

Übung 7: CYK und Kellerautomaten

Theoretische Informatik Sommersemester 2013

Markus Kaiser

June 10, 2013

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

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

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

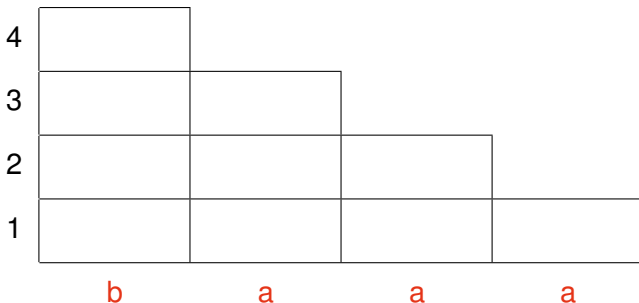
$$V_{ij} = \{A \in V \mid \exists k, B \in V_{ik}, C \in V_{k+1,j} \cdot (A \rightarrow BC) \in P\}$$

Idee

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

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

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



Idee

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

Idee

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

Idee

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

Idee

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

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

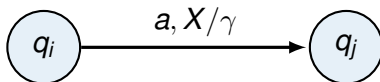
$S \rightarrow AB \mid BC, \quad A \rightarrow BA \mid a, \quad B \rightarrow CC \mid b, \quad C \rightarrow AB \mid a$

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

Definition (Kellerautomat)

Ein **PDA** (Push-Down-Automat) ist ein Tupel $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ aus einer/einem

- endlichen Menge von **Zuständen** Q
- endlichen **Eingabealphabet** Σ
- endlichen **Kelleralphabet** Γ
- **Übergangsfunktion** $\delta : Q \times \Sigma \times \Gamma \mapsto P(Q \times \Gamma^*)$
- **Startzustand** $q_0 \in Q$
- **Kellerinitialisierung** $Z_0 \in \Gamma$
- Menge von **Endzuständen** $F \subseteq Q$



Definition (Kellerautomat)

Ein **PDA** (Push-Down-Automat) ist ein Tupel $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ aus einer/einem

- **Übergangsfunktion** $\delta : Q \times \Sigma \times \Gamma \mapsto P(Q \times \Gamma^*)$

Definition (Akzeptanz)

Ein PDA P akzeptiert $w \in \Sigma^*$ **mit Endzustand** gdw

$$\exists f \in F, \gamma \in \Gamma^*. (q_0, w, Z_0) \rightarrow_P^* (f, \epsilon, \gamma)$$

Ein PDA P akzeptiert $w \in \Sigma^*$ **mit leerem Keller** gdw

$$\exists q \in Q. (q_0, w, Z_0) \rightarrow_P^* (q, \epsilon, \epsilon)$$

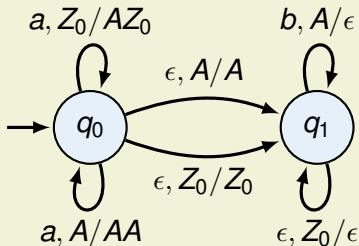
Definition (Kellerautomat)

Ein **PDA** (Push-Down-Automat) ist ein Tupel $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ aus einer/einem

- **Übergangsfunktion** $\delta : Q \times \Sigma \times \Gamma \mapsto P(Q \times \Gamma^*)$

Beispiel

PDA akzeptierend **mit leerem Keller** zu $L = \{a^n b^n \mid n \in \mathbb{N}\}$.



CFG \longleftrightarrow CNF \longleftrightarrow PDA _{ϵ} \longleftrightarrow PDA _{F}

■ Abschlusseigenschaften

	Schnitt	Vereinigung	Komplement	Produkt	Stern
REG	ja	ja	ja	ja	ja
CFL	nein	ja	nein	ja	ja

■ Entscheidbarkeit

	Wortproblem	Leerheit	Äquivalenz	Schnittproblem
DFA	$\mathcal{O}(n)$	ja	ja	ja
CFG	$\mathcal{O}(n^3)$	ja	nein	nein

CFG \longleftrightarrow CNF \longleftrightarrow PDA_ε \longleftrightarrow PDA_F

■ Abschlusseigenschaften

	Schnitt	Vereinigung	Komplement	Produkt	Stern
REG	ja	ja	ja	ja	ja
CFL	nein	ja	nein	ja	ja

■ Entscheidbarkeit

	Wortproblem	Leerheit	Äquivalenz	Schnittproblem
DFA	$\mathcal{O}(n)$	ja	ja	ja
CFG	$\mathcal{O}(n^3)$	ja	nein	nein