Anwendung des Pumping-Lemmas

Mit Hilfe des Pumping-Lemmas kann gezeigt werden, dass bestimmte Sprachen nicht kontextfrei sind.

Behauptung

Die Sprache $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ ist nicht kontextfrei.

Beweis (Skizze)

Annahme: L ist kontextfrei.

Sei p die Konstante aus dem Pumping-Lemma.

Sei $z = a^p b^p c^p$ (d.h., $z \in L$ und $length(z) \geq p$), und sei $z = uvwxy$ mit $length(vwx) \leq p$ und $vx \neq \lambda$.

1.Fall: $vx = a^m b^n$ mit $1 \leq m + n \leq p$. Dann gilt $uv^0wx^0y = a^{p-m} b^{p-n} c^p \notin L$. (Widerspruch)

2.Fall: $vx = b^m c^n$ mit $1 \leq m + n \leq p$. Analog.

Korollar. Die Klasse $\mathcal{L}_{KFS}$ der kontextfreien Sprachen ist unter Schnitt nicht abgeschlossen.

Beweisidee: Die Sprachen $L_1 = \{a^n b^n c^m \mid m, n \in \mathbb{N}\}$ und $L_2 = \{a^m b^n c^n \mid m, n \in \mathbb{N}\}$ sind kontextfrei, ihr Schnitt $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ aber nicht.

Ein paar Umformungen kontextfreier Grammatiken

- Nützlich für den Beweis des Pumping-Lemmas für kontextfreie Sprachen.
- Nützlich für die Lösung des Wortproblems für kontextfreie Sprachen.

Eliminierung von λ-Produktionen

Eine Produktion der Form

$$A ::= \lambda$$

heißt λ-Produktion.

Satz

Für jede kontextfreie Grammatik G gibt es eine kontextfreie Grammatik $G_{\lambda-\mathrm{frei}}$ ohne λ-Produktionen, so dass

$$L(G) = L(G_{\lambda-\mathrm{frei}}) \setminus \{\lambda\}.$$

Beispiel

$$S ::= AB, \quad A ::= aAA|\lambda, \quad B ::= bBB|\lambda$$

---

1. Sammeln aller $A \in N$ mit $A \xrightarrow{*} \lambda$:
   - Alle $A \in N$, aus denen λ direkt ableitbar ist:

$$M_0 = \{A, B\}$$

   - Hinzufügung aller $A \in N$, deren rechte Seiten nur aus Zeichen aus M_0 zusammengesetzt sind:

$$M_1 = M_0 \cup \{S\} = \{A, B, S\}$$

$$S ::= AB, \quad A ::= aAA|\lambda, \quad B ::= bBB|\lambda$$

- Hinzufügung aller $A \in N$, deren rechte Seiten nur aus Zeichen aus M_1 zusammengesetzt sind:

$$M_2 = M_1 \cup \emptyset = \{A, B, S\}$$

- Alle $A \in N$, aus denen λ in beliebig vielen Schritten ableitbar ist:

$$M = M_0 \cup M_1 \cup M_2 \cup M_3 \cdots = \{A, B, S\}$$

$$S ::= AB, \quad A ::= aAA|\lambda, \quad B ::= bBB|\lambda$$

---

2. Sukzessives Erweitern der Regelmenge durch Streichen der Zeichen $A \in M = \{A, B, S\}$ aus den rechten Regelseiten:
   - $P_0 = \{S ::= AB, A ::= aAA|\lambda, B ::= bBB|\lambda\}$
   - $P_1 = P_0 \cup \{S ::= A|B, A ::= aA, B ::= bB\}$
   - $P_2 = P_1 \cup \{S ::= \lambda, A ::= a, B ::= b\}$
   - $P_3 = P_2 \cup \emptyset = P_2$
   - $P' = P_2 = \{S ::= AB|A|B|\lambda, A ::= aAA|aA|a|\lambda, B ::= bBB|bB|b|\lambda\}$

3. Löschen aller λ-Produktionen aus P':

Aus

$$\{S ::= AB|A|B|\lambda, A ::= aAA|aA|a|\lambda, B ::= bBB|bB|b|\lambda\}$$

erhalten wir

$$P_{\lambda-\mathrm{frei}} = \{S ::= AB|A|B, A ::= aAA|aA|a, B ::= bBB|bB|b\}.$$

Konstruktion von $G_{\lambda-\text{frei}}$

Sei $G = (N, T, P, S)$ eine kontextfreie Grammatik.

1. (Sammeln aller $A \in N$ mit $A \xrightarrow{*} \lambda$.)

$M' := \{A \in N \mid (A ::= \lambda) \in P\}$

repeat

$M := M';$

$M' := M \cup \{A \in N \mid (A ::= w) \in P, w \in M^*\}$

until $M' = M$

2. (Erweitern der Regelmenge)

$X := P$
repeat
$P' := X;$
$X := P' \cup \{A ::= u_1 u_2 \mid (A ::= u_1 B u_2) \in P', B \in M\}$
until $P' = X$

3. (Konstruktion von $G_{\lambda-\mathrm{frei}}$)

$$G_{\lambda-\mathrm{frei}} = (N, T, P_{\lambda-\mathrm{frei}}, S)$$

mit $P_{\lambda-\mathrm{frei}} = X \setminus \{A ::= \lambda \mid A \in N\}$.

Eliminierung von Kettenregeln

Eine Produktion der Form $A ::= B$ mit $B \in N$ heißt Kettenregel.

Satz

Für jede kontextfreie Grammatik G gibt es eine kontextfreie Grammatik $G_{KR-frei}$ ohne Kettenregeln, so dass

$$L(G) = L(G_{KR-frei}).$$

Idee

1. M: Menge aller Paare $(A, B) \in N \times N$ mit $A \xrightarrow{*} B$ (algorithmisch konstruierbar)

Beispiel

$$\boxed{S ::= A, A ::= B, B ::= b}$$

$$M = \{(S, S), (A, A), (B, B), (S, A), (A, B), (S, B)\}$$

2. $G_{KR-frei} = (N, T, P_{KR-frei}, S)$ mit

$$P_{KR-frei} = \{A ::= r \mid (A, B) \in M, B ::= r \in P, r \notin N\}.$$

Beispiel

$$\boxed{S ::= A, A ::= B, B ::= b}$$

$$M = \{(S, S), (A, A), (B, B), (S, A), (A, B), (S, B)\}$$

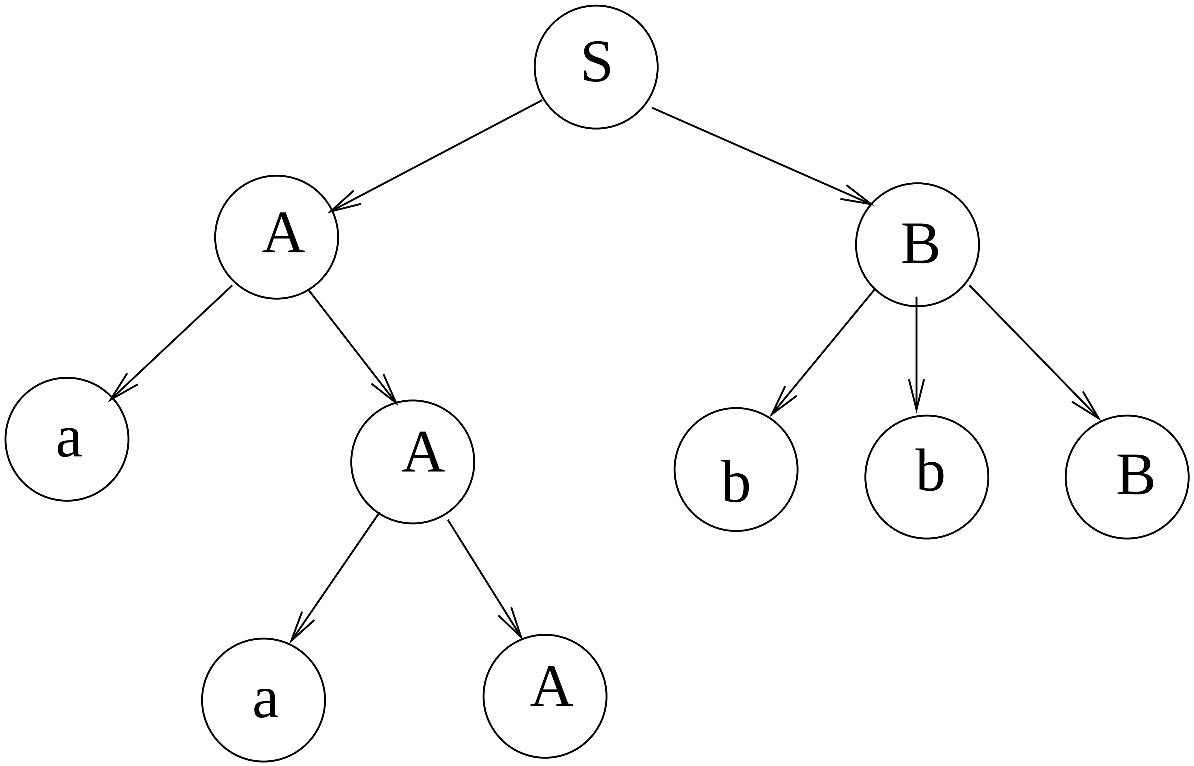
$$P_{KR-frei} : \; S ::= b, A ::= b, B ::= b$$

Ableitungsbäume

- Veranschaulichen Ableitungen kontextfreier Grammatiken
- Nützlich für den Beweis des Pumping-Lemmas für kontextfreie Sprachen
- Konstuktion beruht auf dem Kontextfreiheitslemma
- Ermöglichen die Nutzung bekannter Baumeigenschaften für die Analyse von Ableitungen
- Spielen eine wichtige Rolle im Compilerbau

Baumdarstellung von Ableitungen

- $S \underset{S ::= AB}{\longrightarrow} AB \underset{A ::= aA}{\longrightarrow} aAB \underset{A ::= aA}{\longrightarrow} a^2AB \underset{B ::= bbB}{\longrightarrow} a^2AbbB$



Definition

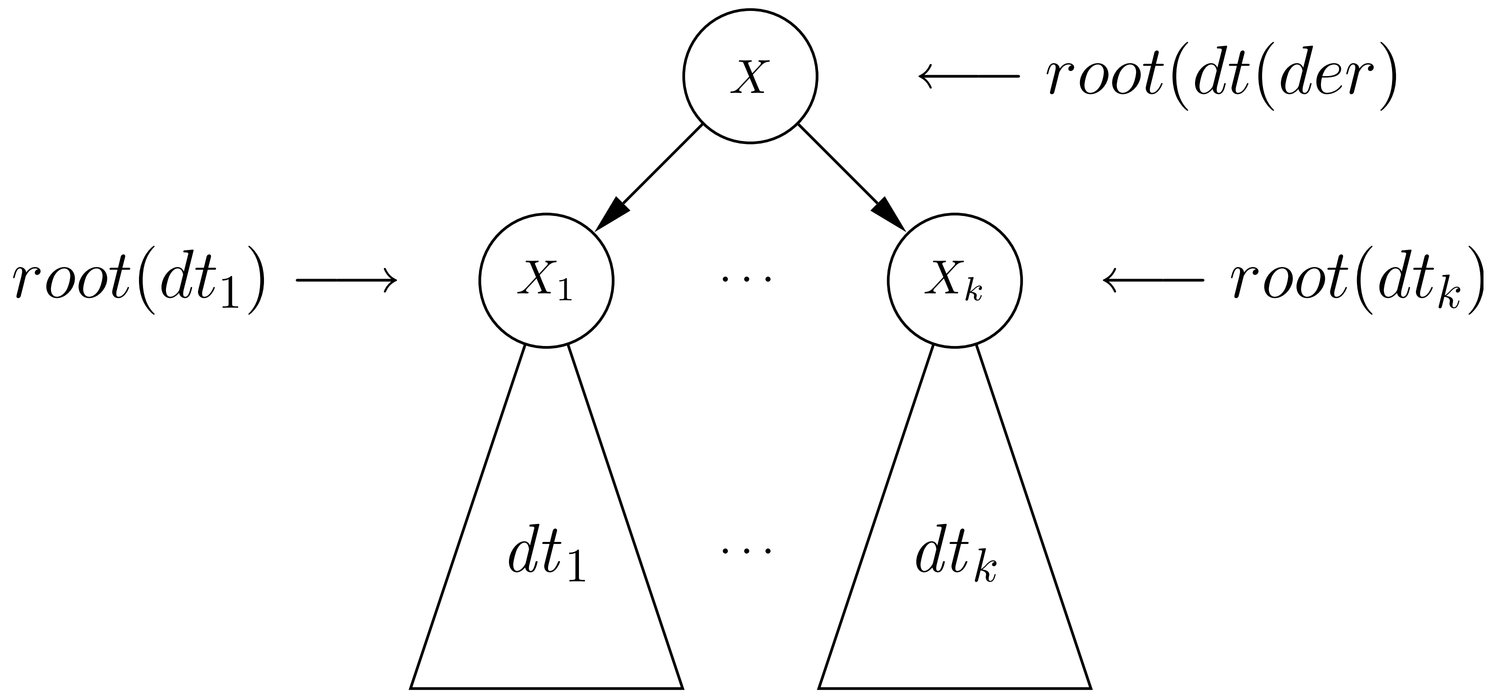
- Sei $G = (N, T, P, S)$ eine kontextfreie Grammatik ohne λ-Produktionen.
- Sei $X \xrightarrow[P]{n} w$ mit $X \in N \cup T$.

Dann ist der Ableitungsbaum $dt(X \xrightarrow[P]{*} w)$ rekursiv wie folgt definiert:

1. $dt(X \xrightarrow[P]{0} X)$: $\quad X \quad \longleftarrow root(dt(X \xrightarrow[P]{0} X))$

   - $height(dt(X \xrightarrow[P]{0} X)) = 0$ und $res(dt(X \xrightarrow[P]{0} X)) = X$

2. Sei $der : X \xrightarrow[P]{} X_1 \dots X_k \xrightarrow{n} w$. Seien $X_i \xrightarrow[P]{n_i} w_i$ $(n_i \leq n)$ die korrespondierenden Ableitungen gemäß dem Kontextfreiheitslemma. Sei $dt_i = dt(X_i \xrightarrow[P]{n_i} w_i)$. Dann hat der Ableitungsbaum $dt(der)$ die Form



- $height(dt(der)) = 1 + \max\{height(dt_i) \mid i = 1, \ldots, k\}$
- $res(dt(der)) = res(dt_1) \cdots res(dt_k)$.

Beziehung zwischen Baumhöhe und Resultatslänge

Beobachtung 1

Sei b die Länge der längsten rechten Seite von Produktionen aus P. Dann gilt für alle Ableitungsbäume dt:

$$length(res(dt)) \leq b^{height(dt)}.$$

Ein langer Weg von der Wurzel zu einem Blatt

Beobachtung 2

In jedem Ableitungsbaum gibt es mindestens einen Weg von der Wurzel zu einem Blatt, der so lang ist, wie der Baum hoch ist.