Teil IV

Kontextfreie Sprachen: Pumping-Lemma und Chomsky-Normalform

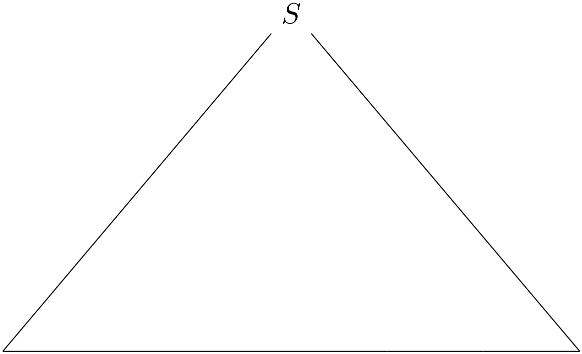
Pumping-Lemma für CFL

Lemma

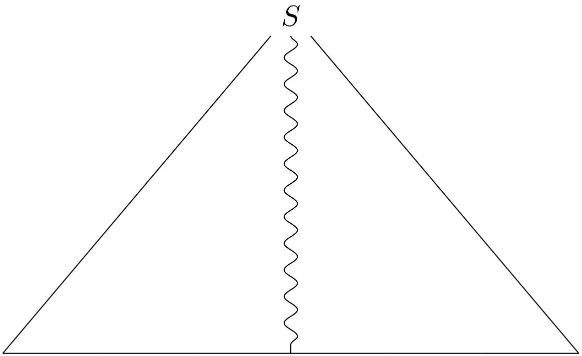
Sei Σ Alphabet. Für jede kontextfreie Sprache $L \subseteq \Sigma^*$ existiert eine **Pumpkonstante** $n_L \in \mathbb{N}_{>0}$, so dass für jedes Wort $z \in L$ mit $|z| \geq n_L$ gilt:
Es existieren Wörter $u, v, w, x, y \in \Sigma^*$ mit

1. $uvwxy = z$,
2. $|vx| \geq 1$,
3. $|vwx| \leq n_L$, und
4. für alle $i \in \mathbb{N}$ ist $uv^iwx^iy \in L$.

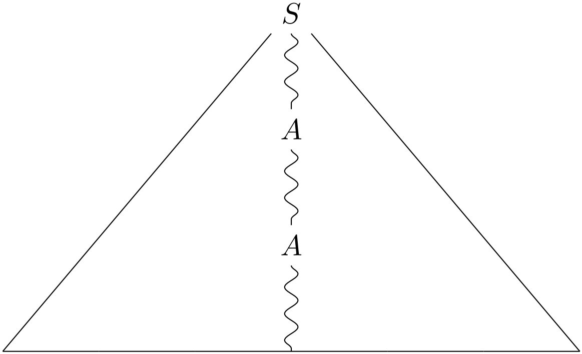
Pumping-Lemma für CFL, Beweis(1/3)



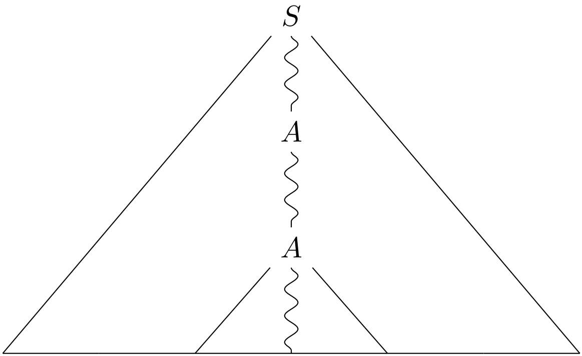
Pumping-Lemma für CFL, Beweis(1/3)



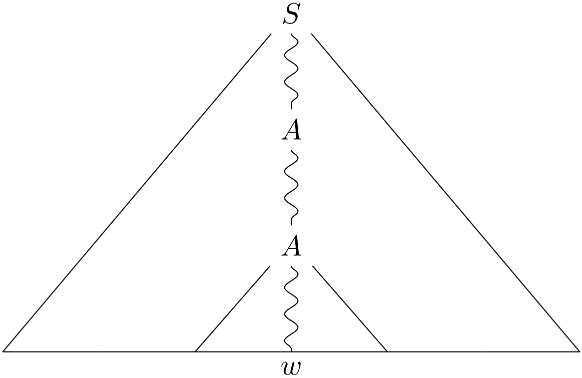
Pumping-Lemma für CFL, Beweis(1/3)



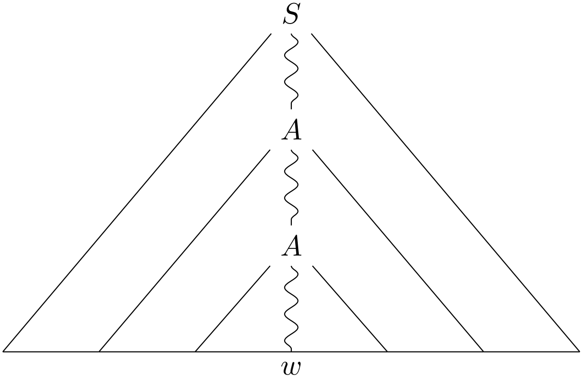
Pumping-Lemma für CFL, Beweis(1/3)



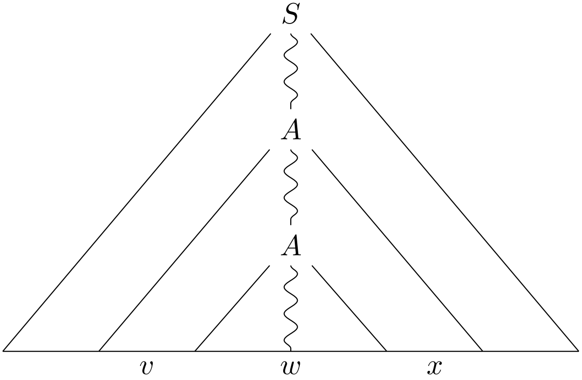
Pumping-Lemma für CFL, Beweis(1/3)



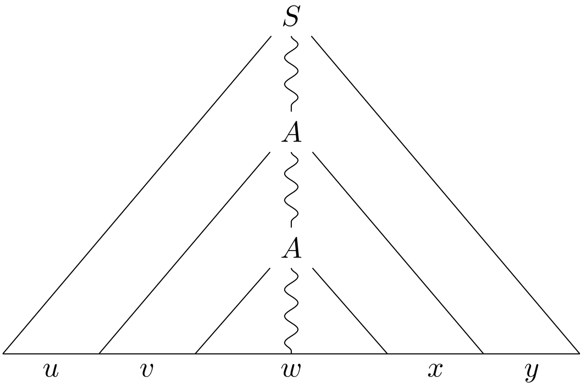
Pumping-Lemma für CFL, Beweis(1/3)



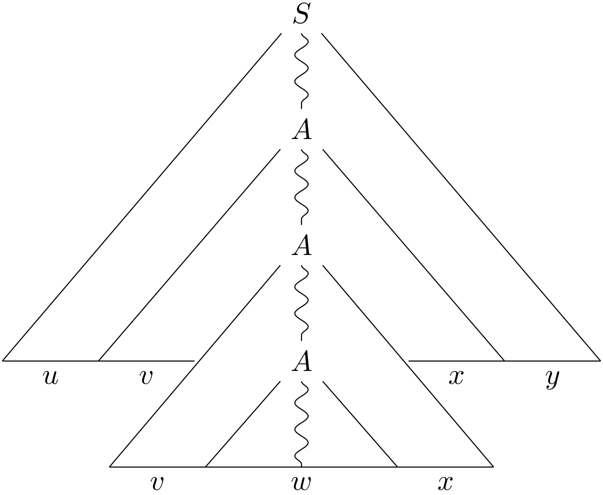
Pumping-Lemma für CFL, Beweis(1/3)



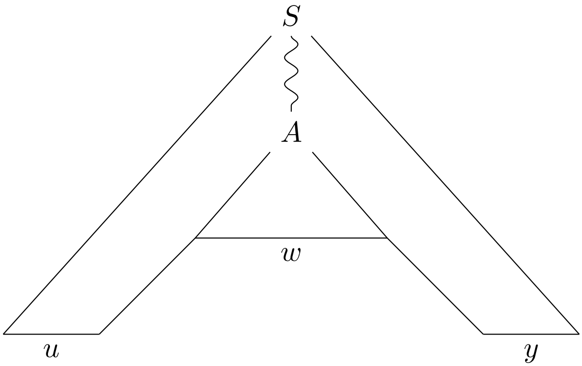
Pumping-Lemma für CFL, Beweis(1/3)



Pumping-Lemma für CFL, Beweis (2/3)



Pumping-Lemma für CFL, Beweis (3/3)



Anwendungen

Nicht kontextfrei:

$L_1 := \{\mathtt{a}^i\mathtt{b}^i\mathtt{c}^i \mid i \in \mathbb{N}\},$

Anwendungen

Nicht kontextfrei:

$$L_1 := \{\mathtt{a}^i\mathtt{b}^i\mathtt{c}^i \mid i \in \mathbb{N}\},$$

$$L_2 := \{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^i\mathtt{b}^j \mid i, j \in \mathbb{N}\}.$$

Anwendungen

Nicht kontextfrei:

$$L_1 := \{\mathtt{a}^i\mathtt{b}^i\mathtt{c}^i \mid i \in \mathbb{N}\},$$
$$L_2 := \{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^i\mathtt{b}^j \mid i, j \in \mathbb{N}\}.$$

Lemma

Die Klasse CFL ist nicht abgeschlossen unter:

Anwendungen

Nicht kontextfrei:

$$L_1 := \{\mathtt{a}^i\mathtt{b}^i\mathtt{c}^i \mid i \in \mathbb{N}\},$$
$$L_2 := \{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^i\mathtt{b}^j \mid i, j \in \mathbb{N}\}.$$

Lemma

Die Klasse CFL ist nicht abgeschlossen unter:

- Schnitt,

Anwendungen

Nicht kontextfrei:

$$L_1 := \{\mathtt{a}^i\mathtt{b}^i\mathtt{c}^i \mid i \in \mathbb{N}\},$$

$$L_2 := \{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^i\mathtt{b}^j \mid i, j \in \mathbb{N}\}.$$

Lemma

Die Klasse CFL ist nicht abgeschlossen unter:

- Schnitt,
- Komplement,

Anwendungen

Nicht kontextfrei:

$$L_1 := \{\mathtt{a}^i\mathtt{b}^i\mathtt{c}^i \mid i \in \mathbb{N}\},$$

$$L_2 := \{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^i\mathtt{b}^j \mid i, j \in \mathbb{N}\}.$$

Lemma

Die Klasse CFL ist nicht abgeschlossen unter:

- Schnitt,
- Komplement,
- Mengendifferenz.

Anwendungen

Nicht kontextfrei:

$$L_1 := \{\mathsf{a}^i\mathsf{b}^i\mathsf{c}^i \mid i \in \mathbb{N}\},$$

$$L_2 := \{\mathsf{a}^i\mathsf{b}^j\mathsf{a}^i\mathsf{b}^j \mid i, j \in \mathbb{N}\}.$$

Lemma

Die Klasse CFL ist nicht abgeschlossen unter:

- Schnitt,
- Komplement,
- Mengendifferenz.

Satz

Sei Σ Alphabet mit $|\Sigma| = 1$. Dann ist

$$\mathrm{REG}_\Sigma = \mathrm{CFL}_\Sigma$$

Die Chomsky-Normalform

Definition

Eine CFG $G := (\Sigma, V, P, S)$ ist in **Chomsky-Normalform (CNF)**, wenn jede Regel aus P eine der folgenden Formen hat:

Die Chomsky-Normalform

Definition

Eine CFG $G := (\Sigma, V, P, S)$ ist in **Chomsky-Normalform (CNF)**, wenn jede Regel aus P eine der folgenden Formen hat:

$$
\begin{aligned}
&A \to BC && \text{mit } A \in V,\ B, C \in (V - \{S\}),\\
&A \to a && \text{mit } A \in V,\ a \in \Sigma,\\
&S \to \varepsilon.
\end{aligned}
$$

Die Chomsky-Normalform

Definition

Eine CFG $G := (\Sigma, V, P, S)$ ist in **Chomsky-Normalform (CNF)**, wenn jede Regel aus P eine der folgenden Formen hat:

$$
\begin{aligned}
&A \to BC && \text{mit } A \in V,\ B, C \in (V - \{S\}), \\
&A \to a && \text{mit } A \in V,\ a \in \Sigma, \\
&S \to \varepsilon.
\end{aligned}
$$

1. nur Startsymbol darf gelöscht werden
2. Startsymbol darf nicht rechts stehen
3. rechts nur genau zwei Variablen oder genau ein Terminal

CNF und CFLs

Satz

Jede kontextfreie Sprache wird von einer CFG in Chomsky-Normalform erzeugt.

CNF und CFLs

Satz

Jede kontextfreie Sprache wird von einer CFG in Chomsky-Normalform erzeugt.

Idee

Wandele beliebige CFG um in CFG in Chomsky-Normalform.

Der CNF-Algorithmus

Algorithmus `CNF`

Eingabe: CFG $G := (\Sigma, V, P, S)$
Ausgabe: CFG G_C in Chomsky-Normalform mit $\mathcal{L}(G_C) = \mathcal{L}(G)$

1. Führe ein neues Startsymbol S_0 ein.

Der CNF-Algorithmus

Algorithmus `CNF`

Eingabe: CFG $G := (\Sigma, V, P, S)$
Ausgabe: CFG G_C in Chomsky-Normalform mit $\mathcal{L}(G_C) = \mathcal{L}(G)$

1. Führe ein neues Startsymbol S_0 ein.
2. Ersetze alle ε-Regeln.

Der CNF-Algorithmus

Algorithmus `CNF`

Eingabe: CFG $G := (\Sigma, V, P, S)$
Ausgabe: CFG G_C in Chomsky-Normalform mit $\mathcal{L}(G_C) = \mathcal{L}(G)$

1. Führe ein neues Startsymbol S_0 ein.
2. Ersetze alle ε-Regeln.
3. Ersetze alle Ein-Variablen-Regeln.

Der CNF-Algorithmus

Algorithmus `CNF`

Eingabe: CFG $G := (\Sigma, V, P, S)$
Ausgabe: CFG G_C in Chomsky-Normalform mit $\mathcal{L}(G_C) = \mathcal{L}(G)$

1. Führe ein neues Startsymbol S_0 ein.
2. Ersetze alle ε-Regeln.
3. Ersetze alle Ein-Variablen-Regeln.
4. Zerlege alle rechten Seiten, die die falsche Form haben.

Der CNF-Algorithmus

Algorithmus CNF

Eingabe: CFG $G := (\Sigma, V, P, S)$
Ausgabe: CFG G_C in Chomsky-Normalform mit $\mathcal{L}(G_C) = \mathcal{L}(G)$

1. Führe ein neues Startsymbol S_0 ein.
2. Ersetze alle ε-Regeln.
3. Ersetze alle Ein-Variablen-Regeln.
4. Zerlege alle rechten Seiten, die die falsche Form haben.
5. Gib Grammatik als G_C aus.

(Eine) Anwendung der CNF

Lemma

Die Klasse CFL ist abgeschlossen unter Schnitt mit REG.

Mit anderen Worten

Aus $L \in \mathrm{CFL}$ und $R \in \mathrm{REG}$ folgt

$$(L \cap R) \in \mathrm{CFL}\,.$$

(Eine) Anwendung der CNF

Lemma

Die Klasse CFL ist abgeschlossen unter Schnitt mit REG.

Mit anderen Worten

Aus $L \in \mathrm{CFL}$ und $R \in \mathrm{REG}$ folgt

$$(L \cap R) \in \mathrm{CFL}\,.$$

Anwendung

Sei $|\Sigma| \geq 2$,
$\mathrm{COPY}_\Sigma := \{ww \mid w \in \Sigma^*\}$.
Dann gilt:

$$\mathrm{COPY}_\Sigma \notin \mathrm{CFL}\,.$$