

Übung 4: Pumping-Lemmata, CNF und CYK

Theoretische Informatik Sommersemester 2014

Markus Kaiser

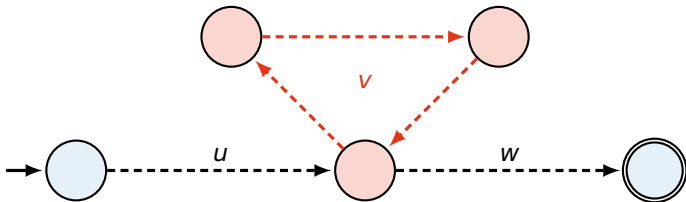
11. Mai 2014

Satz (Pumping Lemma für reguläre Sprachen)

Sei $R \subseteq \Sigma^*$ regulär.

Dann gibt es ein $n > 0$, so dass sich **jedes** $z \in R$ mit $|z| \geq n$ so in $z = uvw$ zerlegen lässt, dass

- $v \neq \epsilon$
- $|uv| \leq n$
- $\forall i \geq 0. uv^i w \in R$



Definition

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

Ein Symbol $X \in V \cup \Sigma$ ist

nützlich es gibt $S \rightarrow_G^* w \in \Sigma^*$ in der X **vorkommt**

erzeugend es gibt $X \rightarrow_G^* w \in \Sigma^*$

erreichbar es gibt $S \rightarrow_G^* \alpha X \beta$

Satz

Nützliche Symbole **sind** erzeugend und erreichbar. Aber **nicht** notwendigerweise umgekehrt.

$$S \rightarrow AB \mid a, \quad A \rightarrow b$$

Definition (Chomsky-Normalform)

Eine kontextfreie Grammatik ist in **Chomsky-Normalform** (CNF) genau dann wenn alle Produktionen die Form

$$A \rightarrow a \quad \text{oder} \quad A \rightarrow BC$$

haben.

Satz

Zu *jeder* CFG G existiert eine CFG G' in Chomsky-Normalform mit

$$L(G') = L(G) \setminus \{\epsilon\}$$

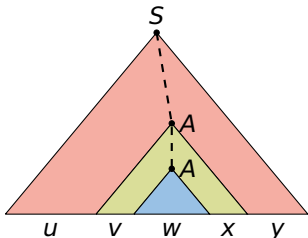
Satz (Pumping Lemma für kontextfreie Sprachen)

Sei $L \subseteq \Sigma^*$ kontextfrei.

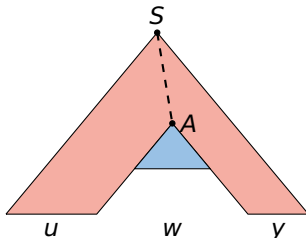
Dann gibt es ein $n > 0$, so dass sich **jedes** $z \in L$ mit $|z| \geq n$ so in $z = uvwxy$ zerlegen lässt, dass

- $vx \neq \epsilon$
- $|vwx| \leq n$
- $\forall i \geq 0. uv^iwx^i y \in L$

$S \rightarrow^* uAy \rightarrow^* uvAxy \rightarrow^* uvwxy$



$S \rightarrow^* uAy \rightarrow^* uwy$



CNF Konstruktion

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

- 1 **Eliminiere ϵ -Produktionen**
- 2 **Eliminiere Kettenproduktionen**
- 3 **Ersetze Terminale** durch Nichtterminale
- 4 **Verkürze Ketten** von Nichtterminalen der Länge ≥ 3

CNF Konstruktion

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

- 1 Eliminiere ϵ -Produktionen
- 2 Eliminiere Kettenproduktionen
- 3 Ersetze Terminale durch Nichtterminale
- 4 Verkürze Ketten von Nichtterminalen der Länge ≥ 3

Sind $B \rightarrow \epsilon$ und $A \rightarrow \alpha B \beta$ in P , dann füge $A \rightarrow \alpha \beta$ hinzu. Entferne danach alle ϵ -Produktionen.

$$S \rightarrow Ab, \quad A \rightarrow aAA \mid \epsilon$$

wird zu:

$$S \rightarrow Ab \mid b$$

$$A \rightarrow aAA \mid aA \mid a$$

CNF Konstruktion

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

- 1 Eliminiere ϵ -Produktionen
- 2 Eliminiere Kettenproduktionen
- 3 Ersetze Terminale durch Nichtterminale
- 4 Verkürze Ketten von Nichtterminalen der Länge ≥ 3

Sind $A \rightarrow B$ und $B \rightarrow \alpha$ in P , dann füge $A \rightarrow \alpha$ hinzu. Entferne danach alle Kettenproduktionen und unerreichbaren Symbole.

$$S \rightarrow A, \quad A \rightarrow a \mid B, \quad B \rightarrow bS$$

wird zu:

$$A \rightarrow a \mid bS$$

$$S \rightarrow a \mid bS$$

CNF Konstruktion

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

- 1 Eliminiere ϵ -Produktionen
- 2 Eliminiere Kettenproduktionen
- 3 Ersetze Terminale durch Nichtterminale
- 4 Verkürze Ketten von Nichtterminalen der Länge ≥ 3

Ersetze jedes $a \in \Sigma$ in einer rechten Seite länger als 1 durch ein neues Nichtterminal.

$$S \rightarrow aa \mid Bb \mid b, \quad B \rightarrow \dots$$

wird zu:

$$S \rightarrow X_a X_a \mid B X_b \mid b$$

$$X_a \rightarrow a, \quad X_b \rightarrow b$$

CNF Konstruktion

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

- 1 Eliminiere ϵ -Produktionen
- 2 Eliminiere Kettenproduktionen
- 3 Ersetze Terminale durch Nichtterminale
- 4 Verkürze Ketten von Nichtterminalen der Länge ≥ 3

Ersetze jede Produktion der Form $A \rightarrow B_1 B_2 \dots B_k$ durch neue Nichtterminale mit Produktionen der Länge 2.

$$S \rightarrow X_a X_b B X_a, \quad X_a \rightarrow a, \quad X_b \rightarrow b, \quad B \rightarrow \dots$$

wird zu:

$$\begin{aligned} S &\rightarrow X_a T_1 \\ T_1 &\rightarrow X_b T_2, \quad T_2 \rightarrow B X_a \end{aligned}$$

Definition (Cocke-Younger-Kasami-Algorithmus)

Der **CYK-Algorithmus** entscheidet das Wortproblem für kontextfreie Grammatiken in Chomsky-Normalform in $\mathcal{O}(n^3)$.

Gegeben eine **Grammatik** $G = (V, \Sigma, P, S)$ in CNF und ein **Wort** $w = a_1 \dots a_n \in \Sigma^*$. Mit

$$V_{ij} := \{A \in V \mid A \rightarrow_G^* a_i \dots a_j\}$$

ist

$$w \in L(G) \Leftrightarrow S \in V_{1n}$$

$$V_{ii} = \{A \in V \mid (A \rightarrow a_i) \in P\}$$

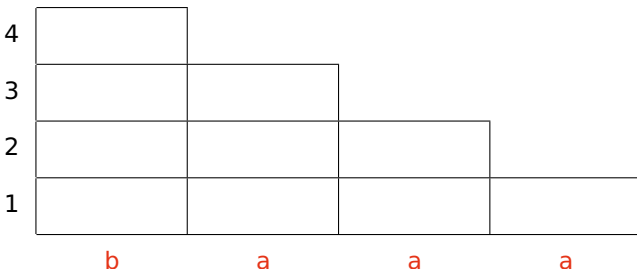
$$V_{ij} = \{A \in V \mid \exists k, B \in V_{ik}, C \in V_{k+1,j} \cdot (A \rightarrow BC) \in P\}$$

CYK-Algorithmus

Kombiniere **Teilwörter** zum ganzen Wort, wenn möglich.

- 1 Initialisiere mit den V_{ij} .
- 2 Befülle die Tabelle von unten nach oben.

$S \rightarrow AB \mid BC, \quad A \rightarrow BA \mid a, \quad B \rightarrow CC \mid b, \quad C \rightarrow AB \mid a$



CYK-Algorithmus

Kombiniere **Teilwörter** zum ganzen Wort, wenn möglich.

- 1 Initialisiere mit den V_{ij} .
- 2 Befülle die Tabelle von unten nach oben.

$S \rightarrow AB \mid BC, \quad A \rightarrow BA \mid a, \quad B \rightarrow CC \mid b, \quad C \rightarrow AB \mid a$

4				
3				
2				
1	B	A, C	A, C	A, C
	b	a	a	a

CYK-Algorithmus

Kombiniere **Teilwörter** zum ganzen Wort, wenn möglich.

- 1 Initialisiere mit den V_{ij} .
- 2 Befülle die Tabelle von unten nach oben.

$S \rightarrow AB \mid BC, \quad A \rightarrow BA \mid a, \quad B \rightarrow CC \mid b, \quad C \rightarrow AB \mid a$

4				
3				
2	A	B	B	
1	B	A,C	A,C	A,C
	b	a	a	a

CYK-Algorithmus

Kombiniere **Teilwörter** zum ganzen Wort, wenn möglich.

- 1 Initialisiere mit den V_{ij} .
- 2 Befülle die Tabelle von unten nach oben.

$S \rightarrow AB \mid BC, \quad A \rightarrow BA \mid a, \quad B \rightarrow CC \mid b, \quad C \rightarrow AB \mid a$

4				
3	\emptyset	S, A, C		
2	A	B	B	
1	B	A, C	A, C	A, C
	b	a	a	a

CYK-Algorithmus

Kombiniere **Teilwörter** zum ganzen Wort, wenn möglich.

- 1 Initialisiere mit den V_{ij} .
- 2 Befülle die Tabelle von unten nach oben.

$S \rightarrow AB \mid BC, \quad A \rightarrow BA \mid a, \quad B \rightarrow CC \mid b, \quad C \rightarrow AB \mid a$

4	S, ...			
3	∅	S, A, C		
2	A	B	B	
1	B	A, C	A, C	A, C
	b	a	a	a