Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Vorlesung am 27.01.2011

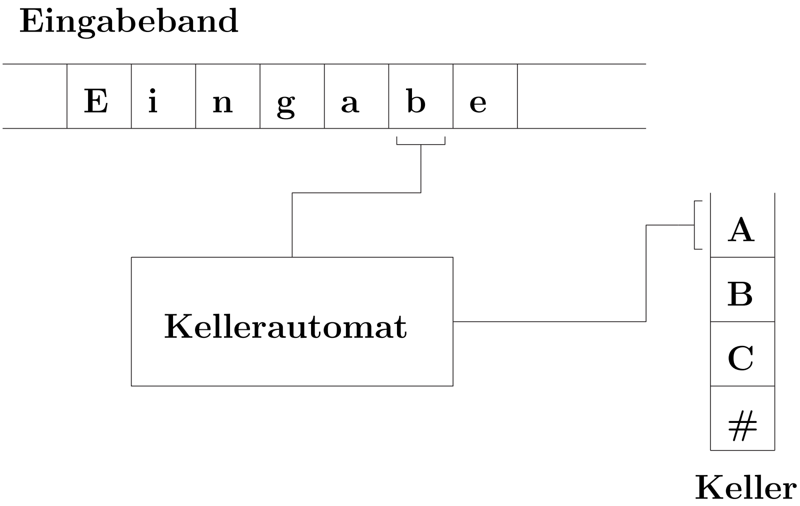
INSTITUT FÜR THEORETISCHE INFORMATIK

Kellerautomaten

Ein (nichtdeterministischer) **Kellerautomat** (NPDA bzw. PDA, Push-down Automaton) besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet
- Γ endliches STACK-Alphabet
- $q_0 \in Q$ Anfangszustand
- $Z_0 \in \Gamma$ Initialisierung des STACK
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$, d.h.
  - $\delta(q, a, Z) \subseteq \{(q, \gamma) : q \in Q, \gamma \in \Gamma^*\}$
  - $\delta(q, \varepsilon, Z) \subseteq \{(q, \gamma) : q \in Q, \gamma \in \Gamma^*\}$
- $F \subseteq Q$ Menge der akzeptierenden Endzustände, $F = \emptyset$ ist möglich.

Kellerautomaten - Visualisierung



Kellerautomaten - Arbeitsweise

Eine **Konfiguration eines PDA** ist ein Tripel (q, w, α) mit
- $q \in Q$,
- $w \in \Sigma^*$ der Teil der Eingabe, der noch nicht gelesen wurde,
- $\alpha \in \Gamma^*$ STACK-Inhalt.

Zu Konfiguration $(q, w_1 \dots w_k, Z_1 \dots Z_m)$ gibt es die **Nachfolgekonfigurationen**:

$$(q', w_2 \dots w_k, Z_1' \dots Z_r' Z_2 \dots Z_m) \text{ für alle } (q', Z_1' \dots Z_r') \in \delta(q, w_1, Z_1)$$

und

$$(q', w_1 \dots w_k, Z_1' \dots Z_r' Z_2 \dots Z_m) \text{ für alle } (q', Z_1' \dots Z_r') \in \delta(q, \varepsilon, Z_1).$$

Kellerautomaten - Arbeitsweise

Ein PDA **akzeptiert** ein $w \in \Sigma^*$ **durch leeren Stack**, wenn es eine zulässige Folge von Konfigurationen aus der Anfangskonfiguration (q_0, w, Z_0) in eine Konfiguration $(q, \varepsilon, \varepsilon)$, $q \in Q$, gibt.

Ein PDA **akzeptiert** ein $w \in \Sigma^*$ **durch einen akzeptierenden Endzustand**, wenn es eine zulässige Folge von Konfigurationen aus der Anfangskonfiguration (q_0, w, Z_0) in eine Konfiguration (q, ε, γ) mit $q \in F$ und $\gamma \in \Gamma^*$ gibt.

Ein PDA ist **deterministisch** (DPDA), falls

$$|\delta(q, a, Z)| + |\delta(q, \varepsilon, Z)| \leq 1$$

für alle $q \in Q$, $a \in \Sigma$, $Z \in \Gamma$.

Kellerautomaten - Beispiel

Ein DPDA für $L = \{w\#w^R | w \in \{0,1\}^*\}$. Informelle Beschreibung:

- Betrachte beliebiges Wort $w_1 \dots w_n \# w_n \dots w_1 \in L$.

Phase 1

- Lies $w_1 \dots w_n$ und schreibe jeweils w_i auf den STACK bis $\#$ gelesen.

Phase 2

- Lies $w_n \dots w_1$ und vergleiche den jeweils gelesenen Buchstaben mit dem jeweils obersten Buchstaben auf dem STACK.
  - Gleichheit: Nimm obersten Buchstaben vom STACK
  - Sonst: Stoppe in nichtakzeptierenden Zustand

Phase 3

- Ist nur noch Z_0 auf dem STACK
  - Entferne Z_0
  - Akzeptiere die Eingabe „mit leerem STACK“

Kellerautomaten - Beispiel

Karlsruher Institut für Technologie

Ein DPDA für $L = \{w\#w^R | w \in \{0,1\}^*\}$.

$$(Q = \{q_0, q_1, q_2\},\ \Sigma = \{0,1,\#\},\ \Gamma = \Sigma \cup \{Z_0\}),\ q_0,\ Z_0,\ \delta,\ \emptyset)$$

$\delta(q_0, 0, Z_0)$	=	$(q_1, 0Z_0)$	Phase 1
$\delta(q_0, 1, Z_0)$	=	$(q_1, 1Z_0)$	
$\delta(q_1, 0, 0)$	=	$(q_1, 00)$	
$\delta(q_1, 0, 1)$	=	$(q_1, 01)$	
$\delta(q_1, 1, 0)$	=	$(q_1, 10)$	
$\delta(q_1, 1, 1)$	=	$(q_1, 11)$	
$\delta(q_1, \#, 0)$	=	$(q_2, 0)$	Trennzeichen gelesen $\Rightarrow$ Zu Phase 2
$\delta(q_1, \#, 1)$	=	$(q_2, 1)$	
$\delta(q_2, 0, 0)$	=	(q_2, ε)	Phase 2
$\delta(q_2, 1, 1)$	=	(q_2, ε)	
$\delta(q_2, \varepsilon, Z_0)$	=	(q_2, ε)	Phase 3

Beispiel - Berechnung für Eingabe 001#100

Karlsruher Institut für Technologie

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[4pt]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[4pt]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[4pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[4pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack

Beispiel - Berechnung für Eingabe 001#100

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[4pt]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[4pt]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[4pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[4pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack
q_0	$001\#100$	Z_0

Beispiel - Berechnung für Eingabe 001#100

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[4pt]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[4pt]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[4pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[4pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack
q_0	$001\#100$	Z_0
q_1	$01\#100$	$0Z_0$

Beispiel - Berechnung für Eingabe 001#100

$$\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0)\\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0)\\[4pt]
\delta(q_1, 0, 0) &= (q_1, 00)\\
\delta(q_1, 0, 1) &= (q_1, 01)\\
\delta(q_1, 1, 0) &= (q_1, 10)\\
\delta(q_1, 1, 1) &= (q_1, 11)\\[4pt]
\delta(q_1, \#, 0) &= (q_2, 0)\\
\delta(q_1, \#, 1) &= (q_2, 1)\\[4pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon)\\
\delta(q_2, 1, 1) &= (q_2, \varepsilon)\\[4pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}$$

Zustand	Eingabe	Stack
q_0	001#100	Z_0
q_1	01#100	$0Z_0$
q_1	1#100	$00Z_0$

Beispiel - Berechnung für Eingabe 001#100

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[6pt]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[6pt]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[6pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[6pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack
q_0	001#100	Z_0
q_1	01#100	$0Z_0$
q_1	1#100	$00Z_0$
q_1	#100	$100Z_0$

Beispiel - Berechnung für Eingabe 001#100

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[1ex]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[1ex]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[1ex]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[1ex]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack
q_0	001#100	Z_0
q_1	01#100	$0Z_0$
q_1	1#100	$00Z_0$
q_1	#100	$100Z_0$
q_2	100	$100Z_0$
q_2	00	$00Z_0$

Beispiel - Berechnung für Eingabe 001#100

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[6pt]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[6pt]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[6pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[6pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack
q_0	001#100	Z_0
q_1	01#100	$0Z_0$
q_1	1#100	$00Z_0$
q_1	#100	$100Z_0$
q_2	100	$100Z_0$
q_2	00	$00Z_0$
q_2	0	$0Z_0$

Beispiel - Berechnung für Eingabe 001#100

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[4pt]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[4pt]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[4pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[4pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack
q_0	001#100	Z_0
q_1	01#100	$0Z_0$
q_1	1#100	$00Z_0$
q_1	#100	$100Z_0$
q_2	100	$100Z_0$
q_2	00	$00Z_0$
q_2	0	$0Z_0$
q_2		Z_0

Beispiel - Berechnung für Eingabe 001#100

$$
\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_1, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, 1Z_0) \\[4pt]
\delta(q_1, 0, 0) &= (q_1, 00) \\
\delta(q_1, 0, 1) &= (q_1, 01) \\
\delta(q_1, 1, 0) &= (q_1, 10) \\
\delta(q_1, 1, 1) &= (q_1, 11) \\[4pt]
\delta(q_1, \#, 0) &= (q_2, 0) \\
\delta(q_1, \#, 1) &= (q_2, 1) \\[4pt]
\delta(q_2, 0, 0) &= (q_2, \varepsilon) \\
\delta(q_2, 1, 1) &= (q_2, \varepsilon) \\[4pt]
\delta(q_2, \varepsilon, Z_0) &= (q_2, \varepsilon)
\end{aligned}
$$

Zustand	Eingabe	Stack
q_0	001#100	Z_0
q_1	01#100	$0Z_0$
q_1	1#100	$00Z_0$
q_1	#100	$100Z_0$
q_2	100	$100Z_0$
q_2	00	$00Z_0$
q_2	0	$0Z_0$
q_2		Z_0

akzeptiert durch leeren Stack

Kellerautomaten - Beispiel 2

Ein NPDA für $L = \{ww^R | w \in \{0,1\}^*\}$. Informelle Beschreibung:

- In diesem Fall fehlt das Trennzeichen.
- Der NPDA funktioniert wie der DPDA aus dem letzten Beispiel
- Der Übergang in Phase 2 funktioniert allerdings nichtdeterministisch.

Bemerkung:

- Für die Sprache $L = \{ww^R | w \in \{0,1\}^*\}$ gibt es keinen DPDA.
- NPDAs können also mehr als DPDAs.

Kellerautomaten - Beispiel 2

Ein NPDA für $L = \{ww^R | w \in \{0,1\}^*\}$.

$$(Q = \{q_0, q_1, q_2\},\ \Sigma = \{0, 1, \#\},\ \Gamma = \Sigma \cup \{Z_0\}),\ q_0,\ Z_0,\ \delta,\ \emptyset)$$

$$
\begin{array}{lcll}
\delta(q_0, 0, Z_0) & = & \{(q_1, 0Z_0)\} & \text{Phase 1} \\
\delta(q_0, 1, Z_0) & = & \{(q_1, 1Z_0)\} & \\
\delta(q_1, 0, 0) & = & \{(q_1, 00), (q_2, \epsilon)\} & \\
\delta(q_1, 0, 1) & = & \{(q_1, 01)\} & \\
\delta(q_1, 1, 0) & = & \{(q_1, 10)\} & \\
\delta(q_1, 1, 1) & = & \{(q_1, 11), (q_2, \epsilon)\} & \\
\delta(q_2, 0, 0) & = & \{(q_2, \varepsilon)\} & \text{Phase 2} \\
\delta(q_2, 1, 1) & = & \{(q_2, \varepsilon)\} & \\
\delta(q_2, \varepsilon, Z_0) & = & \{(q_2, \varepsilon)\} & \text{Phase 3}
\end{array}
$$

Satz:
Zu einem PDA, der eine Sprache L durch einen akzeptierenden Endzustand akzeptiert, kann ein PDA konstruiert werden, der L mit leerem STACK akzeptiert.

Beweis - Beschreibung

- Sei $\mathcal{A}_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_0^1, Z_0^1, F_1)$ PDA, der L durch Übergang in einen Zustand aus F_1 akzeptiert.
- Wir konstruieren dazu einen PDA $\mathcal{A}_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_0^2, Z_0^2)$, der L durch leeren STACK akzeptiert.
- Sei q_E ein neuer Zustand
- Sei Z_0^2 ein neues Stack-Symbol

Idee der Konstruktion von $\mathcal{A}_2$.

- Lege zu Beginn Z_0^2 vor Z_0^1 auf den Stack, so dass der Stack nicht „versehentlich" geleert werden kann.
- Dann Verfahre wie in $\mathcal{A}_1$.
- Wenn Zustand in F_1 erreicht wird: Gehe zu q_E und leere den Stack

Beweis - Konstruktion

- $\mathcal{A}_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_0^1, Z_0^1, F_1)$ akzeptiert durch Endzustand
- $\mathcal{A}_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_0^2, Z_0^2)$, akzeptiert durch leeren Stack
- Sei q_E ein neuer Zustand
- Sei Z_0^2 ein neues Stack-Symbol

$$
\begin{aligned}
Q_2 &= Q_1 \cup \{q_0^2, q_E\} \\
\Gamma_2 &= \Gamma_1 \cup \{Z_0^2\} \\
\delta_2(q_0^2, \varepsilon, Z_0^2) &= \{(q_0^1, Z_0^1 Z_0^2)\} \\
\delta_2(q, a, Z) &= \delta_1(q, a, Z) \text{ für } q \in Q_1,\ a \neq \varepsilon,\ Z \in \Gamma_1 \\
&\qquad q \in Q_1 \setminus F_1,\ a = \varepsilon,\ Z \in \Gamma_1 \\
\delta_2(q, \varepsilon, Z) &= \delta_1(q, \varepsilon, Z) \cup \{(q_E, \varepsilon)\} \text{ für } q \in F_1,\ Z \in \Gamma_2 \\
\delta_2(q_E, \varepsilon, Z) &= \{(q_E, \varepsilon)\} \text{ für } Z \in \Gamma_2 \\
\delta(\cdot) &= \emptyset \text{ sonst}
\end{aligned}
$$

Satz:
Zu einem PDA, der eine Sprache L mit leerem STACK akzeptiert, kann ein PDA konstruiert werden, der L durch einen akzeptierenden Endzustand akzeptiert.

Beweis - Beschreibung

- Sei $\mathcal{A}_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_0^1, Z_0^1)$, ein PDA der $w \in L$ mit leerem STACK akzeptiert
- Wir konstruieren dazu einen PDA $\mathcal{A}_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_0^2, Z_0^2, F_2)$, der genau die $w \in L$ durch Übergang in einen Zustand $q \in F_2$ akzeptiert.
- Sei Z_0^2 ein neues Stack-Symbol
- Sei q_F ein neuer (End-)Zustand
- Sei q_0^2 ein neuer (Anfangs-)Zustand

Idee der Konstruktion von $\mathcal{A}_2$.

- Lege zu Beginn Z_0^2 vor Z_0^1 auf den Stack, und lösche Z_0^2 nur, wenn die Abarbeitung von $\mathcal{A}_1$ durch leeren Stack akzeptiert hätte.
- Gehe in Endzustand q_F, wenn $\mathcal{A}_1$ durch leeren Stack akzeptiert hätte.

Beweis - Konstruktion

- $\mathcal{A}_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_0^1, Z_0^1, F_1)$ akzeptiert durch leeren Stack
- $\mathcal{A}_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_0^2, Z_0^2)$, akzeptiert durch Endzustand
- q_0^2 neuer Anfangszustand
- q_F neuer (End-)Zustand
- Z_0^2 ein neues Stack-Symbol

$$\begin{aligned} Q_2 &= Q_1 \cup \{q_0^2, q_F\}, \\ F_2 &= \{q_F\} \\ \Gamma_2 &= \Gamma_1 \cup \{Z_0^2\} \end{aligned}$$

$$\begin{aligned} \delta_2(q_0^2, a, X) &= \begin{cases} \{q_0^1, Z_0^1 Z_0^2\} & \text{falls } a = \varepsilon \text{ und } X = Z_0^2 \\ \emptyset & \text{sonst} \end{cases} \\ \delta_2(q, a, Z) &= \delta_1(q, a, Z), \text{ falls } q \in Q_1, a \in \Sigma \cup \{\varepsilon\} \text{ und } Z \in \Gamma_1 \\ \delta_2(q, \varepsilon, Z_0^2) &= \{(q_F, \varepsilon)\} \text{ für } q \in Q_1. \end{aligned}$$

Satz:
Für eine Grammatik G in Greibach-Normalform kann ein PDA konstruiert werden, der $L(G)$ mit leerem STACK akzeptiert.

Beweis

- Sei $G = (\Sigma, V, S, R)$ eine Grammatik in Greibach Normalform
- Konstruiere gewünschten Automaten $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$

$$\begin{array}{rcl} Q & := & \{q_0\} \\ \Gamma & := & V \\ Z_0 & := & S \\ \delta(q_0, a, A) & := & \{(q_0, \alpha) | (A \to a\alpha) \in R\} \end{array}$$

Per Induktion über die Länge i einer Ableitung beweisen wir:

- $S \xrightarrow{*} w_1 \dots w_i A_1 \dots A_m \quad \Leftrightarrow \quad \mathcal{A}$ kann beim Lesen von $w_1 \dots w_i$ den STACK-Inhalt $A_1 \dots A_m$ erzeugen. Möglicherweise ist $A_1 \dots A_m = \epsilon$.

Daraus folgt:

- $\mathcal{A}$ erkennt $w_1 \dots w_n$ mit leerem STACK $\Leftrightarrow S \xrightarrow{*} w_1 \dots w_n$ in G

Beweis

$$Q := \{q_0\} \quad \Gamma := V \quad Z_0 := S$$
$$\delta(q_0, a, A) := \{(q_0, \alpha) | (A \to a\alpha) \in R\}$$

Induktionsanfang ist mit $i = 0$ trivialerweise erfüllt.
Induktionsschritt:
Sei $i \geq 1$ und „$\xrightarrow{j}$" stehe für eine Ableitung der Länge j. Dann gilt

$$S \xrightarrow{i} w_1 \ldots w_i A_1 \ldots A_m \iff \begin{array}{l} \exists A' \in V, r \in \{1, \ldots, m\} \text{ mit} \\ S \xrightarrow{i-1} w_1 \ldots w_{i-1} A' A_r \ldots A_m \\ \to w_1 \ldots w_i A_1 \ldots A_m. \end{array}$$

Mit Induktionsvoraussetzung ist dies äquivalent zu

$\exists A' \in V, r \in \{1, \ldots, m\}$ so, dass

- $\mathcal{A}$ das Wort $w_1 \ldots w_{i-1}$ lesen und dabei STACK-Inhalt $A' A_r \ldots A_m$ erzeugen kann, und
- $A' \to w_i A_1 \ldots A_{r-1}$ Regel von G ist.

Beweis

$$Q := \{q_0\} \quad \Gamma := V \quad Z_0 := S$$
$$\delta(q_0, a, A) := \{(q_0, \alpha) | (A \to a\alpha) \in R\}$$

Induktionsanfang ist mit $i = 0$ trivialerweise erfüllt.
Induktionsschritt:
Sei $i \geq 1$ und „$\xrightarrow{j}$“ stehe für eine Ableitung der Länge j.
Mit Induktionsvoraussetzung ist dies äquivalent zu

$\exists A' \in V, r \in \{1, \ldots, m\}$ so, dass

- $\mathcal{A}$ das Wort $w_1 \ldots w_{i-1}$ lesen und dabei STACK-Inhalt $A' A_r \ldots A_m$ erzeugen kann, und
- $A' \to w_i A_1 \ldots A_{r-1}$ Regel von G ist.

Dies ist genau dann erfüllt, wenn $\mathcal{A}$ das Wort $w_1 \ldots w_i$ lesen und dabei den STACK-Inhalt $A_1 \ldots A_m$ erzeugen kann.

Satz:
Jede durch einen PDA (mit leerem STACK oder durch akzeptierende Endzustände) akzeptierte Sprache ist kontextfrei.