

Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2018/19

Bemerkung

- Es ist klar, dass jede reguläre Grammatik auch kontextfrei ist
- Zudem ist die Sprache $L = \{a^n b^n \mid n \geq 0\}$ nicht regulär
- Es ist aber leicht, eine kontextfreie Grammatik für L anzugeben:

$$G = (\{S\}, \{a, b\}, P, S) \text{ mit } P = \{S \rightarrow aSb, \varepsilon\}$$

- Also gilt $\text{REG} \not\subseteq \text{CFL}$
- Allerdings sind nicht alle kontextfreien Grammatiken kontextsensitiv
- Z.B. ist obige Grammatik G nicht kontextsensitiv, da sie die Regel $S \rightarrow \varepsilon$ enthält und S auf der rechten Seite der Regel $S \rightarrow aSb$ vorkommt
- Wir können G jedoch wie folgt in eine Grammatik G' umwandeln:
 - ersetze die Regel $S \rightarrow \varepsilon$ durch die Regel $S \rightarrow ab$ und
 - füge ein neues Startsymbol S' sowie die Regeln $S' \rightarrow S, \varepsilon$ hinzu
- Tatsächlich lässt sich jede kontextfreie Grammatik G in eine äquivalente kontextfreie Grammatik G' umwandeln, die auch kontextsensitiv ist

Definition

Eine Grammatik $G = (V, \Sigma, P, S)$ ist in **Chomsky-Normalform (CNF)**, falls $P \subseteq V \times (V^2 \cup \Sigma)$ ist, d.h. alle Regeln haben die Form $A \rightarrow BC$ oder $A \rightarrow a$.

Satz

Zu jeder kontextfreien Grammatik G lässt sich eine CNF-Grammatik G' mit $L(G') = L(G) \setminus \{\varepsilon\}$ konstruieren.

Korollar

CFL \subseteq CSL.

Beweis

- Sei $L \in \text{CFL}$ und sei $G = (V, \Sigma, P, S)$ eine CNF-Grammatik mit $L(G) = L \setminus \{\varepsilon\}$
- Im Fall $\varepsilon \notin L$ folgt sofort $L = L(G) \in \text{CSL}$, da G kontextsensitiv ist
- Ist $\varepsilon \in L$, so erzeugt folgende kontextsensitive (und kontextfreie) Grammatik G' die Sprache $L = L(G) \cup \{\varepsilon\}$:

$$G' = (V \cup \{S_{neu}\}, \Sigma, P \cup \{S_{neu} \rightarrow S, \varepsilon\}, S_{neu})$$

□

- Der Beweis des Pumping-Lemmas für kontextfreie Sprachen basiert auf CNF-Grammatiken
- Zudem ermöglichen sie einen effizienten Algorithmus zur Lösung des Wortproblems für kontextfreie Sprachen

Das Pumping-Lemma für kontextfreie Sprachen

Zu jeder kontextfreien Sprache $L \in \text{CFL}$ gibt es eine Zahl l , so dass sich alle Wörter $z \in L$ mit $|z| \geq l$ in $z = uvwxy$ zerlegen lassen mit

- 1 $vx \neq \varepsilon$,
- 2 $|vwx| \leq l$ und
- 3 $uv^iwx^iy \in L$ für alle $i \geq 0$.

Das Wortproblem für kontextfreie Grammatiken

Gegeben: Eine kontextfreie Grammatik G und ein Wort x .

Gefragt: Ist $x \in L(G)$?

Beispiel

- Betrachte die Sprache $L = \{a^n b^n \mid n \geq 0\}$.
- Dann lässt sich jedes Wort $z = a^n b^n = a^{n-1} a b b^{n-1}$ in L mit $|z| \geq l = 2$ pumpen.
- Zerlegen wir nämlich z in

$$z = uvwxy \text{ mit } u = a^{n-1}, v = a, w = \varepsilon, x = b \text{ und } y = b^{n-1},$$

dann gilt

- ① $vx = ab \neq \varepsilon$
- ② $|vwx| = |ab| \leq 2$ und
- ③ $uv^i wx^i y = a^{n-1} a^i b^i b^{n-1} \in L$ für alle $i \geq 0$



Beispiel

- Die Sprache $L = \{a^n b^n c^n \mid n \geq 0\}$ ist nicht kontextfrei
- Für eine vorgegebene Zahl $l \geq 0$ hat nämlich das Wort $z = a^l b^l c^l \in L$ die Länge $|z| = 3l \geq l$
- Dieses Wort lässt sich aber nicht pumpen:

Für jede Zerlegung $z = uvwxy$ mit $vx \neq \varepsilon$ und $|vwx| \leq l$ gehört $z' = uv^0wx^0y$ nicht zu L :

- Wegen $vx \neq \varepsilon$ ist $|z'| < |z|$
- Wegen $|vwx| \leq l$ kommen in vx nicht alle drei Zeichen a, b, c vor
- Kommt aber in vx beispielsweise kein a vor, so ist $\#_a(z) = \#_a(z')$ und somit gilt

$$|z'| < |z| = 3 \#_a(z) = 3 \#_a(z')$$

- Also gehört z' nicht zu L



Abschlusseigenschaften von CFL

Satz

CFL ist abgeschlossen unter Vereinigung, Produkt und Sternhülle.

Beweis

- Seien $G_1 = (V_1, \Sigma, P_1, S_1)$ und $G_2 = (V_2, \Sigma, P_2, S_2)$ kontextfreie Grammatiken mit $V_1 \cap V_2 = \emptyset$ und sei S eine neue Variable
- Dann gilt
 - $L(G_1) \cup L(G_2) = L(G_3)$ für die kontextfreie Grammatik

$$G_3 = (V_1 \cup V_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \rightarrow S_1, S_2\}, S)$$
 - $L(G_1)L(G_2) = L(G_4)$ für die kontextfreie Grammatik

$$G_4 = (V_1 \cup V_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \rightarrow S_1S_2\}, S)$$
 und
 - $L(G_1)^* = L(G_5)$ für die kontextfreie Grammatik

$$G_5 = (V_1 \cup \{S\}, \Sigma, P_1 \cup \{S \rightarrow S_1S, \varepsilon\}, S)$$

Abschlusseigenschaften von CFL

Satz

CFL ist nicht abgeschlossen unter Schnitt und Komplement.

Beweis von $L_1, L_2 \in \text{CFL} \not\Rightarrow L_1 \cap L_2 \in \text{CFL}$

- Folgende Sprachen sind kontextfrei (siehe Übungen):

$$L_1 = \{a^n b^m c^m \mid n, m \geq 0\} \quad \text{und} \quad L_2 = \{a^n b^n c^m \mid n, m \geq 0\}$$

- Nicht jedoch ihr Schnitt $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ □

Beweis von $L \in \text{CFL} \not\Rightarrow \bar{L} \in \text{CFL}$

- Wäre CFL unter Komplement abgeschlossen, so wäre CFL wegen de Morgan auch unter Schnitt abgeschlossen
- Mit $A, B \in \text{CFL}$ wären dann nämlich auch $\bar{A}, \bar{B} \in \text{CFL}$, woraus wegen

$$\bar{A}, \bar{B} \in \text{CFL} \Rightarrow \overline{\bar{A} \cap \bar{B}} = \overline{\bar{A} \cap \bar{B}} \in \text{CFL}$$

wiederum $A \cap B \in \text{CFL}$ folgen würde □

Umwandlung in Chomsky-Normalform

Satz

Zu jeder kontextfreien Grammatik G lässt sich eine CNF-Grammatik G' mit $L(G') = L(G) \setminus \{\varepsilon\}$ konstruieren.

Beweis

Wir wandeln $G = (V, \Sigma, P, S)$ wie folgt in eine CNF-Grammatik G' um:

- Wir beseitigen zunächst alle Regeln der Form $A \rightarrow \varepsilon$ und danach alle Regeln der Form $A \rightarrow B$ (siehe folgende Folien)
- Dann fügen wir für jedes Terminal $a \in \Sigma$ eine neue Variable X_a und eine neue Regel $X_a \rightarrow a$ hinzu und ersetzen jedes Vorkommen von a , bei dem a nicht alleine auf der rechten Seite einer Regel steht, durch X_a
- Anschließend führen wir für jede Regel $A \rightarrow B_1 \dots B_k$, $k \geq 3$, neue Variablen A_1, \dots, A_{k-2} ein und ersetzen sie durch die $k - 1$ Regeln

$$A \rightarrow B_1 A_1, A_1 \rightarrow B_2 A_2, \dots, A_{k-3} \rightarrow B_{k-2} A_{k-2}, A_{k-2} \rightarrow B_{k-1} B_k \quad \square$$

Falls G Regeln mit vielen Variablen auf der rechten Seite hat, empfiehlt es sich, Regeln der Form $A \rightarrow \varepsilon$ und $A \rightarrow B$ zuletzt zu beseitigen (s. Übungen)

Beseitigung von ε -Regeln

Satz

Zu jeder kontextfreien Grammatik $G = (V, \Sigma, P, S)$ gibt es eine kontextfreie Grammatik $G' = (V, \Sigma, P', S)$ ohne ε -Regeln mit $L(G') = L(G) \setminus \{\varepsilon\}$.

Beweis

- Zuerst berechnen wir die Menge $E = \{A \in V \mid A \Rightarrow^* \varepsilon\}$ aller Variablen, die nach ε ableitbar sind:

```

1    $E' := \{A \in V \mid A \rightarrow \varepsilon\}$ 
2   repeat
3      $E := E'$ 
4      $E' := E \cup \{A \in V \mid \exists B_1, \dots, B_k \in E : A \rightarrow B_1 \dots B_k\}$ 
5   until  $E = E'$ 
  
```

- Nun bilden wir P' wie folgt:

$$\left\{ A \rightarrow v' \mid \begin{array}{l} \text{es ex. eine Regel } A \rightarrow_G v, \text{ so dass } v' \neq \varepsilon \text{ aus } v \text{ durch} \\ \text{Entfernen von beliebig vielen Variablen } A \in E \text{ entsteht} \end{array} \right\} \quad \square$$

Beseitigung von ε -Regeln

Beispiel

Betrachte die Grammatik $G = (\{S, T, U, X, Y, Z\}, \{a, b, c\}, P, S)$ mit

$$P: \begin{array}{lll} S \rightarrow aY, bX, Z & Y \rightarrow bS, aYY & T \rightarrow U \\ X \rightarrow aS, bXX & Z \rightarrow \varepsilon, S, T, cZ & U \rightarrow abc \end{array}$$

- Berechnung von E :

E'	$\{Z\}$	$\{Z, S\}$
E	$\{Z, S\}$	$\{Z, S\}$

- Entferne $Z \rightarrow \varepsilon$ und füge die Regeln $Y \rightarrow b$ (wegen $Y \rightarrow bS$), $X \rightarrow a$ (wegen $X \rightarrow aS$) und $Z \rightarrow c$ (wegen $Z \rightarrow cZ$) hinzu:

$$P': \begin{array}{lll} S \rightarrow aY, bX, Z & Y \rightarrow b, bS, aYY & T \rightarrow U \\ X \rightarrow a, aS, bXX & Z \rightarrow c, S, T, cZ & U \rightarrow abc \end{array}$$

Beseitigung von Variablenumbenennungen

Satz

Zu jeder kontextfreien Grammatik $G = (V, \Sigma, P, S)$ gibt es eine kontextfreie Grammatik $G' = (V, \Sigma, P', S)$ ohne Regeln der Form $A \rightarrow B$ mit $L(G') = L(G)$.

Beweis

- Zuerst entfernen wir sukzessive alle Zyklen $A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_k \rightarrow A_1$
- Hierzu entfernen wir diese Regeln aus P und ersetzen alle Vorkommen der Variablen A_2, \dots, A_k in den übrigen Regeln durch A_1
- Befindet sich die Startvariable unter A_1, \dots, A_k , so sei dies o.B.d.A. A_1
- Nun eliminieren wir sukzessive die restlichen Variablenumbenennungen, indem wir
 - eine Regel $A \rightarrow B$ wählen, so dass in P keine Variablenumbenennung $B \rightarrow C$ mit B auf der linken Seite existiert,
 - diese Regel $A \rightarrow B$ aus P entfernen und
 - für jede Regel $B \rightarrow v$ in P die Regel $A \rightarrow v$ zu P hinzunehmen □

Beseitigung von Variablenumbenennungen

Beispiel (Fortsetzung)

$$P: \quad S \rightarrow aY, bX, Z \quad Y \rightarrow b, bS, aYY \quad T \rightarrow U \\ X \rightarrow a, aS, bXX \quad Z \rightarrow c, S, T, cZ \quad U \rightarrow abc$$

- Entferne den Zyklus $S \rightarrow Z \rightarrow S$ und ersetze Z durch S :

$$S \rightarrow aY, bX, c, T, cS \quad Y \rightarrow b, bS, aYY \quad T \rightarrow U \\ X \rightarrow a, aS, bXX \quad U \rightarrow abc$$

- Ersetze die Regel $T \rightarrow U$ durch $T \rightarrow abc$ (wegen $U \rightarrow abc$):

$$S \rightarrow aY, bX, c, T, cS \quad Y \rightarrow b, bS, aYY \quad T \rightarrow abc \\ X \rightarrow a, aS, bXX \quad U \rightarrow abc$$

- Ersetze dann auch die Regel $S \rightarrow T$ durch $S \rightarrow abc$ (wegen $T \rightarrow abc$):

$$S \rightarrow abc, aY, bX, c, cS \quad Y \rightarrow b, bS, aYY \quad T \rightarrow abc \\ X \rightarrow a, aS, bXX \quad U \rightarrow abc$$

- Da T und U nirgends mehr auf der rechten Seite vorkommen, können wir die Regeln $T \rightarrow abc$ und $U \rightarrow abc$ weglassen:

$$S \rightarrow abc, aY, bX, c, cS \quad Y \rightarrow b, bS, aYY \quad X \rightarrow a, aS, bXX$$

Beispiel (Schluss)

Betrachte die Grammatik $G = (\{S, X, Y, Z\}, \{a, b, c\}, P, S)$ mit

$$P: S \rightarrow abc, aY, bX, c, cS \quad Y \rightarrow b, bS, aYY \quad X \rightarrow a, aS, bXX$$

- Ersetze a , b und c durch A , B und C (außer wenn sie alleine auf der rechten Seite einer Regel stehen) und füge die Regeln $A \rightarrow a$, $B \rightarrow b$, $C \rightarrow c$ hinzu:

$$S \rightarrow ABC, AY, BX, c, CS \quad Y \rightarrow b, BS, AYY \quad X \rightarrow a, AS, BXX$$

$$A \rightarrow a \quad B \rightarrow b \quad C \rightarrow c$$

- Ersetze die Regeln $S \rightarrow ABC$, $Y \rightarrow AYY$ und $X \rightarrow BXX$ durch die Regeln $S \rightarrow AS'$, $S' \rightarrow BC$, $Y \rightarrow AY'$, $Y' \rightarrow YY$ und $X \rightarrow BX'$, $X' \rightarrow XX$:

$$S \rightarrow AS', AY, BX, c, CS \quad S' \rightarrow BC \quad Y \rightarrow b, BS, AY' \quad Y' \rightarrow YY$$

$$X \rightarrow a, AS, BX' \quad X' \rightarrow XX \quad A \rightarrow a \quad B \rightarrow b \quad C \rightarrow c$$



Links- und Rechtsableitungen

Definition

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

- Eine Ableitung

$$\underline{S} \Rightarrow l_1 \underline{A_1} r_1 \Rightarrow \dots \Rightarrow l_{m-1} \underline{A_{m-1}} r_{m-1} \Rightarrow \alpha_m$$

heißt **Linksableitung** von α_m (kurz $S \Rightarrow_L^* \alpha_m$), falls in jedem Ableitungsschritt die am weitesten links stehende Variable ersetzt wird, d.h. es gilt $l_i \in \Sigma^*$ für $i = 1, \dots, m-1$

- **Rechtsableitungen** $S_0 \Rightarrow_R^* \alpha_m$ sind analog definiert
- G heißt **mehrdeutig**, wenn es ein Wort $x \in L(G)$ gibt, das zwei verschiedene Linksableitungen hat
- Andernfalls heißt G **eindeutig**

Für alle $x \in \Sigma^*$ gilt: $x \in L(G) \Leftrightarrow S \Rightarrow^* x \Leftrightarrow S \Rightarrow_L^* x \Leftrightarrow S \Rightarrow_R^* x$

Ein- und mehrdeutige Grammatiken

Beispiel

- In $G = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, \varepsilon\}, S)$ gibt es **8** Ableitungen für $aabb$:

$$\underline{S} \Rightarrow_L a\underline{S}bS \Rightarrow_L aa\underline{S}bSbS \Rightarrow_L aab\underline{S}bS \Rightarrow_L aabb\underline{S} \Rightarrow_L aabb$$

$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aa\underline{S}bSbS \Rightarrow aabSb\underline{S} \Rightarrow aab\underline{S}b \Rightarrow aabb$$

$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aa\underline{S}bSbS \Rightarrow aa\underline{S}bbS \Rightarrow aabb\underline{S} \Rightarrow aabb$$

$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aa\underline{S}bSbS \Rightarrow aaSbb\underline{S} \Rightarrow aa\underline{S}bb \Rightarrow aabb$$

$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aa\underline{S}bSb\underline{S} \Rightarrow aa\underline{S}bSb \Rightarrow aab\underline{S}b \Rightarrow aabb$$

$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aa\underline{S}bSb\underline{S} \Rightarrow aaSb\underline{S}b \Rightarrow aa\underline{S}bb \Rightarrow aabb$$

$$\underline{S} \Rightarrow aSb\underline{S} \Rightarrow a\underline{S}b \Rightarrow aa\underline{S}bSb \Rightarrow aab\underline{S}b \Rightarrow aabb$$

$$\underline{S} \Rightarrow_R aSb\underline{S} \Rightarrow_R a\underline{S}b \Rightarrow_R aaSb\underline{S}b \Rightarrow_R aa\underline{S}bb \Rightarrow_R aabb$$

- Darunter sind genau eine **Links-** und genau eine **Rechtsableitung**
- In $G' = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, ab, \varepsilon\}, S)$ gibt es **3** Ableitungen für ab :

$$\underline{S} \Rightarrow ab$$

$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow ab\underline{S} \Rightarrow ab$$

$$\underline{S} \Rightarrow aSb\underline{S} \Rightarrow a\underline{S}b \Rightarrow ab$$

- Darunter sind **zwei Links-** und **zwei Rechtsableitungen**

Beispiel

- Die Grammatik $G = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, \varepsilon\}, S)$ ist eindeutig
- Dies liegt daran, dass keine Satzform von G das Teilwort Sa enthält
- Daher muss auf die aktuelle Satzform $y\underline{S}\beta$ einer Linksableitung

$$S \Rightarrow_L^* y\underline{S}\beta \Rightarrow_L^* yz = x$$

genau dann die Regel $S \rightarrow aSbS$ angewandt werden, wenn in x auf das Präfix y ein a folgt

- Dagegen ist die Grammatik $G' = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, ab, \varepsilon\}, S)$ mehrdeutig, da das Wort $x = ab$ zwei Linksableitungen hat:

$$\underline{S} \Rightarrow ab \quad \text{und} \quad \underline{S} \Rightarrow a\underline{S}bS \Rightarrow ab\underline{S} \Rightarrow ab$$



Gerichtete Bäume und Wälder

Sei $G = (V, E)$ ein Digraph.

- Ein (**gerichteter**) v_0 - v_k -**Weg** in G ist eine Folge von Knoten v_0, \dots, v_k mit $(v_i, v_{i+1}) \in E$ für $i = 0, \dots, k-1$. Seine **Länge** ist k .
- Ein Weg heißt **Pfad**, falls alle Knoten paarweise verschieden sind.
- Ein u - v -Weg der Länge ≥ 1 mit $u = v$ heißt **Zyklus**.
- G heißt **azyklisch**, wenn es in G keinen Zyklus gibt.
- G heißt **gerichteter Wald**, wenn G azyklisch ist und jeder Knoten $v \in V$ Eingangsgrad $\deg^-(v) \leq 1$ hat.
- Ein Knoten $u \in V$ vom Ausgangsgrad $\deg^+(u) = 0$ heißt **Blatt**.
- Ein Knoten $w \in V$ heißt **Wurzel** von G , falls alle Knoten $v \in V$ von w aus erreichbar sind (d.h. es gibt einen w - v -Weg in G).
- Ein **gerichteter Wald**, der eine Wurzel hat, heißt **gerichteter Baum**.
- Da die Kantenrichtungen durch die Wahl der Wurzel eindeutig bestimmt sind, kann auf ihre Angabe verzichtet werden. Man spricht dann auch von einem **Wurzelbaum**.

Wir ordnen einer Ableitung

$$\underline{A_0} \Rightarrow l_1 \underline{A_1} r_1 \Rightarrow \dots \Rightarrow l_{m-1} \underline{A_{m-1}} r_{m-1} \Rightarrow \alpha_m$$

den **Syntaxbaum** (oder **Ableitungsbaum**, engl. *parse tree*) T_m zu, wobei die Bäume T_0, \dots, T_m induktiv wie folgt definiert sind:

- T_0 besteht aus einem einzigen Knoten, der mit A_0 markiert ist.
- Wird im $(i+1)$ -ten Ableitungsschritt die Regel $A_i \rightarrow v_1 \dots v_k$ mit $v_1, \dots, v_k \in \Sigma \cup V$ angewandt, so entsteht T_{i+1} aus T_i , indem wir das Blatt A_i durch folgenden Unterbaum ersetzen:



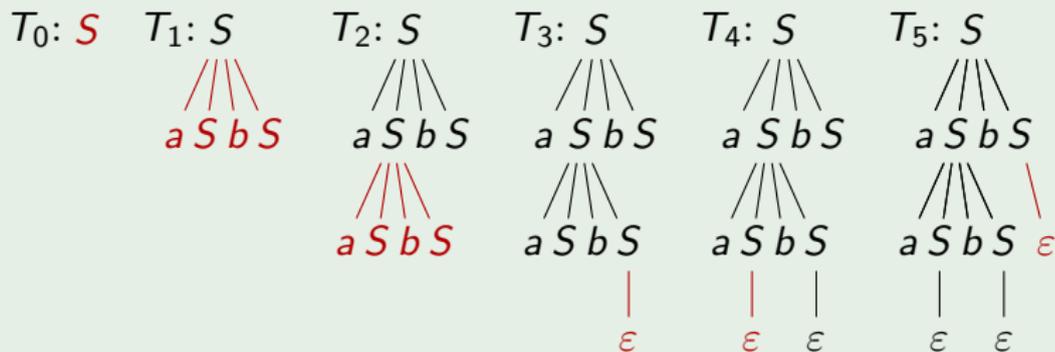
- Hierbei stellen wir uns die Kanten von oben nach unten gerichtet und die Kinder $v_1 \dots v_k$ von links nach rechts geordnet vor.
- Syntaxbäume sind also **geordnete** Wurzelbäume.

Beispiel

- Betrachte die Grammatik $G = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, \epsilon\}, S)$ und die Ableitung

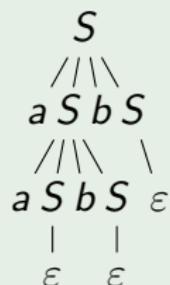
$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aa\underline{S}b\underline{S}bS \Rightarrow aa\underline{S}bbS \Rightarrow aabb\underline{S} \Rightarrow aabb$$

- Die zugehörigen Syntaxbäume sind dann

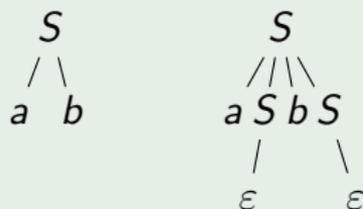


Beispiel

- In $G = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, \varepsilon\}, S)$ führen alle acht Ableitungen des Wortes $aabb$ auf denselben Syntaxbaum:



- Dagegen führen in $G' = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, ab, \varepsilon\}, S)$ die drei Ableitungen des Wortes ab auf zwei unterschiedliche Syntaxbäume:



Syntaxbäume und Linksableitungen

- Seien T_0, \dots, T_m die zu einer Ableitung $S = \alpha_0 \Rightarrow \dots \Rightarrow \alpha_m$ gehörigen Syntaxbäume.
- Dann haben alle Syntaxbäume T_0, \dots, T_m die Wurzel S .
- Die Satzform α_i ergibt sich aus T_i , indem wir die Blätter von T_i von links nach rechts zu einem Wort zusammensetzen.
- Auf den Syntaxbaum T_m führen neben $\alpha_0 \Rightarrow \dots \Rightarrow \alpha_m$ alle Ableitungen, die sich von dieser nur in der Reihenfolge der Regelanwendungen unterscheiden.
- Dazu gehört genau eine Linksableitung.
- Linksableitungen und Syntaxbäume entsprechen sich also eineindeutig.
- Dasselbe gilt für Rechtsableitungen.
- Ist T Syntaxbaum einer CNF-Grammatik, so hat jeder Knoten in T höchstens zwei Kinder (d.h. T ist ein **Binärbaum**).

Definition

Die **Tiefe** eines Baumes mit Wurzel w ist die maximale Länge eines Weges von w zu einem Blatt.

Lemma

Ein Binärbaum B der Tiefe $\leq k$ hat $\leq 2^k$ Blätter.

Beweis durch Induktion über k :

$k = 0$: Ein Baum der Tiefe 0 kann nur einen Knoten haben.

$k \rightsquigarrow k + 1$: Sei B ein Binärbaum der Tiefe $\leq k + 1$.

Dann hängen an B 's Wurzel maximal zwei Unterbäume.

Da deren Tiefe $\leq k$ ist, haben sie nach IV $\leq 2^k$ Blätter.

Also hat $B \leq 2^{k+1}$ Blätter. □

Lemma

Ein Binärbaum B der Tiefe $\leq k$ hat $\leq 2^k$ Blätter.

Korollar

Ein Binärbaum B mit $> 2^{k-1}$ Blättern hat eine Tiefe $\geq k$.

Beweis

Wäre die Tiefe von B kleiner als k (also $\leq k - 1$), so hätte B nach obigem Lemma $\leq 2^{k-1}$ Blätter (Widerspruch). □