# Theorie der Informatik

## 11. Kontextfreie Sprachen II

Malte Helmert Gabriele Röger

Universität Basel

1. April 2014



# Theorie der Informatik

1. April 2014 — 11. Kontextfreie Sprachen II





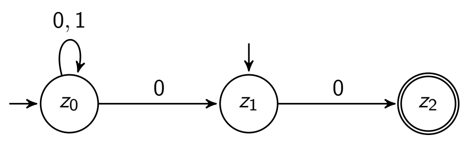


# 11.1 Kellerautomaten





## Limitierung von endlichen Automaten

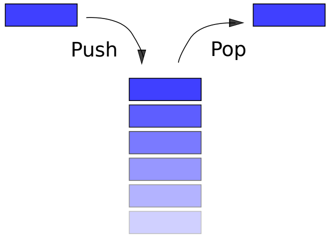



- Sprache $L$ ist regulär
  $\Leftrightarrow$ es gibt einen endlichen Automaten, der $L$ akzeptiert
- Welche Information kann ein endlicher Automat über das bereits gelesene Teilwort „speichern“?
- Für $L = \{a_1 a_2 \dots a_n a_n \dots a_2 a_1 \mid n > 0, a_i \in \{0, 1\}\}$ wäre unendlicher Speicher notwendig.
- Daher: Erweiterung von Automatenmodell um Speicher

## Keller

Ein Keller (oder Stapel, engl. Stack) ist eine Datenstruktur nach dem Last-In-First-Out-Prinzip (LIFO) mit folgenden Operationen:

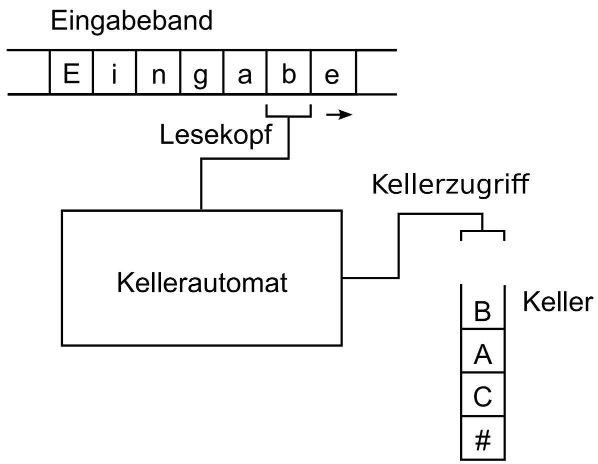
- push: Legt ein Objekt oben auf den Stapel
- pop: Nimmt das oberste Objekt vom Stapel
- peek: Liefert das oberste Objekt zurück ohne es zu entfernen








## Kellerautomat: anschaulich








## Kellerautomat: Definition

### Definition (Kellerautomat = PDA)

Ein Kellerautomat (push-down automaton, PDA) ist ein 6-Tupel $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ mit

- $Z$ endliche Menge der Zustände,
- $\Sigma$ das Eingabealphabet,
- $\Gamma$ das Kelleralphabet,
- $\delta : Z \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_e(Z \times \Gamma^*)$ die Überführungsfunktion (mit $\mathcal{P}_e$ Menge aller endlichen Teilmengen)
- $z_0 \in Z$ der Startzustand
- $\# \in \Gamma$ das unterste Kellerzeichen



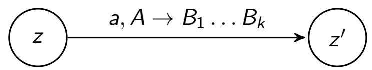


## Kellerautomat: Übergangsfunktion

Sei $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ Kellerautomat.

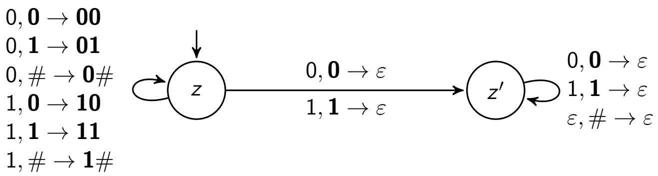
Was bedeutet Übergangsfunktion $\delta$ intuitiv?

- $(z', B_1 \dots B_k) \in \delta(z, a, A)$: Wenn $M$ im Zustand $z$ das Zeichen $a$ liest und $A$ das oberste Kellerzeichen ist, dann kann $M$ im nächsten Schritt in $z'$ übergehen und $A$ durch $B_1 \dots B_k$ ersetzen (danach $B_1$ oberstes Kellerzeichen)




- Spezialfall $a = \varepsilon$ zugelassen (spontaner Übergang)

# Kellerautomat: Beispiel




$M = (\{z, z'\}, \{0, 1\}, \{\mathbf{0}, \mathbf{1}, \#\}, \delta, z, \#)$ mit

| | | |
|---|---|---|
| $\delta(z, 0, \mathbf{0}) = \{(z, \mathbf{00}), (z', \varepsilon)\}$ | $\delta(z, 1, \mathbf{0}) = \{(z, \mathbf{10})\}$ | $\delta(z, \varepsilon, \mathbf{0}) = \emptyset$ |
| $\delta(z, 0, \mathbf{1}) = \{(z, \mathbf{01})\}$ | $\delta(z, 1, \mathbf{1}) = \{(z, \mathbf{11}), (z', \varepsilon)\}$ | $\delta(z, \varepsilon, \mathbf{1}) = \emptyset$ |
| $\delta(z, 0, \#) = \{(z, \mathbf{0}\#)\}$ | $\delta(z, 1, \#) = \{(z, \mathbf{1}\#)\}$ | $\delta(z, \varepsilon, \#) = \emptyset$ |
| $\delta(z', 0, \mathbf{0}) = \{(z', \varepsilon)\}$ | $\delta(z', 1, \mathbf{0}) = \emptyset$ | $\delta(z', \varepsilon, \mathbf{0}) = \emptyset$ |
| $\delta(z', 0, \mathbf{1}) = \emptyset$ | $\delta(z', 1, \mathbf{1}) = \{(z', \varepsilon)\}$ | $\delta(z', \varepsilon, \mathbf{1}) = \emptyset$ |
| $\delta(z', 0, \#) = \{(z', \varepsilon)\}$ | $\delta(z', 1, \#) = \emptyset$ | $\delta(z', \varepsilon, \#) = \{(z', \varepsilon)\}$ |



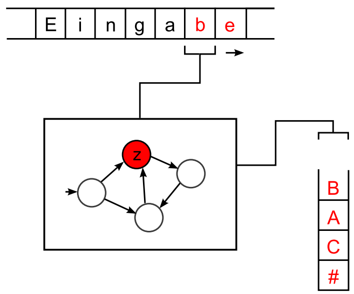


# Kellerautomat: Konfiguration

### Definition (Konfiguration eines Kellerautomaten)

Eine Konfiguration eines Kellerautomaten $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ ist gegeben durch ein Tripel $k \in Z \times \Sigma^* \times \Gamma^*$.

### Beispiel




Konfiguration $(z, be, BAC\#)$.





# Kellerautomat: Übergang

### Definition (Übergang eines Kellerautomaten)

Wir schreiben $k \vdash_M k'$, falls ein Kellerautomat $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ von Konfiguration $k$ in Konfiguration $k'$ übergehen kann. Es sind genau folgende Übergänge möglich:

$$(z, a_1 \dots a_n, A_1 \dots A_m) \vdash_M \begin{cases} (z', a_2 \dots a_n, B_1 \dots B_k A_2 \dots A_m) \\ \qquad \text{falls } (z', B_1 \dots B_k) \in \delta(z, a_1, A_1) \\ (z', a_1 a_2 \dots a_n, B_1 \dots B_k A_2 \dots A_m) \\ \qquad \text{falls } (z', B_1 \dots B_k) \in \delta(z, \varepsilon, A_1) \end{cases}$$

Falls $M$ aus dem Kontext klar ist, schreiben wir nur $k \vdash k'$.





# Kellerautomat: Erreichbarkeit von Konfigurationen

### Definition (Erreichbare Konfiguration)

Konfiguration $k'$ ist in PDA $M$ von Konfiguration $k$ aus erreichbar ($k \vdash_M^* k'$), falls $k = k'$ oder es gibt Konfigurationen $k_0, \dots, k_n$ ($n \geq 1$), so dass

- $k_0 = k$,
- $k_i \vdash_M k_{i+1}$ für $i \in \{0, \dots, n-1\}$, und
- $k_n = k'$.

# Kellerautomat: Erkanntes Wort

### Definition (erkanntes Wort bei Kellerautomaten)

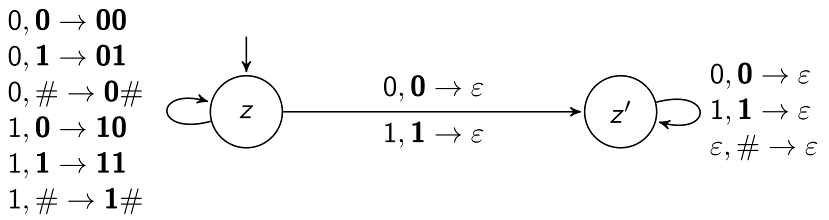
PDA $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ erkennt das Wort $w = a_0 \dots a_n$ genau dann, wenn $M$ von der Startkonfiguration $(z_0, w, \#)$ durch endliches Anwenden von $\delta$ in eine Konfiguration $(z, \varepsilon, \varepsilon)$ übergehen kann (Wort verarbeitet und Keller leer):

$$M \text{ erkennt } w \text{ gdw. } (z_0, w, \#) \vdash_M^* (z, \varepsilon, \varepsilon) \text{ für ein } z \in Z.$$





# Kellerautomat: Beispiel für erkanntes Wort




Der PDA erkennt zum Beispiel das Wort 11011011.
(Begründung an Tafel)





# Kellerautomat: Akzeptierte Sprache

### Definition (akzeptierte Sprache eines PDAs)

Sei $M$ ein Kellerautomat mit Eingabealphabet $\Sigma$. Die von $M$ akzeptierte Sprache ist definiert durch

$$\mathcal{L}(M) = \{w \in \Sigma^* \mid w \text{ wird von } M \text{ erkannt}\}.$$

Beispiel: Tafel





# PDAs akzeptieren genau kontextfreie Sprachen

### Satz

*Eine Sprache $L$ ist genau dann kontextfrei, wenn $L$ von einem Kellerautomaten akzeptiert wird.*

## PDAs akzeptieren genau kontextfreie Sprachen

Beweis.

$\Rightarrow$: Sei $G = (\Sigma, V, P, S)$ eine kontextfreie Grammatik für $L$.
Der Kellerautomat $M = (\{z\}, \Sigma, V \cup \Sigma, \delta, z, S)$ mit folgendem $\delta$ akzeptiert $L$.

- Für jede Regel $A \to w \in P$ mit $w \in (V \cup \Sigma)^*$ ist $(z, w) \in \delta(z, \varepsilon, A)$.
- Für $a \in \Sigma$ ist $(z, \varepsilon) \in \delta(z, a, a)$.

Denn:
$x \in \mathcal{L}(G)$
gdw. es gibt eine Ableitung in $G$ der Form $S \Rightarrow \cdots \Rightarrow x$
gdw. es gibt eine Folge von Konfigurationen von $M$ mit $(z, x, S) \vdash \cdots \vdash (z, \varepsilon, \varepsilon)$
gdw. $x \in \mathcal{L}(M)$ ...





## PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

$\Leftarrow$: Sei $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ PDA mit $\mathcal{L}(M) = L$.
Wir nehmen o.B.d.A. an, dass für jede $\delta$-Überführung $(z', B_1 \dots B_k) \in \delta(z, a, A)$ gilt, dass $k \leq 2$.

Sonst führen wir für jede Regel $(z', B_1 \dots B_k) \in \delta(z, a, A)$ mit $k > 2$ neue Zustände $z_1, \dots, z_{k-2}$ ein und ersetzen die Regel durch

$$\begin{aligned}
\delta(z, a, A) &\ni (z_1, B_{k-1}B_k)\\
\delta(z_1, \varepsilon, B_{k-1}) &= \{(z_2, B_{k-2}B_{k-1})\}\\
&\vdots\\
\delta(z_{k-3}, \varepsilon, B_3) &= \{(z_{k-2}, B_2B_3)\}\\
\delta(z_{k-2}, \varepsilon, B_2) &= \{(z', B_1B_2)\}
\end{aligned}$$

...





## PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

Konstruiere Grammatik $G = (V, \Sigma, P, S)$, die Rechenschritte von $M$ durch Linksableitungsschritte simuliert:

$$\begin{aligned}
V = {} & \{S\} \cup Z \times \Gamma \times Z\\
P = {} & \{S \to (z_0, \#, z) \mid z \in Z\} \cup\\
& \{(z, A, z') \to a \mid (z', \varepsilon) \in \delta(z, a, A)\} \cup\\
& \{(z, A, z') \to a(z_1, B, z') \mid (z_1, B) \in \delta(z, a, A), z' \in Z\} \cup\\
& \{(z, A, z') \to a(z_1, B, z_2)(z_2, C, z') \mid (z_1, BC) \in \delta(z, a, A),\\
& \qquad z', z_2 \in Z\}
\end{aligned}$$

...





## PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

Wir werden zunächst allgemein für $x \in \Sigma^*$ zeigen, dass

$$(z, A, z') \Rightarrow_G^* x \text{ genau dann wenn } (z, x, A) \vdash_M^* (z', \varepsilon, \varepsilon)$$

Für einen einzelnen Ableitungsschritt und $x = a \in \Sigma \cup \{\varepsilon\}$ gilt:

$$\begin{aligned}
(z, A, z') \Rightarrow_G a \;&\text{gdw. } (z, A, z') \to a \in P\\
&\text{gdw. } (z', \varepsilon) \in \delta(z, a, A)\\
&\text{gdw. } (z, a, A) \vdash_M (z', \varepsilon, \varepsilon)
\end{aligned}$$

...

# PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

Wir zeigen per Induktion über die Anzahl $n$ der Übergänge von $M$, dass allgemein $(z, A, z') \Rightarrow_G^* x$ aus $(z, x, A) \vdash_M^* (z', \varepsilon, \varepsilon)$ folgt.

Für $n = 1$ (einzelner Übergang) ist das bereits gezeigt.

Falls $n > 1$, hat $x$ Form $x = ay$ mit $a \in \Sigma \cup \{\varepsilon\}$.
Es gibt daher Zustand $z_1$ und Kellerinhalt $\alpha$, so dass $(z, ay, A) \vdash_M (z_1, y, \alpha) \vdash_M^+ (z', \varepsilon, \varepsilon)$. Unterscheide drei Fälle für $\alpha$:

- Fall $\alpha = \varepsilon$ nicht möglich, da $(z_1, y, \varepsilon)$ keine Folgekonfiguration besitzt.
- Fall $\alpha = B$: Dann gilt nach IV $(z_1, B, z') \Rightarrow_G^* y$. Wegen Regel $(z, A, z') \to a(z_1, B, z')$ gibt es die Gesamtableitung $(z, A, z') \Rightarrow a(z_1, B, z') \Rightarrow_G^* ay = x$.

...





# PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

- Fall $\alpha = BC$: $(z_1, y, BC) \vdash_M^* (z', \varepsilon, \varepsilon)$ kann zerlegt werden in $(z_1, y, BC \vdash_M^* (z_2, y_2, C)$ und $(z_2, y_2, C) \vdash_M^* (z', \varepsilon, \varepsilon)$, so dass $y_2$ Suffix von $y$ ist, d.h. $y = y_1 y_2$. Für $y_1$ gilt zudem, dass $(z_1, y_1, B) \vdash_M^* (z_2, \varepsilon, \varepsilon)$.

  Nach IV gilt daher $(z_1, B, z_2) \Rightarrow_G^* y_1$ und $(z_2, C, z') \Rightarrow_G^* y_2$.

  Wegen des Übergangs $(z, ay, A) \vdash_M (z_1, y, BC)$ muss es in $P$ eine Regel der Form $(z, A, z') \to a(z_1, B, z_2)(z_2, C, z')$ geben.

  Wir erhalten zusammen die Ableitung $(z, A, z') \Rightarrow_G a(z_1, B, z_2)(z_2, C, z') \Rightarrow_G^* ay_1y_2 = x$.

...





# PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

Bleibt zu zeigen, dass aus der Ableitbarkeit $(z, A, z') \Rightarrow_G^* x$ die Übergangsmöglichkeit $(z, x, A) \vdash_M^* (z', \varepsilon, \varepsilon)$ folgt. Wir zeigen dies per Induktion über die Länge $k$ der Linksableitung von $x$.

Für $k = 1$ (ein Ableitungsschritt) ist dies bereits erledigt.

Für $k > 1$ unterscheide drei Fälle:

- Fall $(z, A, z') \Rightarrow_G a \Rightarrow_G^* x$: Dann ist $x = a$, was bei $k > 1$ nicht möglich ist.
- Fall $(z, A, z') \Rightarrow_G a(z_1, B, z') \Rightarrow_G^* ay = x$: Dann ist $(z_1, B) \in \delta(z, a, A)$ und nach IV gilt $(z_1, y, B) \vdash_M^* (z', \varepsilon, \varepsilon)$. Insgesamt folgt $(z, ay, A) \vdash_M (z_1, y, B) \vdash_M^* (z', \varepsilon, \varepsilon)$.

...





# PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

- Fall $(z, A, z') \Rightarrow_G a(z_1, B, z_2)(z_2, C, z') \Rightarrow_G^* ay = x$: Dann ist $(z_1, BC) \in \delta(z, a, A)$ und nach IV gilt $(z_1, y_1, B) \vdash_M^* (z_2, \varepsilon, \varepsilon)$ und $(z_2, y_2, C) \vdash_M^* (z', \varepsilon, \varepsilon)$, wobei $y = y_1y_2$. Insgesamt folgt $(z, ay_1y_2, A) \vdash_M (z_1, y_1y_2, BC) \vdash_M^* (z_2, y_2, C) \vdash_M^* (z', \varepsilon, \varepsilon)$.

...

# PDAs akzeptieren genau kontextfreie Sprachen

Beweis (Fortsetzung).

Insgesamt haben wir für einen gegebenen PDA $M$ eine kontextfreie Grammatik $G$ angegeben, so dass für alle Wörter $x$ gilt

$$(z, A, z') \Rightarrow_G^* x \text{ genau dann wenn } (z, x, A) \vdash_M^* (z', \varepsilon, \varepsilon)$$

Damit gilt

$$\begin{aligned} x \in \mathcal{L}(M) &\text{ gdw. } (z_0, x, \#) \vdash_M^* (z, \varepsilon, \varepsilon) \text{ für ein } z \in Z \\ &\text{ gdw. } S \Rightarrow_G (z_0, \#, z) \Rightarrow_G^* x \text{ für ein } z \in Z \\ &\text{ gdw. } x \in \mathcal{L}(G). \end{aligned}$$

Die Grammatik erzeugt also die vom PDA akzeptierte Sprache. □





# 11.2 Zusammenfassung





# Zusammenfassung

- Kellerautomaten (PDAs) erweitern NFAs um Speicher.
- PDAs akzeptieren nicht mit Endzuständen, sondern mit leerem Keller.
- Die von PDAs akzeptierten Sprachen sind genau die kontextfreien Sprachen.





# Weitere Themen zu kontextfreien Sprachen und PDAs

- Mit dem CYK-Algorithmus (nach Cocke, Younger und Kasami) kann man für eine Grammatik $G$ in CNF und ein Wort $w$ in Zeit $O(|w|^3)$ entscheiden, ob $w \in \mathcal{L}(G)$.
- In der Greibach-Normalform für kontextfreie Sprachen haben alle Regeln die Form $A \rightarrow aB_1B_2 \dots B_k$ ($k \geq 0$) oder $S \rightarrow \varepsilon$ mit Startsymbol $S$.
- Deterministische Kellerautomaten haben die Einschränkung, dass für $z \in Z, a \in \Sigma, A \in \Gamma$ gilt $|\delta(z, a, A)| + |\delta(z, \varepsilon, A)| \leq 1$. Zudem akzeptieren sie nicht mit leerem Keller, sondern mit Endzuständen.
- Die Klasse der von deterministischen PDAs akzeptierten Sprachen heisst deterministisch kontextfreie Sprachen. Sie ist echte Obermenge der regulären Sprachen und echte Teilmenge der kontextfreien Sprachen.