LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Formale Sprachen und Komplexität
Sommersemester 2019

Worte, Formale Sprachen, Grammatiken und die Chomsky-Hierarchie

Prof. Dr. David Sabel

LFE Theoretische Informatik

Letzte Änderung der Folien: 9. Mai 2019

Worte

Alphabet

Ein Alphabet Σ ist eine endliche nicht-leere Menge von Zeichen (oder Symbolen).

Z.B. $\Sigma = \{a, b, c, d, e\}$

Wort

Ein Wort w über Σ ist eine endliche Folge von Zeichen aus Σ.

Beispiele:

- $bade$ ist ein Wort über $\{a, b, c, d, e\}$
- $baden$ ist kein Wort über $\{a, b, c, d, e\}$

Weitere Notationen zu Worten

- Das leere Wort wird als ε notiert.
- Für $w = a_1 \cdots a_n$ ist $|w| = n$ die Länge des Wortes
- Für $1 \leq i \leq |w|$ ist $w[i]$ das Zeichen an i. Position in w.
- Für $a \in \Sigma$ und w ein Wort über Σ sei $\#_a(w) \in \mathbb{N}$ die Anzahl an Vorkommen des Zeichens a im Wort w

Beispiele:

- Es gilt $|\varepsilon| = 0$ und $\#_a(\varepsilon) = 0$ für alle $a \in \Sigma$.
- Für $\Sigma = \{a, b, c\}$ ist
  - $|abbccc| = 6$
  - $|aabbbccc| = 8$
  - $\#_a(abbccc) = 1$
  - $\#_c(aabbbccc) = 3$
- Für $w = abbbcd$ ist $w[1] = a$, $w[5] = c$ und $w[7]$ undefiniert.

Konkatenation und Kleene-Stern

Konkatenation

Das Wort uv (alternativ $u \circ v$) entsteht, indem Wort v hinten an Wort u angehängt wird.

Σ^* bezeichnen wir die Menge aller Wörter über Σ.

Definition von $\Sigma^i, \Sigma^*, \Sigma^+$

Sei Σ ein Alphabet, dann definieren wir:

$$\begin{array}{rcl} \Sigma^0 & := & \{\varepsilon\} \\ \Sigma^i & := & \{aw \mid a \in \Sigma, w \in \Sigma^{i-1}\} \text{ für } i > 0 \\ \Sigma^* & := & \bigcup_{i \in \mathbb{N}} \Sigma^i \\ \Sigma^+ & := & \bigcup_{i \in \mathbb{N}_{>0}} \Sigma^i \end{array}$$

Beachte: $\mathbb{N} = \{0, 1, 2, \ldots\}$ und $\mathbb{N}_{>0} = \{1, 2, \ldots\}$

Beispiele

Sei $\Sigma = \{a, b\}$.

Dann ist

$\Sigma^0 = \{\varepsilon\}$
$\Sigma^1 = \Sigma = \{a, b\}$
$\Sigma^2 = \{aa, ab, ba, bb\}$
$\Sigma^3 = \{\mathsf{a}w \mid \mathsf{a} \in \{a, b\}, w \in \Sigma^2\} = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$
$\ldots$
und
$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb, aaaa, \ldots\}$

Weitere Notationen und Begriffe

Sei w ein Wort über Σ

- w^m entsteht aus m-maligen Konkatenieren von w, d.h.

$$w^0 = \varepsilon \text{ und } w^m = ww^{m-1} \text{ für } m > 0$$

- $\overline{w}$ ist das rückwärts gelesene Wort w, d.h.

$$\overline{\varepsilon} = \varepsilon \text{ und für } w = a_1 \cdots a_n \text{ ist } \overline{w} = a_n a_{n-1} \cdots a_1$$

- w ist ein Palindrom g.d.w. $w = \overline{w}$

Beispiele für Palindrome:

anna, reliefpfeiler, lagerregal, annasusanna

Sprechweisen: Präfix, Suffix, Teilwort

Seien u, v Wörter über einem Alphabet Σ.

- u ist ein **Präfix** von v, wenn es ein Wort w gibt mit $uw = v$.

$$\underbrace{\boxed{u}\boxed{w}}_{v}$$

- u ist ein **Suffix** von v, wenn es ein Wort w gibt mit $wu = v$.

$$\underbrace{\boxed{w}\boxed{u}}_{v}$$

- u ist ein **Teilwort** von v, wenn es Wörter w_1, w_2 gibt mit $w_1 u w_2 = v$.

$$\underbrace{\boxed{w_1}\boxed{u}\boxed{w_2}}_{v}$$

Beispiel: Sei $w = ababbaba$

- aba ist ein Präfix, Suffix und Teilwort von w
- $ababb$ ist ein Präfix (und Teilwort) von w, aber kein Suffix von w
- bab ist Teilwort von w, aber weder ein Präfix noch ein Suffix

Formale Sprache

Formale Sprache

Eine (formale) Sprache L über dem Alphabet Σ ist eine Teilmenge von Σ^* d.h. $L \subseteq \Sigma^*$

Beachte: (wir verwenden L für „language“)

Operationen auf formalen Sprachen

Seien L, L_1, L_2 formale Sprachen über Σ

- Vereinigung: $L_1 \cup L_2 := \{w \mid w \in L_1 \text{ oder } w \in L_2\}$
- Schnitt: $L_1 \cap L_2 := \{w \mid w \in L_1 \text{ und } w \in L_2\}$
- Komplement zu L: $\overline{L} := \Sigma^* \setminus L$
- Produkt: $L_1 L_2 = L_1 \circ L_2 = \{uv \mid u \in L_1 \text{ und } v \in L_2\}$

Beispiele

Sei $\Sigma = \{a, b\}$ und $L_1 = \{a^i \mid i \in \mathbb{N}\}$ $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

- $L_1 \cup L_2 =$ Sprache aller Wörter, die nur aus a's oder nur aus b's bestehen
- $L_1 \cap L_2 = \{\varepsilon\}$
- $\overline{L_1}$ ist die Sprache der Worte, die mindestens ein b enthalten
- $L_1 L_2 = \{a^i b^j \mid i, j \in \mathbb{N}\}$,
- $L_2 L_1 = \{b^i a^j \mid i, j \in \mathbb{N}\}$
- $L_1 L_1 = L_1$.

Für $L_1 = \{\spadesuit, \clubsuit, \diamondsuit, \heartsuit\}$ und $L_2 = \{7, 8, 9, 10, J, D, K, A\}$ stellt $L_1 L_2$ eine Repräsentation der Spielkarten eines Skatblatts dar.

Kleenescher Abschluss

Sei L eine Sprache. Dann ist:

$$L^0 := \{\varepsilon\} \qquad L^* := \bigcup_{i \in \mathbb{N}} L^i$$

$$L^i := L \circ L^{i-1} \text{ für } i > 0 \qquad L^+ := \bigcup_{i \in \mathbb{N}_{>0}} L^i$$

Die Sprache L^* nennt man auch den Kleeneschen Abschluss von L (benannt nach Stephen Cole Kleene).

Beispiel: $L = \{ab, ac\}$

- $L^0 = \{\varepsilon\}$
- $L^1 = L \circ L^0 = L = \{ab, ac\}$
- $L^2 = \{abab, abac, acab, acac\}$
- $L^3 = \{ababab, ababac, abacab, abacac, acabab, acabac, acacab, acacac\}$
- $L^* = \{\varepsilon\} \cup \{ax_1ax_2 \cdots ax_i \mid i \in \mathbb{N}_{>0}, x_j \in \{b, c\}, j = 1, \ldots, i\}$.

Weitere Beispiele

$$((\{\varepsilon, 1\} \circ \{0, \ldots, 9\}) \cup (\{2\} \circ \{0, 1, 2, 3\})) \\ \circ \{:\} \circ \{0, 1, 2, 3, 4, 5\} \circ \{0, \ldots, 9\}$$

Beschriebene Sprache = ?

$$\{0\} \cup (\{1, \ldots, 9\} \circ \{0, \ldots, 9\}^*)$$

Beschriebene Sprache = ?

Weitere Beispiele

$$((\{\varepsilon, 1\} \circ \{0, \ldots, 9\}) \cup (\{2\} \circ \{0, 1, 2, 3\})) \\ \circ \{:\} \circ \{0, 1, 2, 3, 4, 5\} \circ \{0, \ldots, 9\}$$

Beschriebene Sprache = Sprache aller gültigen Uhrzeiten

$$\{0\} \cup (\{1, \ldots, 9\} \circ \{0, \ldots, 9\}^*)$$

Beschriebene Sprache = ?

Weitere Beispiele

$$((\{\varepsilon,1\}\circ\{0,\ldots,9\})\cup(\{2\}\circ\{0,1,2,3\}))$$
$$\circ\{:\}\circ\{0,1,2,3,4,5\}\circ\{0,\ldots,9\}$$

Beschriebene Sprache = Sprache aller gültigen Uhrzeiten

$$\{0\}\cup(\{1,\ldots,9\}\circ\{0,\ldots,9\}^*)$$

Beschriebene Sprache = Sprache aller natürlichen Zahlen

Formale Sprachen darstellen

- Sei Σ ein Alphabet.
- Eine Sprache über Σ ist eine Teilmenge von Σ^*.
- 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

Grammatiken

Grammatik für einen sehr kleinen Teil der deutschen Sprache:

$$
\begin{array}{ll}
<\text{Satz}> & \rightarrow <\text{Subjekt}><\text{Prädikat}><\text{Objekt}> \\
<\text{Subjekt}> & \rightarrow <\text{Artikel}><\text{Attribut}><\text{Nomen}> \\
<\text{Objekt}> & \rightarrow <\text{Artikel}><\text{Attribut}><\text{Nomen}> \\
<\text{Artikel}> & \rightarrow \varepsilon \\
<\text{Artikel}> & \rightarrow \text{der} \\
<\text{Artikel}> & \rightarrow \text{das} \\
<\text{Attribut}> & \rightarrow <\text{Adjektiv}> \\
<\text{Attribut}> & \rightarrow <\text{Adjektiv}><\text{Attribut}> \\
<\text{Adjektiv}> & \rightarrow \text{kleine} \\
<\text{Adjektiv}> & \rightarrow \text{große} \\
<\text{Adjektiv}> & \rightarrow \text{nette} \\
<\text{Adjektiv}> & \rightarrow \text{blaue} \\
<\text{Nomen}> & \rightarrow \text{Mann} \\
<\text{Nomen}> & \rightarrow \text{Auto} \\
<\text{Prädikat}> & \rightarrow \text{fährt} \\
<\text{Prädikat}> & \rightarrow \text{liebt}
\end{array}
$$

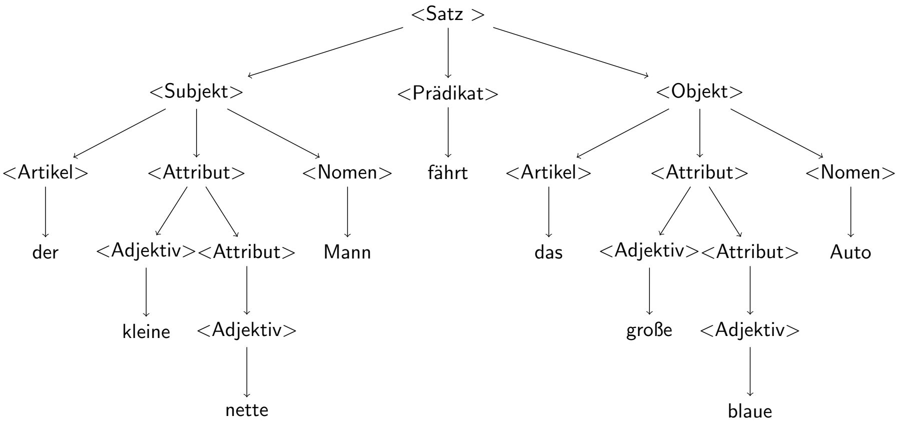
Grammatiken

- Endliche Menge von Regeln „linke Seite$\to$ rechte Seite”
- Symbole in spitzen Klammern wie <Artikel> sind Variablen, d.h. sie sind Platzhalter, die weiter ersetzt werden müssen.
- Z.B. kann

  „der kleine nette Mann fährt das große blaue Auto“

  durch die obige Grammatik abgeleitet werden,

Syntaxbaum zum Beispiel



Definition einer Grammatik

Definition (Grammatik)

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

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

Oft genügt es, P alleine zu notieren
(wenn klar ist, was Variablen, Zeichen und Startsymbol sind)

Beispiel für eine Grammatik

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

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

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

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

Ableitung

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

Ableitungsschritt $\Rightarrow_G$

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

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

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

$\Rightarrow_G^*$ sei die reflexiv-transitive Hülle von $\Rightarrow_G$

Ableitung

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

Beispiel

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

$$P = \{ \begin{array}{llllllllllll} E & \to & M, & E & \to & E+M, & M & \to & Z, & M & \to & M*Z, \\ Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \end{array} \}$$

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

$$
\begin{array}{llllllllllll}
P = \{E & \to & M, & E & \to & E + M, & M & \to & Z, & M & \to & M * Z, \\
Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \}
\end{array}
$$

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

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

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

$$P = \{ \begin{array}{llllllllllll} E & \to & M, & E & \to & E+M, & M & \to & Z, & M & \to & M*Z, \\ Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \end{array} \}$$

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

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

Beispiel

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

$$\begin{aligned} P = \{ & E \to M, \quad E \to E + M, \quad M \to Z, \quad M \to M * Z, \\ & Z \to 1, \quad Z \to 2, \quad Z \to (E) \quad \} \end{aligned}$$

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

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

Beispiel

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

$$
\begin{aligned}
P = \{ E &\to M, & E &\to E + M, & M &\to Z, & M &\to M * Z, \\
Z &\to 1, & Z &\to 2, & {\color{red}Z} &\to {\color{green}(E)} & & \}
\end{aligned}
$$

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

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

Beispiel

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

$$\begin{array}{llllllllllll} P = \{E & \to & M, & E & \to & E + M, & M & \to & Z, & M & \to & M * Z, \\ Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \} \end{array}$$

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

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

Beispiel

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

$$
\begin{array}{llllllllllll}
P = \{E & \to & M, & E & \to & E + M, & M & \to & Z, & M & \to & M * Z, \\
Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \}
\end{array}
$$

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

$$
\begin{array}{l}
E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\
\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z)
\end{array}
$$

Beispiel

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

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

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

$$
\begin{array}{l}
E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M) \\
\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z) \\
\Rightarrow (M + M) * (E + Z)
\end{array}
$$

Beispiel

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

$$\begin{aligned} P = \{ & E \to M, && E \to E + M, && M \to Z, && M \to M * Z, \\ & Z \to 1, && Z \to 2, && Z \to (E) && \} \end{aligned}$$

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

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

Beispiel

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

$$
\begin{aligned}
P = \{E &\to M, & E &\to E + M, & M &\to Z, & M &\to M * Z,\\
Z &\to 1, & Z &\to 2, & Z &\to (E) & & \}
\end{aligned}
$$

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

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

Beispiel

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

$$
\begin{array}{rlllllllllll}
P = \{E & \to & M, & E & \to & E + M, & M & \to & Z, & M & \to & M * Z, \\
Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \}
\end{array}
$$

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

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

Beispiel

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

$$\begin{array}{rlrlrlrl} P = \{E & \to M, & E & \to E + M, & M & \to Z, & M & \to M * Z, \\ Z & \to 1, & Z & \to 2, & Z & \to (E) & & \} \end{array}$$

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

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

Beispiel

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

$$
\begin{array}{llllllllllll}
P = \{E & \to & M, & E & \to & E + M, & M & \to & Z, & M & \to & M * Z, \\
Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \}
\end{array}
$$

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

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

Beispiel

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

$$\begin{array}{llllllllllll} P = \{ E & \to & M, & E & \to & E+M, & M & \to & Z, & M & \to & M*Z, \\ Z & \to & 1, & Z & \to & 2, & Z & \to & (E) & & & \} \end{array}$$

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

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

Beispiel

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

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

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

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

Beispiel

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

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

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

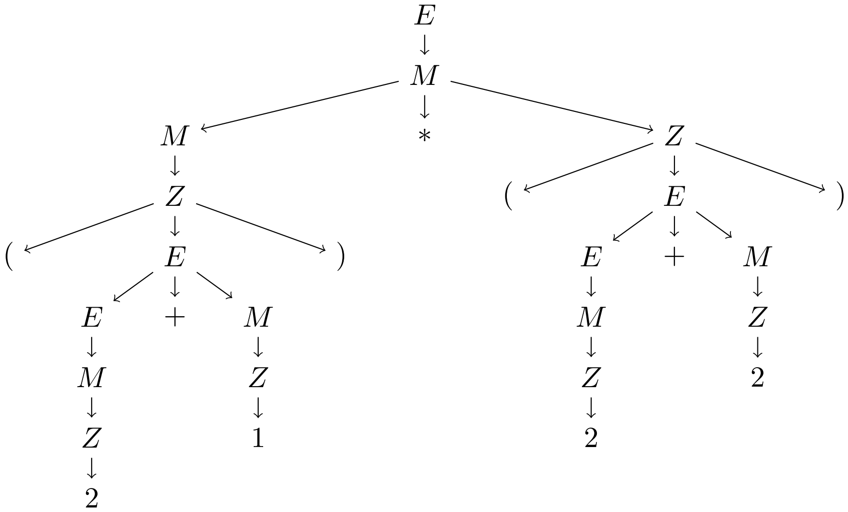
Ableitung von letzter Folie (keine Linksableitung):

$E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow Z * (E) \Rightarrow Z * (E + M)$
$\Rightarrow (E) * (E + M) \Rightarrow (E) * (E + Z) \Rightarrow (E + M) * (E + Z)$
$\Rightarrow (M + M) * (E + Z) \Rightarrow (M + M) * (M + Z)$
$\Rightarrow (M + M) * (Z + Z) \Rightarrow (M + M) * (Z + 2)$
$\Rightarrow (M + Z) * (Z + 2) \Rightarrow (M + Z) * (2 + 2)$
$\Rightarrow (Z + Z) * (2 + 2) \Rightarrow (2 + Z) * (2 + 2)$
$\Rightarrow (2 + 1) * (2 + 2)$

Linksableitung: ersetzt immer das linkeste Nichtterminal

$E \Rightarrow M \Rightarrow M * Z \Rightarrow Z * Z \Rightarrow (E) * Z$
$\Rightarrow (E + M) * Z \Rightarrow (M + M) * Z \Rightarrow (Z + M) * Z$
$\Rightarrow (2 + M) * Z \Rightarrow (2 + Z) * Z \Rightarrow (2 + 1) * Z \Rightarrow (2 + 1) * (E)$
$\Rightarrow (2 + 1) * (E + M) \Rightarrow (2 + 1) * (M + M) \Rightarrow (2 + 1) * (Z + M)$
$\Rightarrow (2 + 1) * (2 + M) \Rightarrow (2 + 1) * (2 + Z)$
$\Rightarrow (2 + 1) * (2 + 2)$

Syntaxbaum (zu beiden Ableitungen)



Nichtdeterminismus beim Ableiten

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

Quellen des Nichtdeterminismus:

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

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

Erzeugte Sprache

Erzeugte Sprache einer Grammatik

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

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

Beispiele

$G_1 = (\{S\}, \{a\}, \{S \to aS\}, S)$
$L(G_1) = \textbf{?}$

$G_2 = (\{S'\}, \{a, b\}, \{S' \to aS', S' \to b\}, S')$
$L(G_2) = \textbf{?}$

Beispiele

$G_1 = (\{S\}, \{a\}, \{S \to aS\}, S)$
$L(G_1) = \textbf{?}$

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

$G_2 = (\{S'\}, \{a, b\}, \{S' \to aS', S' \to b\}, S')$
$L(G_2) = \textbf{?}$

Beispiele

$G_1 = (\{S\}, \{a\}, \{S \to aS\}, S)$
$L(G_1) = \emptyset$

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

$G_2 = (\{S'\}, \{a, b\}, \{S' \to aS', S' \to b\}, S')$
$L(G_2) = $ **?**

Beispiele

$G_1 = (\{S\}, \{a\}, \{S \to aS\}, S)$
$L(G_1) = \emptyset$

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

$G_2 = (\{S'\}, \{a, b\}, \{S' \to aS', S' \to b\}, S')$
$L(G_2) =$ **?**

- $$\begin{tikzcd} S' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aaS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aaaS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aaaaS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \cdots \\ b & ab & aab & aaab & aaaab & \end{tikzcd}$$
- Für alle $i \in \mathbb{N}$ gilt $S \Rightarrow^i a^i S \Rightarrow a^i b$

Beispiele

$G_1 = (\{S\}, \{a\}, \{S \to aS\}, S)$
$L(G_1) = \emptyset$

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

$G_2 = (\{S'\}, \{a, b\}, \{S' \to aS', S' \to b\}, S')$
$L(G_2) = \{a^n b \mid n \in \mathbb{N}\}$

- $$\begin{tikzcd} S' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aaS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aaaS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & aaaaS' \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \cdots \\ b & ab & aab & aaab & aaaab & \end{tikzcd}$$
- Für alle $i \in \mathbb{N}$ gilt $S \Rightarrow^i a^i S \Rightarrow a^i b$

Beispiel (kontextsensitive Grammatik)

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

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

Beispiel-Ableitung:

$$\begin{aligned} S &\Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaSBCBCBC \Rightarrow aaaaBCBCBCBC \\ &\Rightarrow aaaabCBCBCBC \Rightarrow aaaabBCCBCBC \Rightarrow aaaabbCCBCBC \\ &\Rightarrow aaaabbCBCCBC \Rightarrow aaaabbBCCCBC \Rightarrow aaaabbBCCBCC \\ &\Rightarrow aaaabbBCBCCC \Rightarrow aaaabbBBCCCC \Rightarrow aaaabbbBCCCC \\ &\Rightarrow aaaabbbbCCCC \Rightarrow aaaabbbbcCCC \Rightarrow aaaabbbbccCC \\ &\Rightarrow aaaabbbbcccC \Rightarrow aaaabbbbcccc \end{aligned}$$

Steckengebliebene Folge von Ableitungsschritten:

$$S \Rightarrow aSBC \Rightarrow aaBCBC \Rightarrow aabCBC \Rightarrow aabcBC$$

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\supseteq$“: Zeige $a^n b^n c^n \in L(G)$ für alle $n \in \mathbb{N}_{>0}$

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

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\supseteq$“: Zeige $a^n b^n c^n \in L(G)$ für alle $n \in \mathbb{N}_{>0}$

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

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\supseteq$“: Zeige $a^n b^n c^n \in L(G)$ für alle $n \in \mathbb{N}_{>0}$

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

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\supseteq$“: Zeige $a^n b^n c^n \in L(G)$ für alle $n \in \mathbb{N}_{>0}$

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

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\supseteq$“: Zeige $a^n b^n c^n \in L(G)$ für alle $n \in \mathbb{N}_{>0}$

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

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\supseteq$“: Zeige $a^n b^n c^n \in L(G)$ für alle $n \in \mathbb{N}_{>0}$

- Wende $n-1$ mal $S \to aSBC$ und dann einmal $S \to aBC$ an:
  $S \Rightarrow^* a^{n-1}S(BC)^{n-1} \Rightarrow a^n(BC)^n$
- Wende $CB \to BC$ solange an, bis es kein Teilwort CB mehr gibt.
  $a^n(BC)^n \Rightarrow^* a^n B^n C^n$
- Wende $aB \to ab$ und anschließend $n-1$ mal $bB \to bb$ 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 $bC \to bc$ und anschließend $n-1$ mal $cC \to cc$ 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, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt (2)

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\subseteq$“: Zeige, dass alle von G erzeugten Worte 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 $aB \to ab$ erzeugt wird und die Regeln keine Terminalsymbole vertauschen.
- Ebenso kann das linkeste c nur durch die Regel $bC \to bc$ erzeugt sein, daher: $S \Rightarrow^* a^n w_1' bCw_2 \Rightarrow a^n w_1' bcw_2 \Rightarrow^* a^n w'$.
- b's werden nur durch $aB \to ab$ und $bB \to bb$ erzeugt.
  Da auf w_1' ein b folgt, muss w_1' nur aus b's bestehen

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt (2)

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\subseteq$“: Zeige, dass alle von G erzeugten Worte 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 $aB \to ab$ erzeugt wird und die Regeln keine Terminalsymbole vertauschen.
- Ebenso kann das linkeste c nur durch die Regel $bC \to bc$ erzeugt sein, daher: $S \Rightarrow^* a^n w_1' bC w_2 \Rightarrow a^n w_1' bc w_2 \Rightarrow^* a^n w'$.
- b's werden nur durch $aB \to ab$ und $bB \to bb$ erzeugt.
  Da auf w_1' ein b folgt, muss w_1' nur aus b's bestehen

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt (2)

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\subseteq$": Zeige, dass alle von G erzeugten Worte 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 $aB \to ab$ erzeugt wird und die Regeln keine Terminalsymbole vertauschen.
- Ebenso kann das linkeste c nur durch die Regel $bC \to bc$ erzeugt sein, daher: $S \Rightarrow^* a^n w'_1 bCw_2 \Rightarrow a^n w'_1 bcw_2 \Rightarrow^* a^n w'$.
- b's werden nur durch $aB \to ab$ und $bB \to bb$ erzeugt.
  Da auf w'_1 ein b folgt, muss w'_1 nur aus b's bestehen

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt (2)

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\subseteq$": Zeige, dass alle von G erzeugten Worte 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 $aB \to ab$ erzeugt wird und die Regeln keine Terminalsymbole vertauschen.
- Ebenso kann das linkeste c nur durch die Regel $bC \to bc$ erzeugt sein, daher: $S \Rightarrow^* a^n w_1' bCw_2 \Rightarrow a^n w_1' bcw_2 \Rightarrow^* a^n w'$.
- b's werden nur durch $aB \to ab$ und $bB \to bb$ erzeugt.
  Da auf w_1' ein b folgt, muss w_1' nur aus b's bestehen

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt (2)

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

„$\subseteq$": Zeige, dass alle von G erzeugten Worte 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 $aB \to ab$ erzeugt wird und die Regeln keine Terminalsymbole vertauschen.
- Ebenso kann das linkeste c nur durch die Regel $bC \to bc$ erzeugt sein, daher: $S \Rightarrow^* a^n w_1' bCw_2 \Rightarrow a^n w_1' bcw_2 \Rightarrow^* a^n w'$.
- b's werden nur durch $aB \to ab$ und $bB \to bb$ erzeugt.
  Da auf w_1' ein b folgt, muss w_1' nur aus b's bestehen

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt (2)

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ 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 Worte 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.
- Ebenso kann das linkeste c nur durch die Regel $bC \rightarrow bc$ erzeugt sein, daher: $S \Rightarrow^* a^n w_1' bCw_2 \Rightarrow a^n w_1' bcw_2 \Rightarrow^* a^n w'$.
- b's werden nur durch $aB \rightarrow ab$ und $bB \rightarrow bb$ erzeugt.
  Da auf w_1' ein b folgt, muss w_1' nur aus b's bestehen

Grammatik, die $\{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ erzeugt (3)

Satz

$L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ für $G = (\{S, B, C\}, \{a, b, c\}, P, S)$ mit
$P = \{S \to aSBC, S \to aBC, CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}$

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

- ...
- $S \Rightarrow^* a^n\{b\}^* c w_2 \Rightarrow^* a^n w'$.
- w_2 kann keine Vorkommen von B enthalten: Sonst könnte man (aufgrund des c vor w_2) kein Terminalwort w erzeugen: Vorkommen von B könnten nicht mehr ersetzt werden.
- Daher gilt $a^n b c w_2 = a^n\{b\}^* c\{C, c\}^*$ und daher $a^n w = a^n b^n c^n$ (da nur noch $cC \to cc$ angewendet werden kann).

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

S

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

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

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

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

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

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

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

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

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

$$\begin{aligned} S & \Rightarrow \$T\$ \Rightarrow \$aAT\$ \Rightarrow \$aAaAT\$ \Rightarrow \$aAaAbBT\$ \\ & \Rightarrow \$aAaAbB\$ \end{aligned}$$

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

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

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

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

Beispiel einer allgemeinen Grammatik

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

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

Eine Ableitung:

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

Beispiel einer allgemeinen Grammatik

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

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

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 allgemeinen Grammatik

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

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

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 allgemeinen Grammatik

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

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

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 allgemeinen Grammatik

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

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

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 allgemeinen Grammatik

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

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

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 allgemeinen Grammatik

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

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

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 allgemeinen Grammatik

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

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

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 allgemeinen Grammatik (2)

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

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

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

Beispiel einer allgemeinen Grammatik (2)

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

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

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

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

Beispiel einer allgemeinen Grammatik (2)

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

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

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

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

Beispiel einer allgemeinen Grammatik (2)

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

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

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

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

Beispiel einer allgemeinen Grammatik (2)

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

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

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

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

Beispiel einer allgemeinen Grammatik (2)

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

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

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

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

Beispiel einer allgemeinen Grammatik (2)

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

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

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

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

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.

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

für alle $(\ell \to r) \in P$: $|\ell| \leq |r|$.

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

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

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

G ist vom Typ 2 und für alle $(A \to r) \in P$ gilt: $r = a$ oder $r = aA'$ für $a \in \Sigma, A' \in V$ (die rechten Seiten sind Worte 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.

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

für alle $(\ell \to r) \in P$: $|\ell| \leq |r|$.

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

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

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

G ist vom Typ 2 und für alle $(A \to r) \in P$ gilt: $r = a$ oder $r = aA'$ für $a \in \Sigma, A' \in V$ (die rechten Seiten sind Worte 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.

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

für alle $(\ell \to r) \in P$: $|\ell| \leq |r|$.

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

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

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

G ist vom Typ 2 und für alle $(A \to r) \in P$ gilt: $r = a$ oder $r = aA'$ für $a \in \Sigma, A' \in V$ (die rechten Seiten sind Worte 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.

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

für alle $(\ell \to r) \in P$: $|\ell| \leq |r|$.

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

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

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

G ist vom Typ 2 und für alle $(A \to r) \in P$ gilt: $r = a$ oder $r = aA'$ für $a \in \Sigma, A' \in V$ (die rechten Seiten sind Worte aus $(\Sigma \cup (\Sigma V))$)

Typ i-Sprachen

Definition

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

Beispiele

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

Beispiele

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

Beispiele

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

Beispiele

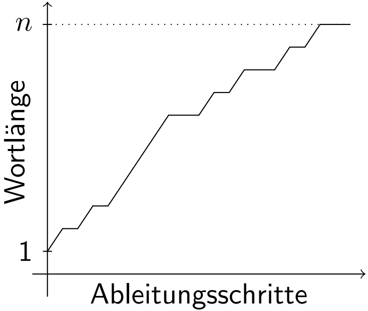
- $G_1 = (\{S\}, \{a\}, \{S \to aS, S \to b\}, S)$ ist regulär (Typ 3)
- $G_2 = (\{E, M, Z\}, \{+, *, 1, 2, (,)\}, P, E)$ mit
  $P = \{E \to M,\ \ E \to E + M,\ \ M \to Z,\ \ M \to M * Z,$
  $\qquad Z \to 1,\ \ Z \to 2,\ \ Z \to (E)\}$ ist kontextfrei (Typ 2)
- $G_3 = (\{S, B, C\}, \{a, b, c\}, P, S)$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

Kontextfrei vs. kontextsensitiv

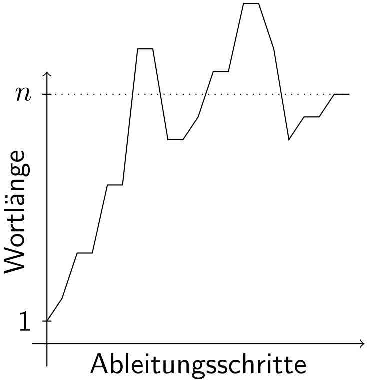
- Kontextfreie Produktionen $A \to r$ sind immer auf ein Vorkommen von A anwendbar.
- Kontextsensitive Produktionen können solche Ersetzungen auf einen Kontext einschränken und erlauben Regeln $uAv \to urv$, die die Ersetzung von A durch r nur erlauben, wenn A durch u und v umrahmt ist.

Typ 0 vs. Typ 1

Ableitung eines Wortes der Länge n



mit Typ 1-Grammatik



mit Typ 0-Grammatik

Syntaxbäume

Definition (Syntaxbaum)

Sei $G = (V, \Sigma, P, S)$ eine Typ 2-Grammatik und $S \Rightarrow w_0 \Rightarrow \ldots \Rightarrow w_n$ eine Ableitung von $w_n \in \Sigma^*$. Der **Syntaxbaum** zur Ableitung wird wie folgt erstellt:

- Die Wurzel des Baums ist mit S markiert.
- Wenn $w_i \Rightarrow w_{i+1}$ und $w_i = uAv$ und $w_{i+1} = urv$ (die angewandte Produktion ist $A \to r$), dann erzeuge im Syntaxbaum $|r|$ viele Knoten als Kinder des mit A markierten Knotens (an der passenden Stelle im Syntaxbaum). 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:

$$
\begin{tikzcd}
S \arrow[d] \arrow[dr] & & & & & \\
a_1 & A_1 \arrow[d] \arrow[dr] & & & & \\
& a_2 & A_2 \arrow[d] \arrow[dr] & & & \\
& & a_3 & A_3 \arrow[dr, dotted, dash] & & \\
& & & & A_{n-1} \arrow[d] \arrow[dr] & \\
& & & & a_n & A_n \arrow[d] \\
& & & & & a_{n+1}
\end{tikzcd}
$$

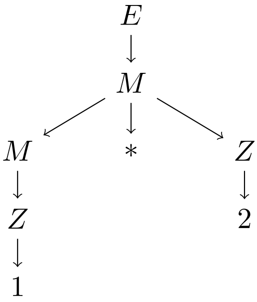
Beispiel

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

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.



Links- und Rechtsableitungen

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

Beispiele:

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

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

Links- und Rechtsableitungen (2)

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

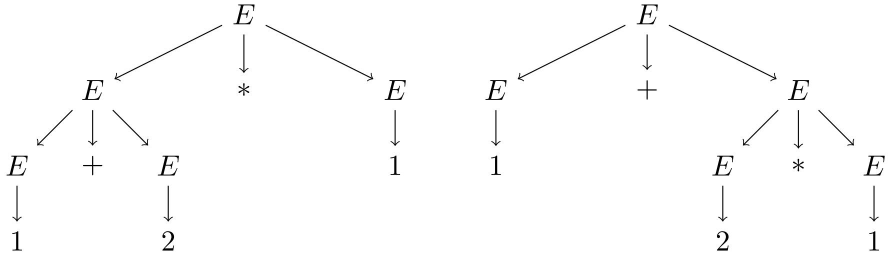
Beispiel:

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

Zwei Ableitungen für $1 + 2 * 1$:

- $\color{red}{E} \Rightarrow \color{red}{E} * E \Rightarrow \color{red}{E} + E * E \Rightarrow 1 + \color{red}{E} * E \Rightarrow 1 + 2 * \color{red}{E} \Rightarrow 1 + 2 * 1$
- $\color{red}{E} \Rightarrow E + \color{red}{E} \Rightarrow \color{red}{E} + E * E \Rightarrow 1 + \color{red}{E} * E \Rightarrow 1 + 2 * \color{red}{E} \Rightarrow 1 + 2 * 1.$

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

$$\{a^m b^m c^n d^n \mid m,n \in \mathbb{N}_{>0}\} \cup \{a^m b^n c^n d^m \mid m,n \in \mathbb{N}_{>0}\}$$

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

Backus-Naur-Form

Erweiterte Backus-Naur-Form (EBNF)

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

1. Statt $A \to w_1, A \to w_2, \ldots A \to w_n$ schreiben wir auch $A \to w_1 \mid w_2 \mid \ldots \mid w_n$.
2. Die Schreibweise $A \to u[v]w$ steht für die beiden Produktionen $A \to uvw$ und $A \to uw$ (d. h. $[v]$ meint, dass v optional ist).
3. Die Schreibweise $A \to u\{v\}w$ steht für $A \to uw$ oder $A \to uBw$ mit $B \to v \mid vB$ (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)

ε-Regel für Typ 1,2,3-Grammatiken

- Das leere Wort ε kann bisher nicht für Typ 1,2,3 Grammatiken erzeugt werden:

  Produktion $S \to \varepsilon$ erfüllt die Typ 1-Bedingung $|S| \leq |\varepsilon|$ nicht.

ε-Regel für Typ 1,2,3-Grammatiken

- Das leere Wort ε kann bisher nicht für Typ 1,2,3 Grammatiken erzeugt werden:

  Produktion $S \to \varepsilon$ erfüllt die Typ 1-Bedingung $|S| \leq |\varepsilon|$ nicht.
- Daher Sonderregel:

ε-Regel für Typ 1,2,3-Grammatiken

Eine Grammatik $G = (V, \Sigma, P, S)$ vom Typ 1, 2 oder 3 darf eine Produktion $(S \to \varepsilon) \in P$ enthalten, vorausgesetzt, dass keine rechte Seite einer Produktion in P, die Variable S enthält.

Sonderregel erlaubt nicht:

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

Sonderregel erlaubt:

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

Das leere Wort kann stets hinzugefügt werden

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
$G' = (V \cup \{S'\}, \Sigma, P \cup \{S' \to \varepsilon\} \cup \{S' \to r \mid S \to r \in P\}, S')$ die Sprache $L(G') = L(G) \cup \{\varepsilon\}$, G' erfüllt die ε-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 $S \to r \in P$ vom Typ i, sind auch $S' \to r$ 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. $S' \Rightarrow_{G'} r$

ε-Produktionen für Typ 2- und Typ 3-Grammatiken

Sonderregel für Typ 2- und Typ 3-Grammatiken:

ε-Produktionen in kontextfreien und regulären Grammatiken

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

Das ist keine echte Erweiterung, denn:

Satz (Entfernen von ε-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 ε-Produktionen.

Beweis: Algorithmus auf der nächsten Folie.

Algorithmus 1: Entfernen von ε-Produktionen

Eingabe: Typ i-Grammatik $G = (V, \Sigma, P, S)$ mit $\varepsilon \notin L(G)$, $i \in \{2, 3\}$
Ausgabe: Typ i-Grammatik G' ohne ε-Produktionen, sodass $L(G) = L(G')$
Beginn
    finde die Menge $W \subseteq V$ aller Variablen A für die gilt $A \Rightarrow^* \varepsilon$:
    Beginn
        $W := \{A \mid (A \to \varepsilon) \in P\}$;
        wiederhole
            füge alle A zu W hinzu mit $A \to A_1 \ldots A_n \in P$ und $\forall i : A_i \in W$;
        bis *sich W nicht mehr ändert*;
    Ende
    $P' := P \setminus \{A \to \varepsilon \mid (A \to \varepsilon) \in P\}$;   /* lösche Regeln $A \to \varepsilon$ */
    wiederhole
        für alle *Produktionen $A' \to uAv \in P'$ mit $|uv| > 0$ und $A \in W$* **tue**
            füge die Produktion $A' \to uv$ zu P' hinzu;
            /* für $A' \to u'Av'Aw'$ gibt es (mindestens) zwei Hinzufügungen: Für das Vorkommen von A nach u' als auch für das Vorkommen direkt vor w' */
        Ende
    bis *sich P' nicht mehr ändert*;
    gebe $G' = (V, \Sigma, P', S)$ als Ergebnisgrammatik aus;
Ende

Algorithmus 1: Entfernen von ε-Produktionen

Eingabe: Typ i-Grammatik $G = (V, \Sigma, P, S)$ mit $\varepsilon \notin L(G)$, $i \in \{2, 3\}$
Ausgabe: Typ i-Grammatik G' ohne ε-Produktionen, sodass $L(G) = L(G')$
Beginn
  finde die Menge $W \subseteq V$ aller Variablen A für die gilt $A \Rightarrow^* \varepsilon$:
  Beginn
    $W := \{A \mid (A \to \varepsilon) \in P\}$;
    wiederhole
      füge alle A zu W hinzu mit $A \to A$
    bis *sich* W *nicht mehr ändert*;
  Ende
  $P' := P \setminus \{A \to \varepsilon \mid (A \to \varepsilon) \in P\}$; /* lösche Regeln $A \to \varepsilon$ */
  wiederhole
    für alle *Produktionen* $A' \to uAv \in P'$ *mit* $|uv| > 0$ *und* $A \in W$ **tue**
      füge die Produktion $A' \to uv$ zu P' hinzu;
      /* für $A' \to u'Av'Aw'$ gibt es (mindestens) zwei Hinzufügungen: Für das Vorkommen von A nach u' als auch für das Vorkommen direkt vor w' */
    Ende
  bis *sich* P' *nicht mehr ändert*;
  gebe $G' = (V, \Sigma, P', S)$ als Ergebnisgrammatik aus;
Ende

Die neuen Produktionen nehmen den Ableitungsschritt $A \to \varepsilon$ vorweg.

Für reguläre Produktion $A' \to aA$ wird $A' \to a$ hinzugefügt (bleibt regulär!)

Beispiel: Entfernen von ε-Produktionen

$G = (\{A, B, C, D, S\}, \{0, 1\}, P, S)$ mit

$$P = \{S \to 1A,\ \ A \to AB,\ \ A \to DA,\ \ A \to \varepsilon,\ \ B \to 0,$$
$$B \to 1,\ \ C \to AAA,\ \ D \to 1AC\}.$$

Beispiel: Entfernen von ε-Produktionen

$G = (\{A, B, C, D, S\}, \{0, 1\}, P, S)$ mit

$$P = \{S \to 1A,\ \ A \to AB,\ \ A \to DA,\ \ A \to \varepsilon,\ \ B \to 0,\\ B \to 1,\ \ C \to AAA,\ \ D \to 1AC\}.$$

- Menge W der Variablen, die ε herleiten:

  $W = \{A, C\}$ da $A \to \varepsilon$ und $C \to AAA$

Beispiel: Entfernen von ε-Produktionen

$G = (\{A, B, C, D, S\}, \{0, 1\}, P, S)$ mit

$$P = \{S \to 1A,\ \ A \to AB,\ \ A \to DA,\ \ A \to \varepsilon,\ \ B \to 0,\\ B \to 1,\ \ C \to AAA,\ \ D \to 1AC\}.$$

- Menge W der Variablen, die ε herleiten:

  $W = \{A, C\}$ da $A \to \varepsilon$ und $C \to AAA$
- Starte mit

  $$P' = \{S \to 1A,\ \ A \to AB,\ \ A \to DA,\ \ B \to 0,\\ B \to 1,\ \ C \to AAA,\ \ D \to 1AC\}.$$

Beispiel: Entfernen von ε-Produktionen

$G = (\{A, B, C, D, S\}, \{0, 1\}, P, S)$ mit

$$P = \{S \to 1A,\ \ A \to AB,\ \ A \to DA,\ \ A \to \varepsilon,\ \ B \to 0,\\ B \to 1,\ \ C \to AAA,\ \ D \to 1AC\}.$$

- Menge W der Variablen, die ε herleiten:

  $W = \{A, C\}$ da $A \to \varepsilon$ und $C \to AAA$
- Starte mit

  $$P' = \{S \to 1A,\ \ A \to AB,\ \ A \to DA,\ \ B \to 0,\\ B \to 1,\ \ C \to AAA,\ \ D \to 1AC\}.$$
- Hinzufügen von Produktionen für Vorkommen von A und C

  $$P' = \{S \to 1A,\ \ S \to 1,\ \ A \to AB,\ \ A \to B,\ \ A \to DA,\ \ A \to D,\\ B \to 0,\ \ B \to 1,\ \ C \to AAA,\ \ C \to AA,\ \ C \to A,\\ D \to 1AC,\ \ D \to 1A,\ \ D \to 1C,\ \ D \to 1\}.$$

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

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

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

- $L = \{a^n b^n \mid n \in \mathbb{N}_{>0}\}$ ist von Typ 2, aber nicht von Typ 3
- $L = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ ist von Typ 1, aber nicht von Typ 2.
- $H = \{w\$x \mid$ Turingmaschine M_w hält für 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.

Abgeschlossenheit von Sprachen

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 $\overline{L} \in \mathcal{L}$ 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

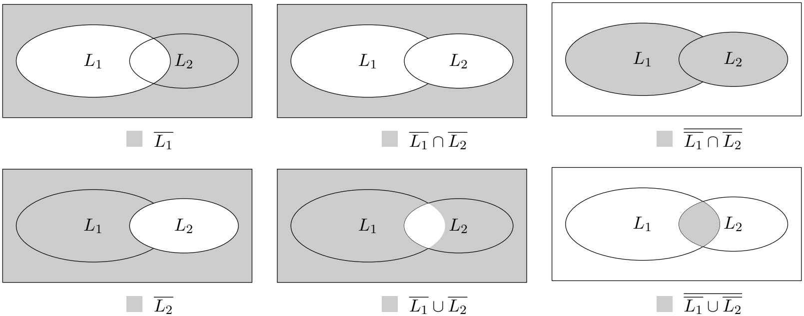
Abgeschlossenheit: Eigenschaften

Satz

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

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

Venn-Diagramme dazu:



Entscheidbarkeit

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.

Entscheidbarkeit

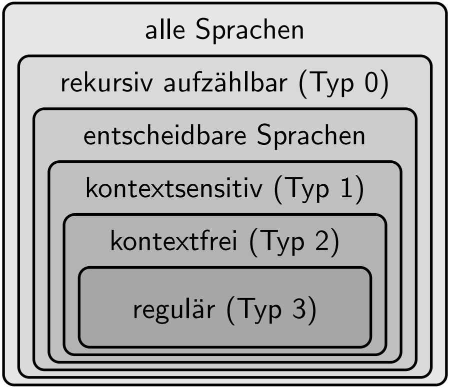
Entscheidbarkeit

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.

Eigenschaften der Typ i-Sprachen:

- Alle Typ 1, 2, 3-Sprachen sind **entscheidbar**.
- 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.**

Übersicht über die Sprachen



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

Typ i-Sprachen aus praktischer Sicht

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 Variable deklariert wurden usw.

Das Wortprobleme

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 $w \notin L(G)$.

Wortproblem für Typ 1-Sprachen

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.

Wortproblem für Typ 1-Sprachen

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.

Wesentliche Idee, warum das Wortproblem entscheidbar ist:

- Betrachte $S \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_m$ 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

Beweis: Entscheidbarkeit des Wortproblems für Typ 1-Sprachen

Sei $G = (V, \Sigma, P, S)$ eine Typ 1-Grammatik und $w \in \Sigma^*$.
Für $m, n \in \mathbb{N}$ 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 $n > 0$):

$$\begin{aligned} L_0^n &:= \{S\} \\ L_m^n &:= \mathit{next}(L_{m-1}^n, n) \text{ für } m > 0 \\ &\quad \text{wobei } \mathit{next}(L, n) := L \cup \{w' \mid w \in L, w \Rightarrow_G w', |w'| \leq n\} \end{aligned}$$

Die Berechnung terminiert, da die Mengen endlich sind: $|L_m^n| \leq |\Sigma \cup V|^n$
Wir können auch iterativ aus L_{i-1}^n die nachfolgende Menge L_i^n berechnen

Beweis: Entscheidbarkeit des Wortproblems für Typ 1-Sprachen (2)

Iterative Berechnung:

- Starte mit $L_0^n = \{S\}$
- Für $i = 1, 2, 3, \ldots$ berechne $L_i^n = next(L_{i-1}^n)$
- 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 $k \in \mathbb{N}$ 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 $i \in \mathbb{N}$ gelten: $L_i^n \supset L_{i-1}^n$
- Daher gilt für alle $i \in \mathbb{N}$: $|L_i^n| > |L_{i-1}^n|$.
- Widerspruch zu $|L_i^n| \leq |V \cup \Sigma|^n$ für alle $i \in \mathbb{N}$.

Algorithmus 2: Entscheiden des Wortproblems 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
    $L_{\text{old}} := L$;
    $L := next(L_{\text{old}}, n)$;
  bis *($w \in L$) oder ($L_{old} = L$)*;
  wenn $w \in L$ **dann**
    return *Ja*;
  sonst
    return *Nein*;
  Ende
Ende

Beispiel

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

Wir berechnen L_i^6 für alle i:

Beispiel

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

Wir berechnen L_i^6 für alle i:

$L_0^6 = \{S\}$

Beispiel

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

Wir berechnen L_i^6 für alle i:

$L_0^6 = \{S\}$

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

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

Beispiel

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

Wir berechnen L_i^6 für alle i:

$L_0^6 = \{S\}$

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

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

$$L_2^6 = next(L_1^6) = L_1^6 \cup \{w' \mid w \in L_1^6, w \Rightarrow w', |w'| \leq 6\}$$
$$= \{S, aSBc, abc, aabcBc\}$$

da $aSBc \Rightarrow aaSBcBc$ (zu lang) und $aSBc \Rightarrow aabcBc$

Beispiel

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

Wir berechnen L_i^6 für alle i:

$L_0^6 = \{S\}$

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

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

$$L_2^6 = \mathit{next}(L_1^6) = L_1^6 \cup \{w' \mid w \in L_1^6, w \Rightarrow w', |w'| \leq 6\}$$
$$= \{S, aSBc, abc, aabcBc\}$$

da $aSBc \Rightarrow aaSBcBc$ (zu lang) und $aSBc \Rightarrow aabcBc$

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

da $aabcBc \Rightarrow aabBcc$

Beispiel (2)

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

...

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

Beispiel (2)

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

...

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

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

da $aabBcc \Rightarrow aabbcc$

Beispiel (2)

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

...

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

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

da $aabBcc \Rightarrow aabbcc$

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

Beispiel (2)

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

...

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

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

da $aabBcc \Rightarrow aabbcc$

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

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

Wortproblem für Typ 2- und Typ 3-Sprachen

Korollar

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

Bemerkung:

- 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.

Weitere Entscheidungsprobleme

Leerheitsproblem

Das Leerheitsproblem für Sprachen vom Typ i ist die Frage, ob für eine Typ i-Grammatik G, die Gleichheit $L(G) = \emptyset$ gilt.

Endlichkeitsproblem

Das Endlichkeitsproblem für Sprachen vom Typ i ist die Frage, ob für eine Typ i-Grammatik G die Ungleichheit $|L| < \infty$ gilt.

Schnittproblem

Das Schnittproblem für Sprachen vom Typ i ist die Frage, ob für Typ i-Grammatiken G_1, G_2 gilt: $L(G_1) \cap L(G_2) = \emptyset$.

Äquivalenzproblem

Das Äquivalenzproblem für Sprachen vom Typ i ist, die Frage, ob Typ i-Grammatiken G_1, G_2 gilt: $L(G_1) = L(G_2)$.