

Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2016/17

Bemerkung

- Wie wir gesehen haben, ist folgende Sprache nicht regulär:

$$L = \{a^n b^n \mid n \geq 0\}.$$

- Es ist aber leicht, eine kontextfreie Grammatik für L zu finden:

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

- Damit ist klar, dass die Klasse der regulären Sprachen echt in der Klasse der kontextfreien Sprachen enthalten ist:

$$\text{REG} \subsetneq \text{CFL}.$$

- Als nächstes wollen wir zeigen, dass die Klasse der kontextfreien Sprachen wiederum echt in der Klasse der kontextsensitiven Sprachen enthalten ist:

$$\text{CFL} \subsetneq \text{CSL}.$$

- Kontextfreie Grammatiken sind dadurch charakterisiert, dass sie nur Regeln der Form $A \rightarrow \alpha$ haben.
- Dies lässt die Verwendung von beliebigen ε -Regeln der Form $A \rightarrow \varepsilon$ zu.
- Eine kontextsensitive Grammatik darf dagegen höchstens die ε -Regel $S \rightarrow \varepsilon$ haben.
- Voraussetzung hierfür ist, dass S das Startsymbol ist und dieses nicht auf der rechten Seite einer Regel vorkommt.
- Daher sind nicht alle kontextfreien Grammatiken kontextsensitiv.
- Beispielsweise ist die Grammatik $G = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$ nicht kontextsensitiv, da sie die Regel $S \rightarrow \varepsilon$ enthält, obwohl S auf der rechten Seite der Regel $S \rightarrow aSb$ vorkommt.
- Wir werden jedoch sehen, dass sich zu jeder kontextfreien Grammatik eine äquivalente kontextsensitive Grammatik konstruieren lässt.

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 *ε -ableitbaren* Variablen:

```

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:

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

Beispiel

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

$$P: \quad S \rightarrow aY, bX, Z; \quad Y \rightarrow bS, aYY; \quad T \rightarrow U; \\ X \rightarrow aS, bXX; \quad Z \rightarrow \varepsilon, S, T, cZ; \quad U \rightarrow abc.$$

- Berechnung von E :

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

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

$$P': \quad S \rightarrow aY, bX, Z; \quad Y \rightarrow b, bS, aYY; \quad T \rightarrow U; \\ X \rightarrow a, aS, bXX; \quad Z \rightarrow c, S, T, cZ; \quad U \rightarrow abc.$$

Satz

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

Korollar

$\text{REG} \not\subseteq \text{CFL} \subseteq \text{CSL} \subseteq \text{RE}$.

Beweis

- Es ist nur noch die Inklusion $\text{CFL} \subseteq \text{CSL}$ zu zeigen.
- Nach obigem Satz ex. zu $L \in \text{CFL}$ eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ohne ε -Regeln mit $L(G) = L \setminus \{\varepsilon\}$.
- Da G dann auch kontextsensitiv ist, folgt hieraus im Fall $\varepsilon \notin L$ unmittelbar $L(G) = L \in \text{CSL}$.
- Im Fall $\varepsilon \in L$ erzeugt die kontextsensitive Grammatik

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

die Sprache $L(G') = L$, d.h. $L \in \text{CSL}$.

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 erzeugen die kontextfreien Grammatiken

$$G_3 = (V_1 \cup V_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \rightarrow S_1, S_2\}, S)$$

die Vereinigung $L(G_3) = L(G_1) \cup L(G_2)$,

$$G_4 = (V_1 \cup V_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}, S)$$

das Produkt $L(G_4) = L(G_1)L(G_2)$ und

$$G_5 = (V_1 \cup \{S\}, \Sigma, P_1 \cup \{S \rightarrow S_1 S, \epsilon\}, S)$$

die Sternhülle $L(G_1)^*$. □

Satz

CFL ist abgeschlossen unter Vereinigung, Produkt und Sternhülle.

Frage

Ist die Klasse CFL auch abgeschlossen unter

- Schnitt und
- Komplement?

Antwort

Nein.

Hierzu müssen wir für bestimmte Sprachen nachweisen, dass sie nicht kontextfrei sind. Dies gelingt mit einem Pumping-Lemma für kontextfreie Sprachen, für dessen Beweis wir Grammatiken in Chomsky-Normalform benötigen.

Definition

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

Satz

Zu jeder kontextfreien Sprache $L \in \text{CFL}$ gibt es eine CNF-Grammatik G' mit $L(G') = L \setminus \{\varepsilon\}$.

Anwendungen der Chomsky-Normalform

- CNF-Grammatiken ermöglichen den Beweis des Pumping-Lemmas für kontextfreie Sprachen.
- Zudem bilden sie die Basis für eine effiziente Lösung des Wortproblems für kontextfreie Sprachen.

Satz (Pumping-Lemma für kontextfreie Sprachen)

Zu jeder kontextfreien Sprache L 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 $x \in L(G)$?

Satz

Das Wortproblem für kontextfreie Grammatiken ist effizient entscheidbar.

Um eine kontextfreie Grammatik in Chomsky-Normalform zu bringen, müssen wir neben den ε -Regeln $A \rightarrow \varepsilon$ auch sämtliche Variablenumbenennungen $A \rightarrow B$ loswerden.

Definition

Regeln der Form $A \rightarrow B$ heißen **Variablenumbenennungen**.

Satz

Zu jeder kontextfreien Grammatik G ex. eine kontextfreie Grammatik G' ohne Variablenumbenennungen 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 .

(Sollte sich unter den entfernten Variablen A_2, \dots, A_k die Startvariable S befinden, so sei A_1 die neue Startvariable.)

Beispiel (Fortsetzung)

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

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

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

Entfernen von Variablenumbenennungen

Satz

Zu jeder kontextfreien Grammatik G ex. eine kontextfreie Grammatik G' ohne Variablenumbenennungen mit $L(G') = L(G)$.

Beweis

- Zuerst entfernen wir alle Zyklen

$$A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_k \rightarrow A_1.$$

- Nun werden wir sukzessive die restlichen Variablenumbenennungen los, 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 $B \rightarrow \alpha$ in P die Regel $A \rightarrow \alpha$ zu P hinzunehmen. \square

Beispiel (Fortsetzung)

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

- Entferne die Regel $T \rightarrow U$ und füge die Regel $T \rightarrow abc$ hinzu (wegen $U \rightarrow abc$):

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

- Entferne dann auch die Regel $S \rightarrow T$ und füge die Regel $S \rightarrow abc$ (wegen $T \rightarrow abc$) hinzu:

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

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

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



Bereits gezeigt:

Korollar

Zu jeder kontextfreien Grammatik G ex. eine kontextfreie Grammatik G' ohne ε -Regeln und ohne Variablenumbenennungen mit $L(G') = L(G) \setminus \{\varepsilon\}$.

Noch zu zeigen:

Satz

Zu jeder kontextfreien Sprache $L \in \text{CFL}$ gibt es eine CNF-Grammatik G' mit $L(G') = L \setminus \{\varepsilon\}$.

Umwandlung in Chomsky-Normalform

Satz

Zu jeder kontextfreien Sprache $L \in \text{CFL}$ gibt es eine CNF-Grammatik G' mit $L(G') = L \setminus \{\varepsilon\}$.

Beweis

- Sei $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik ohne ε -Regeln und ohne Variablenumbenennungen für $L \setminus \{\varepsilon\}$.
- Wir transformieren G wie folgt in eine CNF-Grammatik.
- Füge für jedes Terminalsymbol $a \in \Sigma$ eine neue Variable X_a zu V und eine neue Regel $X_a \rightarrow a$ zu P hinzu.
- Ersetze alle Vorkommen von a durch X_a , außer wenn a alleine auf der rechten Seite einer Regel steht.
- Ersetze jede Regel $A \rightarrow B_1 \dots B_k$, $k \geq 3$, durch die $k - 1$ Regeln

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

wobei A_1, \dots, A_{k-2} neue Variablen sind. □

Beispiel (Fortsetzung)

- Betrachte die Regeln

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

- Ersetze a , b und c durch A , B und C (außer wenn sie alleine rechts vorkommen) und füge die Regeln $A \rightarrow a$, $B \rightarrow b$, $C \rightarrow c$ hinzu:

$$S \rightarrow ABC, AY, BX, CS, c; \quad X \rightarrow AS, BXX, a;$$

$$Y \rightarrow BS, AYY, b; \quad A \rightarrow a; \quad B \rightarrow b; \quad C \rightarrow c.$$

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

$$S \rightarrow AS', AY, BX, CS, c; \quad S' \rightarrow BC; \quad X \rightarrow AS, BX', a; \quad X' \rightarrow XX;$$

$$Y \rightarrow BS, AY', b; \quad Y' \rightarrow YY; \quad A \rightarrow a; \quad B \rightarrow b; \quad C \rightarrow c.$$


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**.

Leicht zu sehen:

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 das Wort $aabb$:

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

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

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

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

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

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

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

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


Beispiel

- Die Grammatik $G = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, \varepsilon\}, S)$ ist eindeutig.
- Dies liegt daran, dass in jeder Satzform $\alpha S \beta$ von G das Suffix β entweder leer ist oder mit einem b beginnt.
- Daher muss jede Linksableitung eines Wortes $x \in L(G)$ die am weitesten links stehende Variable der aktuellen Satzform $\alpha S \beta$ genau dann nach $aSbS$ expandieren, wenn in x auf das Präfix α 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.

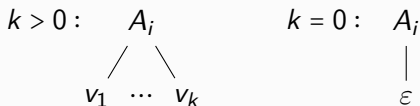
- Ein (**gerichteter**) v_0 - v_k -Weg in G ist eine Folge von Knoten v_0, \dots, v_k mit $(v_i, v_{i+1}) \in E$ für $i = 0, \dots, k-1$. Seine **Länge** ist k .
- Ein Weg heißt **Pfad**, falls alle Knoten paarweise verschieden sind.
- Ein u - v -Weg der Länge ≥ 1 mit $u = v$ heißt **Zyklus**.
- G heißt **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:



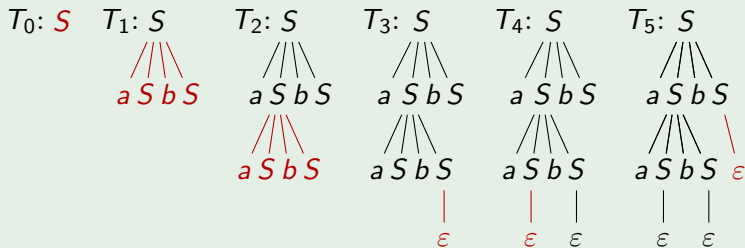
- Hierbei stellen wir uns die Kanten von oben nach unten gerichtet und die Kinder $v_1 \dots v_k$ von links nach rechts geordnet vor.
- Syntaxbäume sind also **geordnete** Wurzelbäume.

Beispiel

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

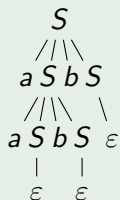
$$\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aa\underline{S}b\underline{S}bS \Rightarrow aa\underline{S}bbS \Rightarrow aabb\underline{S} \Rightarrow aabb$$

- Die zugehörigen Syntaxbäume sind dann



Beispiel

- In $G = (\{S\}, \{a, b\}, \{S \rightarrow aSbS, \epsilon\}, S)$ führen alle 8 Ableitungen des Wortes $aabb$ auf denselben Syntaxbaum:



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). □

Satz (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 $i \geq 0$.

Beispiel

- Betrachte die Sprache $L = \{a^n b^n | n \geq 0\}$.
- Dann lässt sich jedes Wort $z = a^n b^n = a^{n-1} a b b^{n-1}$ in L mit $|z| \geq 2$ pumpen:
 - Zerlege z in $z = uvwxy$ mit $u = a^{n-1}$, $v = a$, $w = \epsilon$, $x = b$, $y = b^{n-1}$.
 - Dann ist für alle $i \geq 0$ das Wort $uv^i wx^i y = a^{n-1} a^i b^i b^{n-1} \in L$.



Beispiel

- Die Sprache $\{a^n b^n c^n \mid n \geq 0\}$ ist nicht kontextfrei.
- Für eine vorgegebene Zahl $l \geq 0$ hat nämlich $z = a^l b^l c^l$ 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 $z' = uv^0wx^0y = uwy$ nicht zu L :

- Wegen $vx \neq \varepsilon$ ist $|z'| < |z|$.
- Wegen $|vwx| \leq l$ kann in vx nicht jedes der drei Zeichen a, b, c vorkommen.
- 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 .

Beweis des Pumping-Lemmas

Satz (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 $i \geq 0$.

Beweis

- Sei $G = (V, \Sigma, P, S)$ eine CNF-Grammatik für $L \setminus \{\varepsilon\}$.
- Ist nun $z = z_1 \dots z_n \in L$ mit $n \geq 1$, so ex. in G eine Ableitung

$$S = \alpha_0 \Rightarrow \alpha_1 \dots \Rightarrow \alpha_m = z.$$

- Da G in CNF ist, werden hierbei genau $n-1$ Regeln der Form $A \rightarrow BC$ und genau n Regeln der Form $A \rightarrow a$ angewandt.

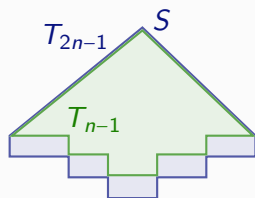
Beweis des Pumping-Lemmas

Beweis

- Sei $G = (V, \Sigma, P, S)$ eine CNF-Grammatik für $L \setminus \{\varepsilon\}$.
- Ist nun $z = z_1 \dots z_n \in L$ mit $n \geq 1$, so ex. in G eine Ableitung

$$S = \alpha_0 \Rightarrow \alpha_1 \dots \Rightarrow \alpha_m = z.$$

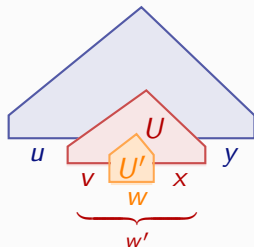
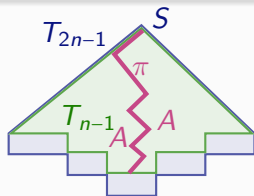
- Da G in CNF ist, werden hierbei genau $n-1$ Regeln der Form $A \rightarrow BC$ und genau n Regeln der Form $A \rightarrow a$ angewandt.
- Folglich ist $m = 2n - 1$ und z hat den Syntaxbaum T_{2n-1} .
- Wir können annehmen, dass die $n-1$ Regeln der Form $A \rightarrow BC$ vor den n Regeln der Form $A \rightarrow a$ zur Anwendung kommen.
- Dann besteht α_{n-1} aus n Variablen und T_{n-1} hat wie T_{2n-1} genau n Blätter.
- Setzen wir $l = 2^k$, wobei $k = \|V\|$ ist, so hat T_{n-1} im Fall $n \geq l$ mindestens die Tiefe k , da T_{n-1} mindestens $l = 2^k > 2^{k-1}$ Blätter hat.



Beweis des Pumping-Lemmas

Beweis (Fortsetzung)

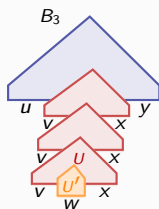
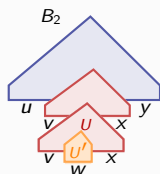
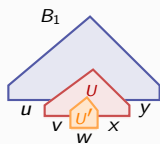
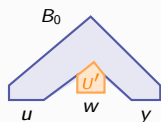
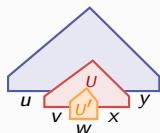
- Setzen wir $l = 2^k$, wobei $k = \|V\|$ ist, so hat T_{n-1} im Fall $n \geq l$ mindestens die Tiefe k , da T_{n-1} mindestens $l = 2^k > 2^{k-1}$ Blätter hat.
- Sei π ein von der Wurzel ausgehender Pfad maximaler Länge in T_{n-1} .
- Dann hat π mindestens die Länge k und unter den letzten $k + 1$ Knoten von π müssen zwei mit derselben Variablen A markiert sein.
- Seien U und U' die von diesen Knoten ausgehenden Unterbäume des vollständigen Syntaxbaums T_{2n-1} .
- Nun zerlegen wir z wie folgt:
 - w' ist das Teilwort von $z = uw'y$, das von U erzeugt wird und
 - w ist das Teilwort von $w' = vwx$, das von U' erzeugt wird.



Beweis des Pumping-Lemmas

Beweis (Schluss)

- Dann ist $vx \neq \varepsilon$ (Bed. 1), da U mehr Blätter hat als U' .
- Zudem hat U höchstens $2^k = l$ Blätter, da der Baum $U^* = U \cap T_{n-1}$ höchstens die Tiefe k hat (andernfalls wäre π nicht maximal).
Folglich ist $|vwx| \leq l$ (Bed. 2).
- Schließlich lassen sich Syntaxbäume B_i für die Wörter uv^iwx^iy , $i \geq 0$, wie folgt konstruieren (Bed. 3):
 - B_0 entsteht aus $B_1 = T_{2n-1}$, indem wir U durch U' ersetzen.
 - B_{i+1} entsteht aus B_i , indem wir U' durch U ersetzen:



Wie wir gesehen haben, ist die Klasse CFL abgeschlossen unter

- Vereinigung,
- Produkt und
- Sternhülle.

Satz

CFL ist nicht abgeschlossen unter

- Schnitt und
- Komplement.

Abschlusseigenschaften von CFL

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

- Die beiden Sprachen

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

sind kontextfrei (siehe Übungen).

- Nicht jedoch ihr Schnitt $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$.
- Also ist CFL nicht unter Schnitt abgeschlossen.

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:

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

Das Wortproblem für kontextfreie Grammatiken

Gegeben: Eine kontextfreie Grammatik G und ein Wort x .

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

Frage

Wie lässt sich das Wortproblem für kontextfreie Grammatiken entscheiden?

- Sei eine Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $x = x_1 \dots x_n$ gegeben.
- Falls $x = \varepsilon$ ist, können wir effizient prüfen, ob $S \Rightarrow^* \varepsilon$ gilt.
- Hierzu genügt es, die Menge $E = \{A \in V \mid A \Rightarrow^* \varepsilon\}$ aller ε -ableitbaren Variablen zu berechnen und zu prüfen, ob $S \in E$ ist.
- Andernfalls bringen wir G in CNF und starten den nach seinen Autoren **Cocke**, **Younger** und **Kasami** benannten **CYK-Algorithmus**.
- Dieser bestimmt mittels **dynamischer Programmierung** für $l = 1, \dots, n$ und $k = 1, \dots, n - l + 1$ die Menge $V_{l,k}$ aller Variablen, aus denen das Teilwort $x_k \dots x_{k+l-1}$ ableitbar ist.
- Dann gilt $x \in L(G) \Leftrightarrow S \in V_{n,1}$.