbBCBCCC \Rightarrow aaaaabbBBCCCC \Rightarrow aaaaabbbbBCCCC$   
 $\Rightarrow aaaaabbbbBCCCC \Rightarrow aaaaabbbbcbCCC \Rightarrow aaaaabbbbcbccc$   
 $\Rightarrow aaaaabbbbcbcccC \Rightarrow aaaaabbbbcbcccc$

Steckengebliebene Folge von Ableitungsschritten:

$S \Rightarrow aSBC \Rightarrow aaBCBC \Rightarrow aabCBC \Rightarrow abcBC$

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\supseteq$ “: Zeige  $a^n b^n c^n \in L(G)$  für alle  $n \in \mathbb{N}$

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\supseteq$ “: Zeige  $a^n b^n c^n \in L(G)$  für alle  $n \in \mathbb{N}$

- Wende  $n - 1$  mal  $S \rightarrow aSBC$  und dann einmal  $S \rightarrow aBC$  an:  
 $S \Rightarrow^* a^{n-1} S (BC)^{n-1} \Rightarrow a^n (BC)^n$

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\supseteq$ “: Zeige  $a^n b^n c^n \in L(G)$  für alle  $n \in \mathbb{N}$

- Wende  $n - 1$  mal  $S \rightarrow aSBC$  und dann einmal  $S \rightarrow aBC$  an:  
 $S \Rightarrow^* a^{n-1} S (BC)^{n-1} \Rightarrow a^n (BC)^n$
- Wende  $CB \rightarrow BC$  solange an, bis es kein Teilwort  $CB$  mehr gibt.  
 $a^n (BC)^n \Rightarrow^* a^n B^n C^n$

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\supseteq$ “: Zeige  $a^n b^n c^n \in L(G)$  für alle  $n \in \mathbb{N}$

- Wende  $n - 1$  mal  $S \rightarrow aSBC$  und dann einmal  $S \rightarrow aBC$  an:  
 $S \Rightarrow^* a^{n-1} S(BC)^{n-1} \Rightarrow a^n (BC)^n$
- Wende  $CB \rightarrow BC$  solange an, bis es kein Teilwort  $CB$  mehr gibt.  
 $a^n (BC)^n \Rightarrow^* a^n B^n C^n$
- Wende  $aB \rightarrow ab$  und anschließend  $n - 1$  mal  $bB \rightarrow bb$  an.  
 $a^n B^n C^n \Rightarrow a^n b B^{n-1} C^n \Rightarrow^* a^n b^n C^n$

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\supseteq$ “: Zeige  $a^n b^n c^n \in L(G)$  für alle  $n \in \mathbb{N}$

- Wende  $n - 1$  mal  $S \rightarrow aSBC$  und dann einmal  $S \rightarrow aBC$  an:  
 $S \Rightarrow^* a^{n-1} S(BC)^{n-1} \Rightarrow a^n (BC)^n$
- Wende  $CB \rightarrow BC$  solange an, bis es kein Teilwort  $CB$  mehr gibt.  
 $a^n (BC)^n \Rightarrow^* a^n B^n C^n$
- Wende  $aB \rightarrow ab$  und anschließend  $n - 1$  mal  $bB \rightarrow bb$  an.  
 $a^n B^n C^n \Rightarrow a^n b B^{n-1} C^n \Rightarrow^* a^n b^n C^n$
- Wende einmal  $bC \rightarrow bc$  und anschließend  $n - 1$  mal  $cC \rightarrow cc$  an  
 $a^n b^n C^n \Rightarrow a^n b^n c C^{n-1} \Rightarrow^* a^n b^n c^n$

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\supseteq$ “: Zeige  $a^n b^n c^n \in L(G)$  für alle  $n \in \mathbb{N}$

- Wende  $n - 1$  mal  $S \rightarrow aSBC$  und dann einmal  $S \rightarrow aBC$  an:  
 $S \Rightarrow^* a^{n-1} S(BC)^{n-1} \Rightarrow a^n (BC)^n$
- Wende  $CB \rightarrow BC$  solange an, bis es kein Teilwort  $CB$  mehr gibt.  
 $a^n (BC)^n \Rightarrow^* a^n B^n C^n$
- Wende  $aB \rightarrow ab$  und anschließend  $n - 1$  mal  $bB \rightarrow bb$  an.  
 $a^n B^n C^n \Rightarrow a^n b B^{n-1} C^n \Rightarrow^* a^n b^n C^n$
- Wende einmal  $bC \rightarrow bc$  und anschließend  $n - 1$  mal  $cC \rightarrow cc$  an  
 $a^n b^n C^n \Rightarrow a^n b^n c C^{n-1} \Rightarrow^* a^n b^n c^n$

Zusammensetzen aller Ableitungsschritte zeigt  $S \Rightarrow^* a^n b^n c^n$ .

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$  (2)

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\subseteq$ “: Zeige, dass alle von  $G$  erzeugten Wörter von der Form  $a^n b^n c^n$  sind.

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$  (2)

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\subseteq$ “: Zeige, dass alle von  $G$  erzeugten Wörter von der Form  $a^n b^n c^n$  sind.

- Für  $S \Rightarrow_G^* u$  mit  $u$  Satzform zeigen die Regeln:  
 $\#_a(u) = \#_b(u) + \#_B(u) = \#_c(u) + \#_C(u)$

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$  (2)

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\subseteq$ “: Zeige, dass alle von  $G$  erzeugten Wörter von der Form  $a^n b^n c^n$  sind.

- Für  $S \Rightarrow_G^* u$  mit  $u$  Satzform zeigen die Regeln:  
 $\#_a(u) = \#_b(u) + \#_B(u) = \#_c(u) + \#_C(u)$
- Für  $S \Rightarrow_G^* w$  mit  $w \in \{a, b, c\}^*$  gilt:  $a$ 's werden ganz links erzeugt, d.h.  $w = a^n w'$  mit  $w' \in \{b, c\}^*$  und  $n = \#_b(w') = \#_c(w')$

Grammatik erzeugt  $\{a^n b^n c^n \mid n \in \mathbb{N}\}$  (2)

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\subseteq$ “: Zeige, dass alle von  $G$  erzeugten Wörter von der Form  $a^n b^n c^n$  sind.

- Für  $S \Rightarrow_G^* u$  mit  $u$  Satzform zeigen die Regeln:  
 $\#_a(u) = \#_b(u) + \#_B(u) = \#_c(u) + \#_C(u)$
- Für  $S \Rightarrow_G^* w$  mit  $w \in \{a, b, c\}^*$  gilt:  $a$ 's werden ganz links erzeugt, d.h.  $w = a^n w'$  mit  $w' \in \{b, c\}^*$  und  $n = \#_b(w') = \#_c(w')$
- Es gilt  $w' = bw_1$ , da jedes auf  $a$  folgende Symbol durch  $aB \rightarrow ab$  erzeugt wird und die Regeln keine Terminalsymbole vertauschen.

### Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  für  $G = (\{S, B, C\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$

„ $\subseteq$ “: Zeige, dass alle von  $G$  erzeugten Wörter von der Form  $a^n b^n c^n$  sind.

- ...
- Ebenso können die Terminalsymbole des Wortes  $w' \in \{b, c\}^*$  nur durch  $bB \rightarrow bb$ ,  $bC \rightarrow bc$  und  $cC \rightarrow cc$  erzeugt worden sein.  
Diese Produktionen erlauben nur **einen Wechsel von  $b$  zu  $c$**  und **keine Wechsel von  $c$  zu  $b$** . Auch ein **Umordnen der Terminalsymbole** ist **nicht möglich** (da es keine Produktion dafür gibt).
- Daher gilt  $w' = b^i c^j$  und mit  $n = \#_b(w') = \#_c(w')$  sogar  $w' = b^n c^n$ . □

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$S$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$ \Rightarrow \$aAT\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ \Rightarrow \$aAaAbB\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ \Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ \Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$S \Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ \Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \end{aligned}$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \Rightarrow \$aabA\$ab \end{aligned}$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \Rightarrow \$aabA\$ab \Rightarrow \$aab\$aab \end{aligned}$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \Rightarrow \$aabA\$ab \Rightarrow \$aab\$aab \Rightarrow a\$ab\$aab \end{aligned}$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \Rightarrow \$aabA\$ab \Rightarrow \$aab\$aab \Rightarrow a\$ab\$aab \\ &\Rightarrow aa\$b\$aab \end{aligned}$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \Rightarrow \$aabA\$ab \Rightarrow \$aab\$aab \Rightarrow a\$ab\$aab \\ &\Rightarrow aa\$b\$aab \Rightarrow aab\$\$aab \end{aligned}$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \Rightarrow \$aabA\$ab \Rightarrow \$aab\$aab \Rightarrow a\$ab\$aab \\ &\Rightarrow aa\$b\$aab \Rightarrow aab\$\$aab \Rightarrow aabaab \end{aligned}$$

# Beispiel einer Typ 0-Grammatik

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Eine Ableitung:

$$\begin{aligned} S &\Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ &\Rightarrow \$aAaAbB\$ \Rightarrow \$aaAAbB\$ \Rightarrow \$aaAbAB\$ \Rightarrow \$aabAAB\$ \\ &\Rightarrow \$aabAA\$b \Rightarrow \$aabA\$ab \Rightarrow \$aab\$aab \Rightarrow a\$ab\$aab \\ &\Rightarrow aa\$b\$aab \Rightarrow aab\$\$aab \Rightarrow aabaab \end{aligned}$$

Beachte:  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  und  $L(G)$  ist Typ 1-Sprache

## Beispiel einer Typ 0-Grammatik (2)

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Begründung dafür, dass  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  gilt:

## Beispiel einer Typ 0-Grammatik (2)

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Begründung dafür, dass  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  gilt:

- Mit  $S \rightarrow \$T\$$  wird zunächst eine Umrahmung mit  $\$\$$  erzeugt

## Beispiel einer Typ 0-Grammatik (2)

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Begründung dafür, dass  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  gilt:

- Mit  $S \rightarrow \$T\$$  wird zunächst eine Umrahmung mit  $\$\$$  erzeugt
- Mit  $T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon$  wird ein Wort aus 2er Blöcken  $aA$  und  $bB$  erzeugt

## Beispiel einer Typ 0-Grammatik (2)

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Begründung dafür, dass  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  gilt:

- Mit  $S \rightarrow \$T\$$  wird zunächst eine Umrahmung mit  $\$\$$  erzeugt
- Mit  $T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon$  wird ein Wort aus 2er Blöcken  $aA$  und  $bB$  erzeugt
- Mit  $Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB$  werden  $A$ 's und  $B$ 's bis vor  $\$$  geschoben

## Beispiel einer Typ 0-Grammatik (2)

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Begründung dafür, dass  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  gilt:

- Mit  $S \rightarrow \$T\$$  wird zunächst eine Umrahmung mit  $\$\$$  erzeugt
- Mit  $T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon$  wird ein Wort aus 2er Blöcken  $aA$  und  $bB$  erzeugt
- Mit  $Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB$  werden  $A$ 's und  $B$ 's bis vor  $\$$  geschoben
- Mit  $A\$ \rightarrow \$a$  und  $B\$ \rightarrow \$b$  werden die  $A$ 's und  $B$ 's in  $a$ 's und  $b$ 's verwandelt, indem sie über das rechte  $\$$  hüpfen.

## Beispiel einer Typ 0-Grammatik (2)

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Begründung dafür, dass  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  gilt:

- Mit  $S \rightarrow \$T\$$  wird zunächst eine Umrahmung mit  $\$\$$  erzeugt
- Mit  $T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon$  wird ein Wort aus 2er Blöcken  $aA$  und  $bB$  erzeugt
- Mit  $Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB$  werden  $A$ 's und  $B$ 's bis vor  $\$$  geschoben
- Mit  $A\$ \rightarrow \$a$  und  $B\$ \rightarrow \$b$  werden die  $A$ 's und  $B$ 's in  $a$ 's und  $b$ 's verwandelt, indem sie über das rechte  $\$$  hüpfen.
- Mit  $\$a \rightarrow a\$, \$b \rightarrow b\$, \$\$ \rightarrow \varepsilon$  wird das linke  $\$$  zum rechten geschoben, mit  $\$\$ \rightarrow \varepsilon$  werden sie dann eliminiert.

## Beispiel einer Typ 0-Grammatik (2)

Grammatik  $G = (\{S, T, A, B, \$\}, \{a, b\}, P, S)$  mit

$$P = \{S \rightarrow \$T\$, T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon, \$a \rightarrow a\$, \$b \rightarrow b\$, Aa \rightarrow aA, \\ Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, A\$ \rightarrow \$a, B\$ \rightarrow \$b, \$\$ \rightarrow \varepsilon\}$$

Begründung dafür, dass  $L(G) = \{ww \mid w \in \{a, b\}^*\}$  gilt:

- Mit  $S \rightarrow \$T\$$  wird zunächst eine Umrahmung mit  $\$\$$  erzeugt
- Mit  $T \rightarrow aAT, T \rightarrow bBT, T \rightarrow \varepsilon$  wird ein Wort aus 2er Blöcken  $aA$  und  $bB$  erzeugt
- Mit  $Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB$  werden  $A$ 's und  $B$ 's bis vor  $\$$  geschoben
- Mit  $A\$ \rightarrow \$a$  und  $B\$ \rightarrow \$b$  werden die  $A$ 's und  $B$ 's in  $a$ 's und  $b$ 's verwandelt, indem sie über das rechte  $\$$  hüpfen.
- Mit  $\$a \rightarrow a\$, \$b \rightarrow b\%$  wird das linke  $\$$  zum rechten geschoben, mit  $\$\$ \rightarrow \varepsilon$  werden sie dann eliminiert.
- Bei allen Schritten wird die relative Lage aller  $a$  zu  $b$  sowie aller  $A$  zu  $B$  nicht geändert.

## $\varepsilon$ -PRODUKTIONEN

- in Typ 1,2,3-Grammatiken
- in Typ 2,3-Grammatiken

- Das leere Wort  $\varepsilon$  kann bisher nicht für Typ 1,2,3 Grammatiken erzeugt werden:  
Produktion  $S \rightarrow \varepsilon$  erfüllt die Typ 1-Bedingung  $|S'| \leq |\varepsilon|$  nicht. Daher Sonderregel:

## $\varepsilon$ -Regel für Typ 1-Grammatiken

Eine Grammatik  $G = (V, \Sigma, P, S)$  vom Typ 1 darf eine Produktion  $(S \rightarrow \varepsilon) \in P$  enthalten, vorausgesetzt, dass keine rechte Seite einer Produktion in  $P$ , die Variable  $S$  enthält.

### Sonderregel erlaubt nicht:

$$G = (\{S\}, \{a\}, \{S \rightarrow \varepsilon, S \rightarrow aSa\}, S)$$

### Sonderregel erlaubt:

$$G = (\{S', S\}, \{a\}, \{S' \rightarrow \varepsilon, S' \rightarrow aSa, S' \rightarrow aa, S \rightarrow aSa, S \rightarrow aa\}, S')$$

## Satz

Sei  $G = (V, \Sigma, P, S)$  vom Typ  $i \in \{1, 2, 3\}$  mit  $\varepsilon \notin L(G)$ . Sei  $S' \notin V$ .

Dann erzeugt  $G' = (V \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow \varepsilon\} \cup \{S' \rightarrow r \mid S \rightarrow r \in P\}, S')$  die Sprache  $L(G') = L(G) \cup \{\varepsilon\}$  und

$G'$  erfüllt die  $\varepsilon$ -Regel für Typ 1,2,3-Grammatiken und  $G'$  ist vom Typ  $i$ .

## Satz

Sei  $G = (V, \Sigma, P, S)$  vom Typ  $i \in \{1, 2, 3\}$  mit  $\varepsilon \notin L(G)$ . Sei  $S' \notin V$ .

Dann erzeugt  $G' = (V \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow \varepsilon\} \cup \{S' \rightarrow r \mid S \rightarrow r \in P\}, S')$  die Sprache  $L(G') = L(G) \cup \{\varepsilon\}$  und

$G'$  erfüllt die  $\varepsilon$ -Regel für Typ 1,2,3-Grammatiken und  $G'$  ist vom Typ  $i$ .

Beweis:

- Da  $S'$  neu, kommt  $S'$  auf keiner rechten Seite vor.
- Da  $S \rightarrow r \in P$  vom Typ  $i$ , sind auch  $S' \rightarrow r$  vom Typ  $i$
- Da  $S' \Rightarrow \varepsilon$ , gilt  $\varepsilon \in L(G')$
- Für  $w \neq \varepsilon$  gilt:  $S \Rightarrow_G^* w$  g.d.w.  $S' \Rightarrow_{G'}^* w$   
Der jeweils erste Ableitungsschritt muss ausgetauscht werden,  
d.h.  $S \Rightarrow_G r$  vs.  $S' \Rightarrow_{G'} r$

Sonderregel für Typ 2- und Typ 3-Grammatiken:

## $\varepsilon$ -Produktionen in kontextfreien und regulären Grammatiken

In Grammatiken des Typs 2 und des Typs 3 erlauben wir Produktionen der Form  $A \rightarrow \varepsilon$  (sogenannte  $\varepsilon$ -Produktionen).

Das ist keine echte Erweiterung, denn:

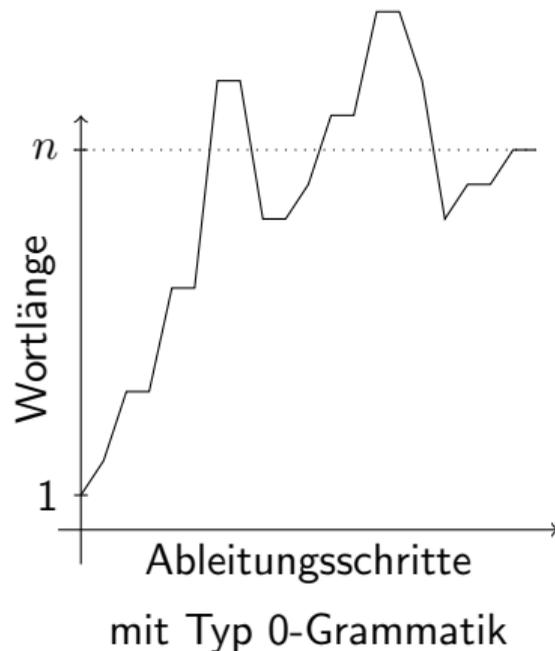
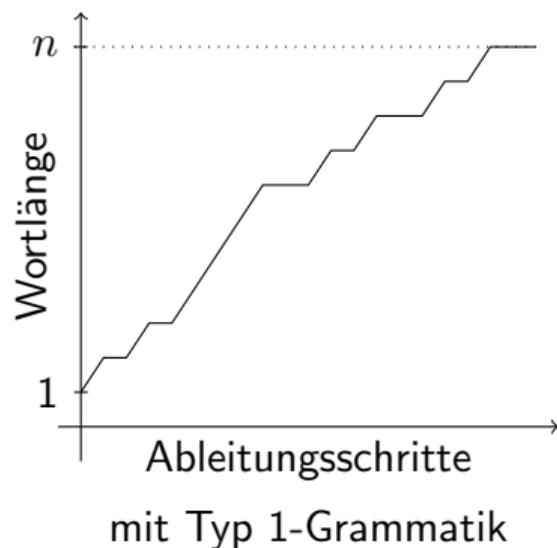
### Satz (Entfernen von $\varepsilon$ -Produktionen)

Sei  $G = (V, \Sigma, P, S)$  eine kontextfreie (bzw. reguläre) Grammatik mit  $\varepsilon \notin L(G)$ . Dann gibt es eine kontextfreie (bzw. reguläre) Grammatik  $G'$  mit  $L(G) = L(G')$  und  $G'$  enthält keine  $\varepsilon$ -Produktionen.

- Kontextfreie Produktionen  $A \rightarrow r$  sind immer auf ein Vorkommen von  $A$  anwendbar.
- Kontextsensitive Produktionen können solche Ersetzungen  
auf einen Kontext einschränken  
und erlauben Regeln  $uAv \rightarrow urv$ , die die Ersetzung von  $A$  durch  $r$  nur erlauben, wenn  $A$  durch  $u$  und  $v$  umrahmt ist.

# Typ 0 vs. Typ 1

Ableitung eines Wortes der Länge  $n$



SYNTAXBÄUME, ABLEITUNGEN,  
MEHRDEUTIGE GRAMMATIKEN,  
BACKUS-NAUR-FORM

## Definition (Syntaxbaum)

Sei  $G = (V, \Sigma, P, S)$  eine Typ 2-Grammatik und

$$S \Rightarrow w_0 \Rightarrow \dots \Rightarrow w_n$$

eine Ableitung von  $w_n \in \Sigma^*$ .

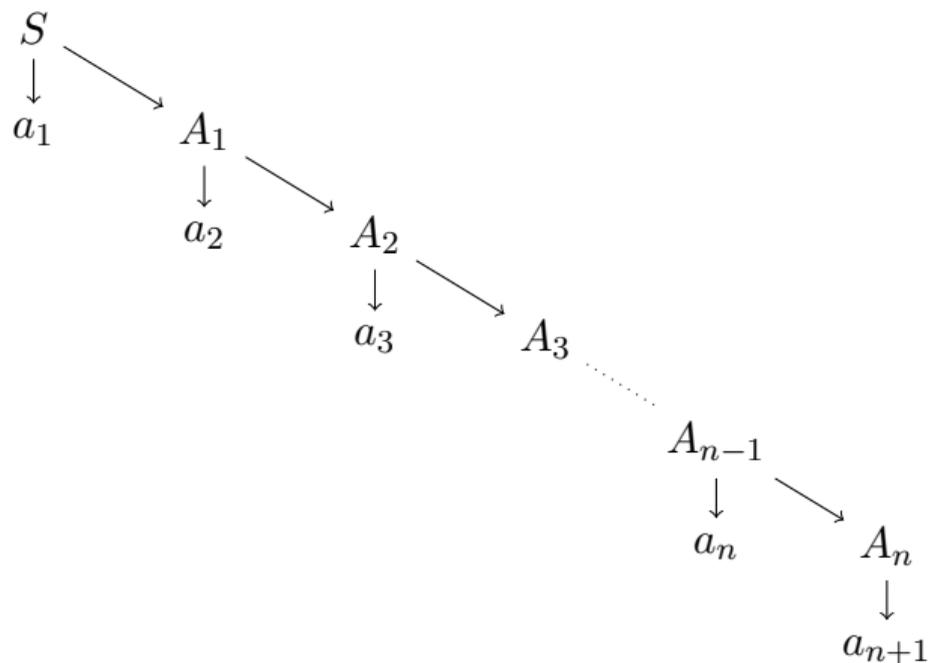
Der **Syntaxbaum** zur Ableitung wird wie folgt erstellt:

- Die Wurzel des Baums ist mit  $S$  markiert.
- Wenn  $w_i \Rightarrow w_{i+1}$ ,  $w_i = uAv$  und  $w_{i+1} = urv$  (Produktion  $A \rightarrow r$  verwendet), dann erzeuge im Syntaxbaum  $|r|$  viele Knoten als Kinder des mit  $A$  markierten Knotens. Markiere die Kinder mit den Symbolen aus  $r$  (in der Reihenfolge von links nach rechts).

Die Blätter sind daher genau mit dem Wort  $w_n$  markiert.

# Syntaxbäume bei Typ 3-Grammatiken

Sind immer Listenartig:



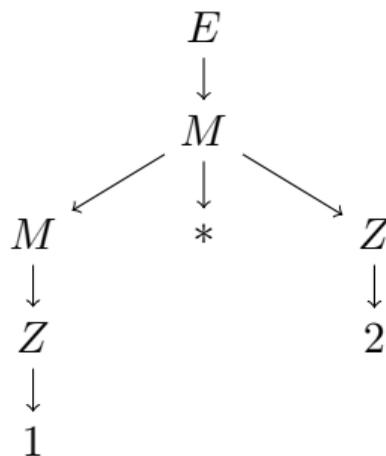
# Beispiel

$$G = (\{E, M, Z\}, \{+, *, 1, 2, (, )\}, P, E) \text{ mit}$$
$$P = \{E \rightarrow M, E \rightarrow E + M, M \rightarrow Z, M \rightarrow M * Z, \\ Z \rightarrow 1, Z \rightarrow 2, Z \rightarrow (E)\}$$

Beide Ableitungen:

- $E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow 1 * Z \Rightarrow 1 * 2$  und
- $E \Rightarrow M \Rightarrow M * Z \Rightarrow M * 2 \Rightarrow Z * 2 \Rightarrow 1 * 2$

haben **denselben** Syntaxbaum.



- **Linksableitung:** Ersetze immer das linkeste Nichtterminal der Satzform.
- **Rechtsableitung:** Ersetze immer das rechteste Nichtterminal der Satzform.

Beispiele:

$$\begin{aligned} E &\Rightarrow E + M \\ &\Rightarrow M + M \\ &\Rightarrow M * Z + M \\ &\Rightarrow Z * Z + M \\ &\Rightarrow 1 * Z + M \\ &\Rightarrow 1 * 2 + M \\ &\Rightarrow 1 * 2 + Z \\ &\Rightarrow 1 * 2 + 3 \end{aligned}$$

$$\begin{aligned} E &\Rightarrow E + M \\ &\Rightarrow E + Z \\ &\Rightarrow E + 3 \\ &\Rightarrow M + 3 \\ &\Rightarrow M * Z + 3 \\ &\Rightarrow M * 2 + 3 \\ &\Rightarrow Z * 2 + 3 \\ &\Rightarrow 1 * 2 + 3 \end{aligned}$$

### Satz

Sei  $G$  eine Typ 2-Grammatik und  $w \in L(G)$ . Dann gibt es eine Linksableitung (und eine Rechtsableitung) von  $w$ .

Beweis:

- Da  $w \in L(G)$ , gibt es irgendeine Ableitung von  $w$ .
- Konstruiere Syntaxbaum zu dieser Ableitung.
- Lese Links- bzw. Rechtsableitung am Syntaxbaum ab.

# Mehrdeutige Grammatiken

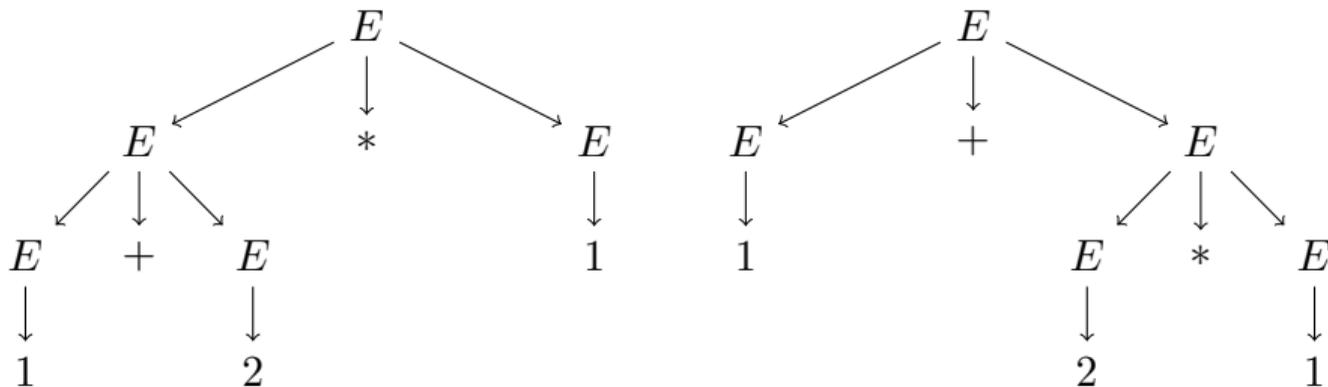
Beispiel:

$$(E, \{*, +, 1, 2\}, \{E \rightarrow E * E, E \rightarrow E + E, E \rightarrow 1, E \rightarrow 2\}, E)$$

Zwei Ableitungen für  $1 + 2 * 1$ :

- $E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow 1 + E * E \Rightarrow 1 + 2 * E \Rightarrow 1 + 2 * 1$
- $E \Rightarrow E + E \Rightarrow E + E * E \Rightarrow 1 + E * E \Rightarrow 1 + 2 * E \Rightarrow 1 + 2 * 1.$

Syntaxbäume dazu:



## Mehrdeutige Grammatik

Eine Typ 2-Grammatik ist mehrdeutig, wenn es verschieden strukturierte Syntaxbäume für dasselbe Wort  $w$  gibt.

## Inhärent mehrdeutige Sprache

Eine Typ 2-Sprache ist inhärent mehrdeutig, wenn es nur mehrdeutige Grammatiken gibt, die diese Sprache erzeugen.

Die Sprache

$$\{a^m b^m c^n d^n \mid m, n \in \mathbb{N}\} \cup \{a^m b^n c^n d^m \mid m, n \in \mathbb{N}\}$$

ist inhärent mehrdeutig (Beweis z.B. in Hopcroft, Motwani, Ullman, 2006)

## Erweiterte Backus-Naur-Form (EBNF)

Für Typ 2-Grammatiken erlauben wir abkürzende Schreibweise für die Menge der Produktionen  $P$ :

- 1 Statt  $A \rightarrow w_1, A \rightarrow w_2, \dots, A \rightarrow w_n$  schreiben wir auch  $A \rightarrow w_1 \mid w_2 \mid \dots \mid w_n$
- 2 Die Schreibweise  $A \rightarrow u[v]w$  steht für die beiden Produktionen  $A \rightarrow uww$  und  $A \rightarrow uw$  (d. h.  $[v]$  meint, dass  $v$  optional ist).
- 3 Die Schreibweise  $A \rightarrow u\{v\}w$  steht für  $A \rightarrow uw$  oder  $A \rightarrow uBw$  mit  $B \rightarrow v \mid vB$  (d. h.  $\{v\}$  meint, dass  $v$  beliebig oft wiederholt werden kann).

Grammatiken, die diese Notation verwenden, nennen wir auch Grammatiken in **erweiterter Backus-Naur-Form (EBNF)**

# EIGENSCHAFTEN VON SPRACHEN UND SPRACHKLASSEN

- Teilmengenbeziehungen der Chomsky-Hierarchie
- Abgeschlossenheit von Sprachklassen
- Entscheidbarkeit von Sprachen

# Chomsky-Hierarchie: Teilmengenbeziehungen

Aus der Definition der Typ  $i$ -Sprachen folgt:

Typ 3-Sprachen  $\subseteq$  Typ 2-Sprachen  $\subseteq$  Typ 1-Sprachen  $\subseteq$  Typ 0-Sprachen

# Chomsky-Hierarchie: Teilmengenbeziehungen

Aus der Definition der Typ  $i$ -Sprachen folgt:

Typ 3-Sprachen  $\subseteq$  Typ 2-Sprachen  $\subseteq$  Typ 1-Sprachen  $\subseteq$  Typ 0-Sprachen

Es gilt sogar:

Typ 3-Sprachen  $\subset$  Typ 2-Sprachen  $\subset$  Typ 1-Sprachen  $\subset$  Typ 0-Sprachen

Aus der Definition der Typ  $i$ -Sprachen folgt:

Typ 3-Sprachen  $\subseteq$  Typ 2-Sprachen  $\subseteq$  Typ 1-Sprachen  $\subseteq$  Typ 0-Sprachen

Es gilt sogar:

Typ 3-Sprachen  $\subset$  Typ 2-Sprachen  $\subset$  Typ 1-Sprachen  $\subset$  Typ 0-Sprachen

Trennende Beispiele sind (Beweise folgen im Laufe der Vorlesung):

- $L = \{a^n b^n \mid n \in \mathbb{N}\}$  ist von Typ 2, aber nicht von Typ 3
- $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$  ist von Typ 1, aber nicht von Typ 2.
- $H = \{w\$x \mid \text{Turingmaschine } M_w \text{ h\u00e4lt f\u00fcr Eingabe } x\}$   
(das sogenannte Halteproblem) ist von Typ 0, aber nicht von Typ 1.

Beachte: Es gibt auch Sprachen, die nicht Typ 0 sind:

Das Komplement von  $H$  ist eine solche Sprache.

Eine Klasse  $\mathcal{L}$  von Sprachen (d.h. eine Menge von Mengen) heißt **abgeschlossen bezüglich**

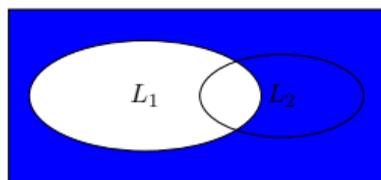
- **Vereinigung** g.d.w. aus  $L_1, L_2 \in \mathcal{L}$  folgt stets  $(L_1 \cup L_2) \in \mathcal{L}$ ,
- **Schnittbildung** g.d.w. aus  $L_1, L_2 \in \mathcal{L}$  folgt stets  $(L_1 \cap L_2) \in \mathcal{L}$ ,
- **Komplementbildung** g.d.w. aus  $L \in \mathcal{L}$  folgt stets  $\bar{L} \in \mathcal{L}$  und
- **Produktbildung** g.d.w. aus  $L_1, L_2 \in \mathcal{L}$  folgt stets  $(L_1 L_2) \in \mathcal{L}$ .

Wir werden im Laufe der Vorlesung untersuchen, ob die Typ i-Sprachen abgeschlossen bezüglich obiger Operationen sind

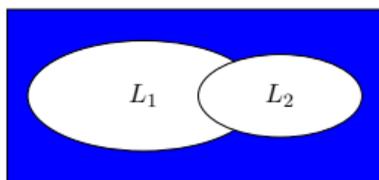
## Satz

Sei Klasse  $\mathcal{L}$  abgeschlossen bez. Komplementbildung.  $\mathcal{L}$  ist abgeschlossen bez. Schnittbildung genau dann, wenn  $\mathcal{L}$  abgeschlossen bez. Vereinigung ist.

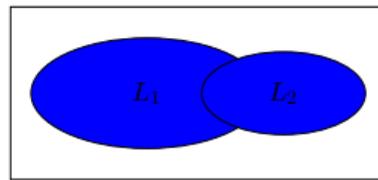
Das gilt, da:  $L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$  und  $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$



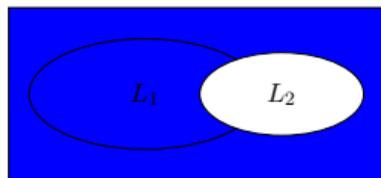
■  $\overline{L_1}$



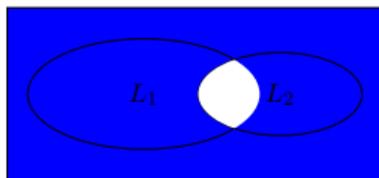
■  $\overline{L_1 \cap L_2}$



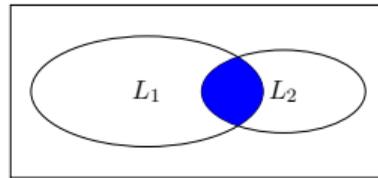
■  $\overline{\overline{L_1} \cap \overline{L_2}}$



■  $\overline{L_2}$



■  $\overline{L_1 \cup L_2}$



■  $\overline{\overline{L_1} \cup \overline{L_2}}$

## Entscheidbarkeit

Eine Sprache heißt **entscheidbar**, wenn es einen Algorithmus gibt, der bei Eingabe der Grammatik  $G$  und einem Wort  $w$  in endlicher Zeit feststellt, ob  $w \in L(G)$  gilt oder nicht.

Man spricht auch von der Entscheidbarkeit des Wortproblems!

## Entscheidbarkeit

Eine Sprache heißt **entscheidbar**, wenn es einen Algorithmus gibt, der bei Eingabe der Grammatik  $G$  und einem Wort  $w$  in endlicher Zeit feststellt, ob  $w \in L(G)$  gilt oder nicht.

Man spricht auch von der Entscheidbarkeit des Wortproblems!

### Eigenschaften der Typ $i$ -Sprachen:

- Alle Typ 1, 2, 3-Sprachen sind **entscheidbar** (sehen wir gleich)
- Es gibt Typ 0-Sprachen, die **nicht entscheidbar** sind.
- Alle Typ 0-Sprachen sind **semi-entscheidbar (rekursiv aufzählbar)**:  
Es gibt einen Algorithmus, der bei Eingabe der Grammatik  $G$  und einem Wort  $w \in G$  in endlicher Zeit feststellt, dass  $w \in L(G)$  gilt, und bei einem Wort  $w \notin G$  entweder feststellt, dass  $w \notin L(G)$  gilt, **oder nicht-terminiert**.

# DAS WORTPRBOLEM

- Definition
- Entscheidbarkeit für Typ 1-Sprachen

## Definition (Wortproblem für Typ i-Sprachen)

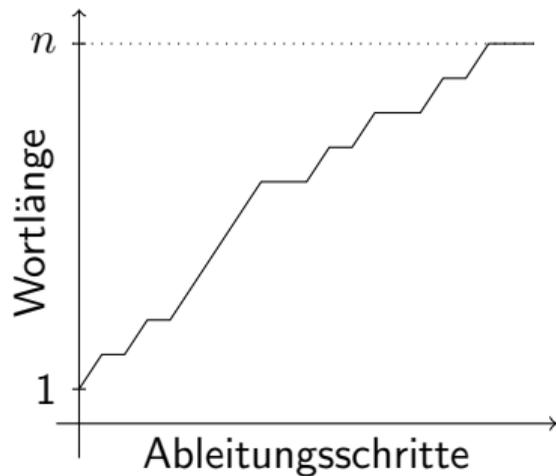
Das **Wortproblem** für Typ i-Sprachen ist die Frage, ob für eine gegebene Typ i-Grammatik  $G = (V, \Sigma, P, S)$  und ein Wort  $w \in \Sigma^*$  gilt:  $w \in L(G)$  oder  $w \notin L(G)$ .

## Satz

Das Wortproblem für Typ 1-Sprachen ist entscheidbar: Es gibt einen Algorithmus, der bei Eingabe von Typ 1-Grammatik  $G$  und Wort  $w$  nach endlicher Zeit entscheidet, ob  $w \in L(G)$  oder  $w \notin L(G)$  gilt.

# Idee zum Beweis des Satzes

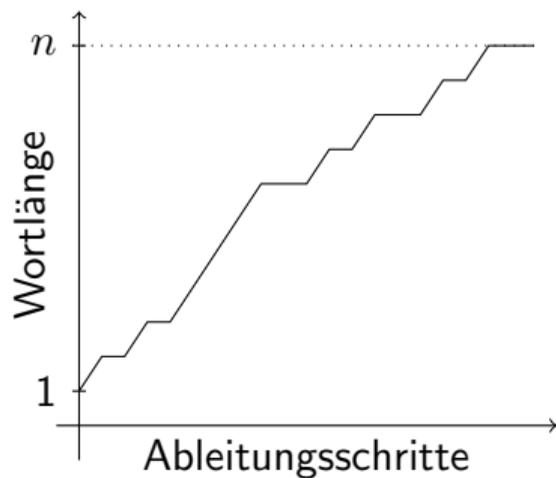
Ableitung eines Wortes der Länge  $n$



mit Typ 1-Grammatik

# Idee zum Beweis des Satzes

Ableitung eines Wortes der Länge  $n$



mit Typ 1-Grammatik

Idee zum Entscheiden des Wortproblems:

- Betrachte  $S \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_m$  mit  $|w_m| = n$
- Da Typ 1-Produktionen nicht verkürzend sind:  
 $\forall 1 \leq i \leq m : |w_i| \leq n$
- Probiere systematisch für alle Satzformen der Länge  $\leq n$  durch, ob sie vom Startsymbol aus ableitbar sind
- Es gibt nur endlich viele Satzformen der Länge  $\leq n$  und jede Typ 1-Grammatik leitet nur endlich viele Satzformen der Länge  $\leq n$  her, **ohne dabei längere Satzformen zwischendrin herzuleiten**
- Herleitbare Satzformen der Länge  $n$  können **rekursiv** aus den Satzformen der Länge  $< n$  berechnet werden

Sei  $G = (V, \Sigma, P, S)$  eine Typ 1-Grammatik und  $w \in \Sigma^*$ .

Für  $m, n \in \mathbb{N}_0$  sei

$L_m^n$  = Menge aller Satzformen der Länge höchstens  $n$ ,  
die in höchstens  $m$  Schritten von  $S$  aus ableitbar sind

$L_m^n := \{w \in (V \cup \Sigma)^* \mid |w| \leq n \text{ und } S \Rightarrow_G^k w \text{ mit } k \leq m\}$

Rekursive Berechnung der Mengen (für  $n > 0$ ):

$$L_0^n := \{S\}$$

$$L_m^n := \text{next}(L_{m-1}^n, n) \text{ für } m > 0$$

wobei  $\text{next}(L, n) := L \cup \{w' \mid w \in L, w \Rightarrow_G w', |w'| \leq n\}$

Die Berechnung terminiert, da die Mengen endlich sind:  $|L_m^n| \leq |\Sigma \cup V|^{n+1}$

Wir können auch iterativ aus  $L_{i-1}^n$  die nachfolgende Menge  $L_i^n$  berechnen

Iterative Berechnung:

- Starte mit  $L_0^n = \{S\}$
- Für  $i = 1, 2, 3, \dots$  berechne  $L_i^n = \text{next}(L_{i-1}^n)$
- Stoppe dabei, wenn  $L_i^n = L_{i-1}^n$
- Prüfe, ob  $w \in L_i^n$  gilt.

**Korrektheit:** Wenn  $L_i^n = L_{i-1}^n$ , dann gilt  $L_{i+k}^n = L_{i-1}^n$  für alle  $k \in \mathbb{N}_0$  und daher enthält  $L_{i-1}^n$  genau alle aus  $S$  ableitbaren Wörter der Länge  $n$ .

**Terminierung:** Beweis durch Widerspruch.

- Nehme an, die Berechnung stoppt nicht.
- Da  $L_{i-1}^n \subseteq L_i^n$ , muss für alle  $i \in \mathbb{N}_0$  gelten:  $L_i^n \supset L_{i-1}^n$
- Daher gilt für alle  $i \in \mathbb{N}_0$ :  $|L_i^n| > |L_{i-1}^n|$ .
- Widerspruch zu  $|L_i^n| \leq |V \cup \Sigma|^{n+1}$  für alle  $i \in \mathbb{N}_0$ .

## Algorithmus 1: Wortproblem entscheiden für Typ 1-Grammatiken

**Eingabe:** Typ 1-Grammatik  $G = (V, \Sigma, P, S)$  und ein Wort  $w \in \Sigma^*$

**Ausgabe:** Ja, wenn  $w \in L(G)$  und Nein, wenn  $w \notin L(G)$

**Beginn**

$n := |w|;$

$L := \{S\};$

**wiederhole**

$L_{\text{old}} := L;$

$L := \text{next}(L_{\text{old}}, n);$

**bis** ( $w \in L$ ) oder ( $L_{\text{old}} = L$ );

**wenn**  $w \in L$  **dann**

**return** Ja;

**sonst**

**return** Nein;

# Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen  $L_i^6$  für alle  $i$ :

# Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen  $L_i^6$  für alle  $i$ :

$$L_0^6 = \{S\}$$

# Beispiel

$$G = (\{S, B\}, \{a, b, c\}, P, S) \text{ mit}$$
$$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$$

Wir berechnen  $L_i^6$  für alle  $i$ :

$$L_0^6 = \{S\}$$

$$L_1^6 = \text{next}(L_0^6) = L_0^6 \cup \{w' \mid w \in L_0^6, w \Rightarrow w', |w'| \leq 6\} = \{S, aSBc, abc\},$$

da  $S \Rightarrow aSBc$  und  $S \Rightarrow abc$

# Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen  $L_i^6$  für alle  $i$ :

$$L_0^6 = \{S\}$$

$$L_1^6 = \text{next}(L_0^6) = L_0^6 \cup \{w' \mid w \in L_0^6, w \Rightarrow w', |w'| \leq 6\} = \{S, aSBc, abc\},$$

da  $S \Rightarrow aSBc$  und  $S \Rightarrow abc$

$$L_2^6 = \text{next}(L_1^6) = L_1^6 \cup \{w' \mid w \in L_1^6, w \Rightarrow w', |w'| \leq 6\} \\ = \{S, aSBc, abc, abcBc\}, \text{ da } aSBc \Rightarrow aaSBcBc \text{ (zu lang) und } aSBc \Rightarrow aabcBc$$

# Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$  mit  
 $P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen  $L_i^6$  für alle  $i$ :

$$L_0^6 = \{S\}$$

$$L_1^6 = \text{next}(L_0^6) = L_0^6 \cup \{w' \mid w \in L_0^6, w \Rightarrow w', |w'| \leq 6\} = \{S, aSBc, abc\},$$

da  $S \Rightarrow aSBc$  und  $S \Rightarrow abc$

$$L_2^6 = \text{next}(L_1^6) = L_1^6 \cup \{w' \mid w \in L_1^6, w \Rightarrow w', |w'| \leq 6\} \\ = \{S, aSBc, abc, abcBc\}, \text{ da } aSBc \Rightarrow aaSBcBc \text{ (zu lang) und } aSBc \Rightarrow aabcBc$$

$$L_3^6 = \text{next}(L_2^6) = L_2^6 \cup \{w' \mid w \in L_2^6, w \Rightarrow w', |w'| \leq 6\} \\ = \{S, aSBc, abc, aabcBc, aabBcc\}, \text{ da } aabcBc \Rightarrow aabBcc$$

## Beispiel (2)

$$G = (\{S, B\}, \{a, b, c\}, P, S)$$

$$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}.$$

...

$$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$$

## Beispiel (2)

$$G = (\{S, B\}, \{a, b, c\}, P, S)$$
$$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}.$$

...

$$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$$

$$L_4^6 = \text{next}(L_3^6) = L_3^6 \cup \{w' \mid w \in L_3^6, w \Rightarrow w', |w'| \leq 6\}$$
$$= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\}$$

da  $aabBcc \Rightarrow aabbcc$

## Beispiel (2)

$$G = (\{S, B\}, \{a, b, c\}, P, S)$$
$$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}.$$

...

$$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$$

$$L_4^6 = \text{next}(L_3^6) = L_3^6 \cup \{w' \mid w \in L_3^6, w \Rightarrow w', |w'| \leq 6\}$$
$$= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\}$$

da  $aabBcc \Rightarrow aabbcc$

$$L_5^6 = \text{next}(L_4^6) = L_4^6 \cup \{w' \mid w \in L_4^6, w \Rightarrow w', |w'| \leq 6\}$$
$$= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\} = L_4^6$$

## Beispiel (2)

$$G = (\{S, B\}, \{a, b, c\}, P, S)$$
$$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}.$$

...

$$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$$

$$L_4^6 = \text{next}(L_3^6) = L_3^6 \cup \{w' \mid w \in L_3^6, w \Rightarrow w', |w'| \leq 6\}$$
$$= \{S, aSBc, abc, aabcBc, aabBcc, aabbbc\}$$

da  $aabBcc \Rightarrow aabbbc$

$$L_5^6 = \text{next}(L_4^6) = L_4^6 \cup \{w' \mid w \in L_4^6, w \Rightarrow w', |w'| \leq 6\}$$
$$= \{S, aSBc, abc, aabcBc, aabBcc, aabbbc\} = L_4^6$$

d.h.  $L_i^6 = \{S, aSBc, abc, aabcBc, aabBcc, aabbbc\}$  für alle  $i \geq 4$

**Erinnerung:**

**ANMELDUNG** für die Studienleistung  
im **Compass**  
bis **05.05.** erforderlich

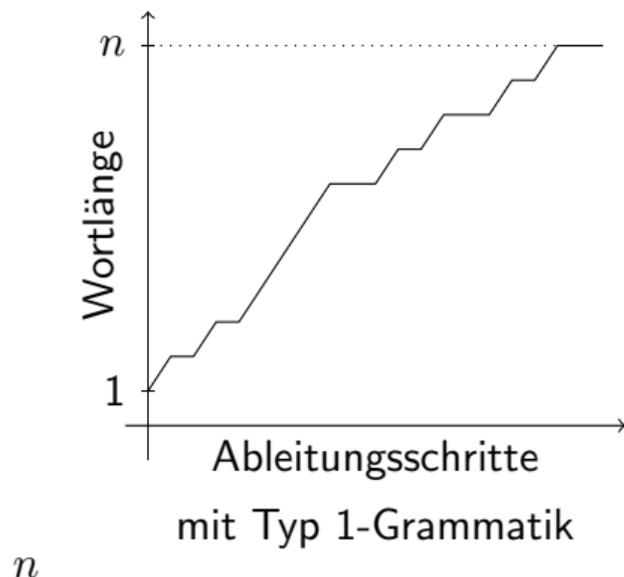
## Definition (Wortproblem)

Das **Wortproblem** für Typ  $i$ -Sprachen ist die Frage, ob für eine gegebene Typ  $i$ -Grammatik  $G = (V, \Sigma, P, S)$  und ein Wort  $w \in \Sigma^*$  gilt:  $w \in L(G)$  oder  $w \notin L(G)$ .

## Satz

Das Wortproblem für Typ 1-Sprachen ist entscheidbar: Es gibt einen Algorithmus, der bei Eingabe von Typ 1-Grammatik  $G$  und Wort  $w$  nach endlicher Zeit entscheidet, ob  $w \in L(G)$  oder  $w \notin L(G)$  gilt.

Ableitung eines Wortes der Länge



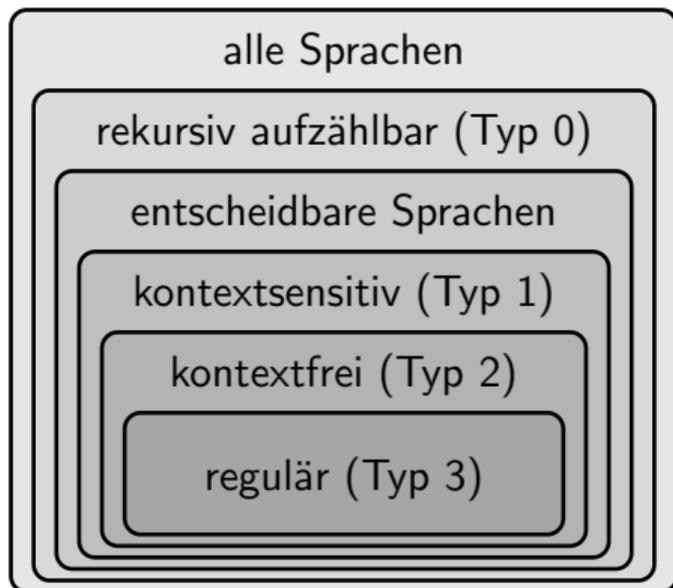
## Korollar

Das Wortproblem für Typ 2- und Typ 3-Sprachen ist entscheidbar.

Bemerkungen:

- Der Algorithmus für Typ 1-Sprachen hat exponentielle Laufzeitkomplexität
- Das Wortproblem für Typ 2- und das Wortproblem für Typ 3-Sprachen sind in polynomieller Zeit lösbar.
- Das Wortproblem für Typ 0-Sprachen ist unentscheidbar (aber rekursiv aufzählbar)

# Übersicht über die Sprachen



- Die Menge der Typ 0-Grammatiken ist abzählbar (jede Grammatik hat eine endliche Beschreibung, daher Grammatiken der Größe nach aufzählen)
- Menge aller Sprachen =  $\mathcal{P}(\Sigma^*)$  ist überabzählbar!

## Definition

- Das **Leerheitsproblem** für Sprachen vom Typ  $i$  ist die Frage, ob für eine Typ  $i$ -Grammatik  $G$  die Gleichheit  $L(G) = \emptyset$  gilt.
- Das **Endlichkeitsproblem** für Sprachen vom Typ  $i$  ist die Frage, ob für eine Typ  $i$ -Grammatik  $G$  die Ungleichheit  $|L| < \infty$  gilt.
- Das **Schnittproblem** für Sprachen vom Typ  $i$  ist die Frage, ob für Typ  $i$ -Grammatiken  $G_1, G_2$  gilt:  $L(G_1) \cap L(G_2) = \emptyset$ .
- Das **Äquivalenzproblem** für Sprachen vom Typ  $i$  ist, die Frage, ob Typ  $i$ -Grammatiken  $G_1, G_2$  gilt:  $L(G_1) = L(G_2)$ .

Aus informatischer Sicht:

- Typ 2- und Typ 3-Sprachen sind wichtig im Rahmen des Compilerbau (lexikalische bzw. syntaktische Analyse)
- Viele Fragestellungen sind jedoch kontextsensitiv oder Typ 0.
- Praktisches Vorgehen: Typ 2-Sprache plus Nebenbedingungen, z.B. Syntax als kontextfreie Grammatik aber noch Nebenbedingungen, dass alle Variablen deklariert wurden usw.