Theoretische Grundlagen der Informatik

Vorlesung am 24. Januar 2012

INSTITUT FÜR THEORETISCHE INFORMATIK

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

www.kit.edu

Greibach Normalform

Greibach Normalform
Eine kontextfreie Grammatik ist in **Greibach-Normalform**, wenn alle Ableitungsregeln von der Form

$$A \to a\alpha \text{ mit } A \in V, a \in \Sigma \text{ und } \alpha \in V^*$$

sind.

Satz:
Für jede kontextfreie Grammatik G, für die $L(G)$ das leere Wort nicht enthält, kann eine (äquivalente) kontextfreie Grammatik G' mit $L(G) = L(G')$ in Greibach-Normalform konstruiert werden.

Beweis - Ersetzung (i)

Folgende Ersetzungen ändern nichts an der erzeugten Sprache:

Ersetzung (i). Eine Regel

$$A \to \alpha_1 B \alpha_2$$

wobei

$$B \to \beta_1, \quad B \to \beta_2, \quad \dots, \quad B \to \beta_r$$

alle Regeln sind, deren linke Seite B ist, kann durch die Regeln

$$\begin{gathered} A \to \alpha_1 \beta_1 \alpha_2 \\ A \to \alpha_1 \beta_2 \alpha_2 \\ \dots \\ A \to \alpha_1 \beta_r \alpha_2 \end{gathered}$$

ersetzt werden.

Beweis - Ersetzung (ii)

Folgende Ersetzungen ändern nichts an der erzeugten Sprache:

Ersetzung (ii). Seien

$$\begin{aligned} A &\rightarrow A\alpha_1, \ldots, A \rightarrow A\alpha_r \\ A &\rightarrow \beta_1, \ldots, A \rightarrow \beta_s \end{aligned}$$

alle Regeln, deren linke Seite A ist, wobei β_i nicht mit A beginnen. Dann können die Regeln

$$A \rightarrow A\alpha_1, \ldots, A \rightarrow A\alpha_r$$

durch die Regeln

$$\begin{aligned} A &\rightarrow \beta_1 B, \ldots, A \rightarrow \beta_s B \\ B &\rightarrow \alpha_1, \ldots, B \rightarrow \alpha_r, \\ B &\rightarrow \alpha_1 B, \ldots, B \rightarrow \alpha_r B \end{aligned}$$

ersetzt werden. Dabei sei B eine neu eingeführte Variable.

Beweis - Definitionen

Annahme G ist in Chomsky-Normalform mit

$$V = \{A_1, \dots, A_m\}$$
$$\Sigma = \{a_1, \dots, a_n\}$$

und damit ausschließlich Regeln der Form

$$A_i \rightarrow A_j A_k$$
$$A_i \rightarrow a_j \ .$$

Die Grammatik in Greibach-Normalform wird zusätzlich die Variablen $\{B_1, \dots, B_m\}$ benutzen. Sei

$$V' := \{A_1, \dots, A_m, B_1, \dots, B_m\}$$

Beweis: Invarianten

$$V' := \{A_1, \ldots, A_m, B_1, \ldots, B_m\} \qquad \Sigma = \{a_1, \ldots, a_n\}$$

zu Beginn mit Regeln der Form

$$A_i \to A_j A_k$$
$$A_i \to a_j \ .$$

Während der Umformung sind folgende Invarianten erfüllt.

1.Invariante
Die rechte Seite einer Regel, deren rechte Seite mit einer Variablen beginnt, besteht nur aus Variablen.

Beweis: Invarianten

$$V' := \{A_1, \ldots, A_m, B_1, \ldots, B_m\} \qquad \Sigma = \{a_1, \ldots, a_n\}$$

zu Beginn mit Regeln der Form

$$A_i \to A_j A_k$$
$$A_i \to a_j \ .$$

Während der Umformung sind folgende Invarianten erfüllt.

2.Invariante
Die rechte Seite einer Regel, deren rechte Seite mit einer Variablen beginnt, beginnt mit einer Variablen aus $V = \{A_1, \ldots, A_m\}$.

Beweis: Invarianten

Karlsruher Institut für Technologie

$$V' := \{A_1, \ldots, A_m, B_1, \ldots, B_m\} \qquad \Sigma = \{a_1, \ldots, a_n\}$$

zu Beginn mit Regeln der Form

$$A_i \to A_j A_k$$
$$A_i \to a_j \ .$$

Während der Umformung sind folgende Invarianten erfüllt.

3.Invariante
Symbole aus Σ kommen nur als erstes Zeichen der rechten Seite einer Regel vor.

Beweis: Invarianten

$$V' := \{A_1, \ldots, A_m, B_1, \ldots, B_m\} \qquad \Sigma = \{a_1, \ldots, a_n\}$$

zu Beginn mit Regeln der Form

$$A_i \to A_j A_k$$
$$A_i \to a_j \ .$$

Während der Umformung sind folgende Invarianten erfüllt.

4.Invariante
Die rechte Seite einer Regel, deren linke Seite aus $V = \{A_1, \ldots, A_m\}$ ist und deren rechte Seite mit einer Variablen aus V beginnt, beginnt sogar mit zwei Variablen aus V.

Beweis: Invarianten

$$V' := \{A_1, \ldots, A_m, B_1, \ldots, B_m\} \qquad \Sigma = \{a_1, \ldots, a_n\}$$

zu Beginn mit Regeln der Form

$$A_i \rightarrow A_j A_k$$
$$A_i \rightarrow a_j \ .$$

Während der Umformung sind folgende Invarianten erfüllt.

5.Invariante
Die rechte Seite einer Regel, deren linke Seite aus $V' \setminus V = \{B_1, \ldots, B_m\}$ ist, besteht nur aus Variablen aus V'.

Beweis: Invarianten

$$V' := \{A_1, \ldots, A_m, B_1, \ldots, B_m\} \qquad \Sigma = \{a_1, \ldots, a_n\}$$

zu Beginn mit Regeln der Form

$$A_i \to A_j A_k$$
$$A_i \to a_j \; .$$

Wir formen G zunächst so um, dass außer Invarianten 1-5 noch die nächste Invariante gilt:

6.Invariante
Falls $A_i \to A_j\alpha$ Regel ist, so gilt $j > i$.

Beweis - Übersicht Invarianten

1.Invariante Die rechte Seite einer Regel, deren rechte Seite mit einer Variablen beginnt, besteht nur aus Variablen.

2.Invariante Die rechte Seite einer Regel, deren rechte Seite mit einer Variablen beginnt, beginnt mit einer Variablen aus $V = \{A_1, \ldots, A_m\}$.

3.Invariante Symbole aus Σ kommen nur als erstes Zeichen der rechten Seite einer Regel vor.

4.Invariante Die rechte Seite einer Regel, deren linke Seite aus $V = \{A_1, \ldots, A_m\}$ ist und deren rechte Seite mit einer Variablen aus V beginnt, beginnt sogar mit zwei Variablen aus V.

5.Invariante Die rechte Seite einer Regel, deren linke Seite aus $V' \backslash V = \{B_1, \ldots, B_m\}$ ist, besteht nur aus Variablen aus V'.

6.Invariante Falls $A_i \rightarrow A_j\alpha$ Regel ist, so gilt $j > i$.

Beweis: Beispielgrammatik

$$
\begin{aligned}
V &= \{A_1, A_2, A_3\} \\
\Sigma &= \{0, 1\} \\
S &= A_1 \\
R &= \{A_1 \to A_2A_3, \\
&\quad A_2 \to A_3A_1,\ A_2 \to 1, \\
&\quad A_3 \to A_1A_2,\ A_3 \to 0\}
\end{aligned}
$$

Beweis - Verfahren - Schritt 1

Vorher: Grammatik in Chomsky-Normalform
Nachher: 6. Invariante hält: Falls $A_i \rightarrow A_j\alpha$ Regel ist, so gilt $j > i$.

Aktion: Dabei wenden wir (in dieser Reihenfolge)

Ersetzung (ii)	zur Ersetzung aller Regeln $A_1 \rightarrow A_1\alpha$
Ersetzung (i)	zur Ersetzung aller Regeln $A_2 \rightarrow A_1\alpha$
Ersetzung (ii)	zur Ersetzung aller Regeln $A_2 \rightarrow A_2\alpha$
Ersetzung (i)	zur Ersetzung aller Regeln $A_3 \rightarrow A_1\alpha$
Ersetzung (i)	zur Ersetzung aller Regeln $A_3 \rightarrow A_2\alpha$
Ersetzung (ii)	zur Ersetzung aller Regeln $A_3 \rightarrow A_3\alpha$
	$\vdots$

Beweis - Schritt 1 - Beispiel

Ersetzung (i)

$$\begin{aligned} A &\to \alpha_1 B \alpha_2 \\ B &\to \beta_1 | \beta_2, \dots | \beta_r \end{aligned}$$

$$\begin{aligned} A &\to \alpha_1 \beta_1 \alpha_2 | \alpha_1 \beta_2 \alpha_2 | \dots | \alpha_1 \beta_r \alpha_2 \\ B &\to \beta_1 | \beta_2, \dots | \beta_r \end{aligned}$$

$$\begin{aligned} V &= \{A_1, A_2, A_3\} \\ \Sigma &= \{0, 1\} \\ S &= A_1 \\ R &= \{A_1 \to A_2 A_3, \\ &\quad A_2 \to A_3 A_1,\ A_2 \to 1, \\ &\quad \color{red}{A_3 \to A_1 A_2},\ A_3 \to 0\} \end{aligned}$$

$$\begin{aligned} V &= \{A_1, A_2, A_3\} \\ \Sigma &= \{0, 1\} \\ S &= A_1 \\ R &= \{A_1 \to A_2 A_3, \\ &\quad A_2 \to A_3 A_1,\ A_2 \to 1, \\ &\quad \color{red}{A_3 \to A_2 A_3 A_2},\ A_3 \to 0\} \end{aligned}$$

Beweis - Schritt 1 - Beispiel

Ersetzung (i)

$$\begin{aligned} A &\to \alpha_1 B \alpha_2 \\ B &\to \beta_1 | \beta_2, \ldots | \beta_r \end{aligned}$$

$$\begin{aligned} V &= \{A_1, A_2, A_3\} \\ \Sigma &= \{0, 1\} \\ S &= A_1 \\ R &= \{A_1 \to A_2A_3, \\ &\quad A_2 \to A_3A_1,\ A_2 \to 1, \\ &\quad A_3 \to A_2A_3A_2,\ A_3 \to 0\} \end{aligned}$$

$$\begin{aligned} A &\to \alpha_1\beta_1\alpha_2 | \alpha_1\beta_2\alpha_2 | \ldots | \alpha_1\beta_r\alpha_2 \\ B &\to \beta_1 | \beta_2, \ldots | \beta_r \end{aligned}$$

$$\begin{aligned} V &= \{A_1, A_2, A_3\} \\ \Sigma &= \{0, 1\} \\ S &= A_1 \\ R &= \{A_1 \to A_2A_3, \\ &\quad A_2 \to A_3A_1,\ A_2 \to 1, \\ &\quad A_3 \to A_3A_1A_3A_2 | 1A_3A_2, \\ &\quad A_3 \to 0\} \end{aligned}$$

Beweis - Schritt 1 - Beispiel

Ersetzung (ii)

$$\begin{array}{lcl} A & \to & A\alpha_1, \ldots, A \to A\alpha_r \\ A & \to & \beta_1, \ldots, A \to \beta_s \end{array}$$

$$\begin{array}{lcl} A & \to & \beta_1 B, \ldots, A \to \beta_s B \\ B & \to & \alpha_1, \ldots, B \to \alpha_r, \\ B & \to & \alpha_1 B, \ldots, B \to \alpha_r B \\ A & \to & \beta_1, \ldots, A \to \beta_s \end{array}$$

$$\begin{array}{lcl} V & = & \{A_1, A_2, A_3\} \\ \Sigma & = & \{0, 1\} \\ S & = & A_1 \\ R & = & \{A_1 \to A_2A_3, \\ & & \quad A_2 \to A_3A_1,\ A_2 \to 1, \\ & & \quad \color{red}{A_3 \to A_3A_1A_3A_2}, \\ & & \quad A_3 \to 0|1A_3A_2\} \end{array}$$

$$\begin{array}{lcl} V & = & \{A_1, A_2, A_3\} \\ \Sigma & = & \{0, 1\} \\ S & = & A_1 \\ R & = & \{A_1 \to A_2A_3, \\ & & \quad A_2 \to A_3A_1,\ A_2 \to 1, \\ & & \quad \color{red}{A_3 \to 0B_3|1A_3A_2B_3}, \\ & & \quad \color{red}{B_3 \to A_1A_3A_2|A_1A_3A_2B_3} \\ & & \quad A_3 \to 0|1A_3A_2\} \end{array}$$

Beweis - Verfahren - Schritt 2

Vorher: Alle Regeln sind von der Form

$$\begin{aligned} A &\to a\alpha \\ A &\to \alpha \text{ mit } \alpha \in (V')^*, a \in \Sigma \end{aligned}$$

wobei es keine ε-Regeln oder Kettenregeln mit linker Seite $A \in V$ gibt.
Wegen Invariante 6

- gibt es keine Regel $A_m \to \alpha$
- beginnen alle $A_{m-1} \to \alpha$-Regeln mit A_m.

Aktion: Ersetze mit absteigenden k alle Regeln der Form

$$A_k \to \alpha \text{ mit } \alpha \in (V')^*$$

mittels Ersetzung (i).

Nachher: Regeln mit linker Seite in V in Form $A_k \to a\alpha, a \in \Sigma, \alpha \in (V')^*$.

Beweis - Schritt 2 - Beispiel

Ersetzung (i)

$$\begin{aligned} A &\rightarrow \alpha_1 B \alpha_2 \\ B &\rightarrow \beta_1 | \beta_2, \dots | \beta_r \end{aligned}$$

$$\begin{aligned} A &\rightarrow \alpha_1 \beta_1 \alpha_2 | \alpha_1 \beta_2 \alpha_2 | \dots | \alpha_1 \beta_r \alpha_2 \\ B &\rightarrow \beta_1 | \beta_2, \dots | \beta_r \end{aligned}$$

$$\begin{aligned} V &= \{A_1, A_2, A_3\} \\ \Sigma &= \{0, 1\} \\ S &= A_1 \\ R &= \{A_1 \rightarrow A_2 A_3, \\ & \quad {\color{red} A_2 \rightarrow A_3 A_1,} \\ & \quad A_2 \rightarrow 1, \\ & \quad A_3 \rightarrow 0 B_3 | 1 A_3 A_2 B_3, \\ & \quad A_3 \rightarrow 0 | 1 A_3 A_2\} \\ & \quad B_3 \rightarrow A_1 A_3 A_2 | A_1 A_3 A_2 B_3 \end{aligned}$$

$$\begin{aligned} V &= \{A_1, A_2, A_3\} \\ \Sigma &= \{0, 1\} \\ S &= A_1 \\ R &= \{A_1 \rightarrow A_2 A_3, \\ & \quad {\color{red} A_2 \rightarrow 0 B_3 A_1 | 1 A_3 A_2 B_3 A_1} \\ & \quad {\color{red} A_2 \rightarrow 0 A_1 | 1 A_3 A_2 A_1} \\ & \quad A_2 \rightarrow 1, \\ & \quad A_3 \rightarrow 0 B_3 | 1 A_3 A_2 B_3, \\ & \quad A_3 \rightarrow 0 | 1 A_3 A_2\} \\ & \quad B_3 \rightarrow A_1 A_3 A_2 | A_1 A_3 A_2 B_3 \end{aligned}$$

Beweis - Schritt 2 - Beispiel

Ersetzung (i)

$$
\begin{aligned}
V &= \{A_1, A_2, A_3\} \\
\Sigma &= \{0, 1\} \\
S &= A_1 \\
R &= \{\color{red}{A_1 \to A_2A_3}, \\
&\quad A_2 \to 0B_3A_1 | 1A_3A_2B_3A_1 \\
&\quad A_2 \to 0A_1 | 1A_3A_2A_1 | 1 \\
&\quad A_3 \to 0B_3 | 1A_3A_2B_3, \\
&\quad A_3 \to 0 | 1A_3A_2\} \\
&\quad B_3 \to A_1A_3A_2 | A_1A_3A_2B_3
\end{aligned}
$$

$$
\begin{aligned}
V &= \{A_1, A_2, A_3\} \\
\Sigma &= \{0, 1\} \\
S &= A_1 \\
R &= \{\color{red}{A_1 \to 0B_3A_1A_3} \\
&\quad \color{red}{A_1 \to 1A_3A_2B_3A_1A_3,} \\
&\quad \color{red}{A_1 \to 0A_1A_3 | 1A_3A_2A_1A_3 | 1A_3,} \\
&\quad A_2 \to 0B_3A_1 | 1A_3A_2B_3A_1 \\
&\quad A_2 \to 0A_1 | 1A_3A_2A_1 | 1 \\
&\quad A_3 \to 0B_3 | 1A_3A_2B_3, \\
&\quad A_3 \to 0 | 1A_3A_2\} \\
&\quad B_3 \to A_1A_3A_2 | A_1A_3A_2B_3
\end{aligned}
$$

Beweis - Verfahren - Schritt 3

Vorher: Rechte Seiten von Regeln, deren linke Seite aus $V' \backslash V = \{B_1, \dots, B_m\}$ ist, beginnen mit einer Variablen aus $\{A_1, \dots, A_m\}$ (wegen Invarianten 2 und 5).

Aktion: Ersetze Regeln $B_i \to A_j\alpha$, $\quad \alpha \in (\Sigma \cup V')^*$ mit Ersetzung (i).

Nachher: G ist in Greibach-Normalform.

Beweis - Schritt 3 - Beispiel

Ersetzung (i)

$$A_1 \rightarrow 0B_3A_1A_3$$
$$A_1 \rightarrow 1A_3A_2B_3A_1A_3,$$
$$A_1 \rightarrow 0A_1A_3|1A_3A_2A_1A_3|1A_3,$$
$$A_2 \rightarrow 0B_3A_1|1A_3A_2B_3A_1$$
$$A_2 \rightarrow 0A_1|1A_3A_2A_1|1$$
$$A_3 \rightarrow 0B_3|1A_3A_2B_3,$$
$$A_3 \rightarrow 0|1A_3A_2\}$$
$$B_3 \rightarrow A_1A_3A_2|A_1A_3A_2B_3$$

$$A_1 \rightarrow 0B_3A_1A_3$$
$$A_1 \rightarrow 1A_3A_2B_3A_1A_3,$$
$$A_1 \rightarrow 0A_1A_3|1A_3A_2A_1A_3|1A_3,$$
$$A_2 \rightarrow 0B_3A_1|1A_3A_2B_3A_1$$
$$A_2 \rightarrow 0A_1|1A_3A_2A_1|1$$
$$A_3 \rightarrow 0B_3|1A_3A_2B_3,$$
$$A_3 \rightarrow 0|1A_3A_2\}$$
$$B_3 \rightarrow 1A_3A_3A_2|0B_3A_1A_3A_3A_2$$
$$B_3 \rightarrow 1A_3A_2A_1A_3A_3A_2|0A_1A_3A_3A_2$$
$$B_3 \rightarrow 1A_3A_2B_3A_1A_3A_3A_2|1A_3A_3A_2B_3$$
$$B_3 \rightarrow 0B_3A_1A_3A_3A_2B_3|1A_3A_2A_1A_3A_3A_2B_3$$
$$B_3 \rightarrow 0A_1A_3A_3A_2B_3$$
$$B_3 \rightarrow 1A_3A_2B_3A_1A_3A_3A_2B_3$$

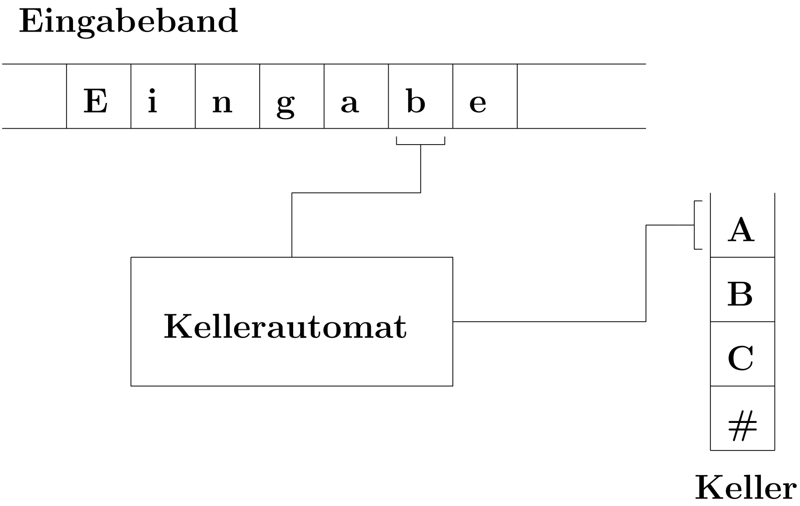
Kellerautomaten

Karlsruher Institut für Technologie

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

Karlsruher Institut für Technologie

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

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

Zustand	Eingabe	Stack

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

Beispiel - Berechnung für Eingabe 001#100

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

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

Beispiel - Berechnung für Eingabe 001#100

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

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

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

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

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	$00Z_0$

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	$0Z_0$

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	$00Z_0$
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 $L = \{ww^R | w \in \{0,1\}^*\}$.

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

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

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

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

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 $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$

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

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

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

Mit Induktionsvoraussetzung ist dies äquivalent zu

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

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