Theoretische Informatik 1

WiSe 23/24

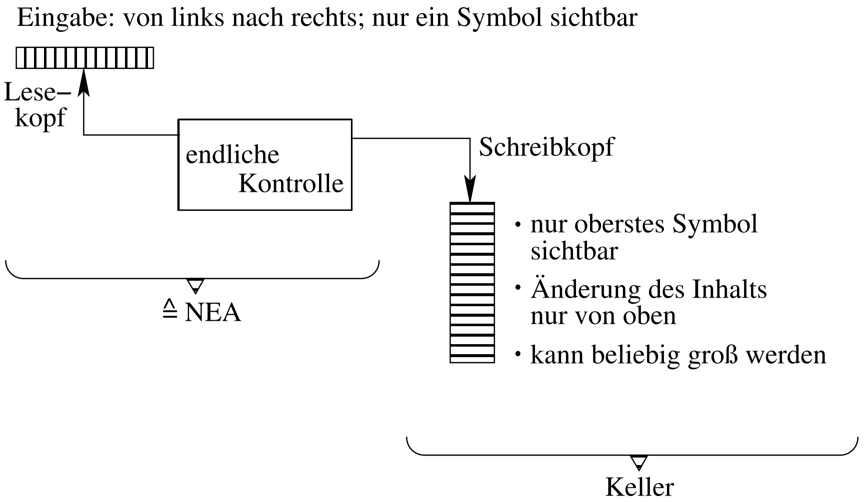
Prof. Dr. Sebastian Siebertz
AG Theoretische Informatik
MZH, Raum 3160
siebertz@uni-bremen.de

Universität
Bremen

Kellerautomaten

- Charakterisierung kontextfreier Sprachen: bisher mittels Grammatiken
- Frage: *Gibt es ein passendes Automatenmodell?*
- Endliche Automaten sind ungeeignet:
  - sie können nicht alle kontextfreien Sprachen erkennen,
  - denn sie können nicht „unbeschränkt zählen“
    (notwendig um z.B. $\{a^n b^n : n \geq 0\}$ zu akzeptieren)
- Daher: *Erweiterung um eine potentiell unendliche Speicherkomponente* – einen *Keller* (engl.: Stack)

Intuition


Eingabe: von links nach rechts; nur ein Symbol sichtbar
Lese–
kopf
endliche
Kontrolle
Schreibkopf
≙ NEA
• nur oberstes Symbol sichtbar
• Änderung des Inhalts nur von oben
• kann beliebig groß werden
Keller

Definition Kellerautomat

Definition Kellerautomat

Ein *Kellerautomat (pushdown automaton, kurz PDA)* ist ein Tupel $\mathcal{A} = (Q, \Sigma, \Gamma, q_s, Z_s, \Delta, F)$, wobei

- Q eine endliche Menge von *Zuständen* ist,
- Σ das *Eingabealphabet* ist,
- Γ das *Kelleralphabet* ist,
- $q_s \in Q$ der *Anfangszustand* ist,
- $Z_s \in \Gamma$ das *Kellerstartsymbol* ist,
- $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q$ eine endliche *Übergangsrelation* ist und
- $F \subseteq Q$ eine Menge von *akzeptierenden Zuständen* ist.

Die Übergangsrelation

- Die Übergangsrelation $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q$.
- (q, a, Z, γ, q') bedeutet:
  - Im Zustand q
  - mit aktuellem Eingabesymbol a
  - und oberstem Kellersymbol Z
  - darf der Automat Z auf dem Keller durch γ ersetzen,
  - in den Zustand q' wechseln
  - und zum nächsten Eingabesymbol übergehen.
- $(q, \varepsilon, Z, \gamma, q')$: wie oben, aber
  - das aktuelle Eingabesymbol ist nicht relevant
  - und wir gehen nicht zum nächsten Eingabesymbol (der Lesekopf ändert seine Position nicht).

Akzeptanz eines Kellerautomaten

- Akzeptanzverhalten eines Kellerautomaten: über Konfigurationen
- Eine Konfiguration beschreibt
  - den *noch zu lesenden Rest* $u \in \Sigma^*$ der Eingabe $w \in \Sigma^*$ (Lesekopf steht auf dem ersten Symbol von u),
  - den *Zustand* $q \in Q$ und
  - den *Kellerinhalt* $\beta \in \Gamma^*$
    (Schreibkopf steht auf dem ersten Symbol von β).

Konfiguration

Definition

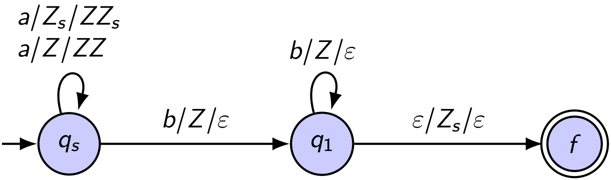
Sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_s, Z_s, \Delta, F)$ ein Kellerautomat.

- Eine *Konfiguration* von $\mathcal{A}$ ist ein Tupel
$$\mathcal{K} = (q, u, \beta) \in Q \times \Sigma^* \times \Gamma^*.$$
- Die Übergangsrelation ermöglicht die folgenden *Konfigurationsübergänge*:
  - ▷ $(q, au, Z\beta) \vdash_{\mathcal{A}} (q', u, \gamma\beta)$ falls $(q, a, Z, \gamma, q') \in \Delta$
  - ▷ $(q, u, Z\beta) \vdash_{\mathcal{A}} (q', u, \gamma\beta)$ falls $(q, \varepsilon, Z, \gamma, q') \in \Delta$
  - ▷ $\mathcal{K} \vdash_{\mathcal{A}}^* \mathcal{K}' \Leftrightarrow$ Konfigurationen $\mathcal{K}_0, \ldots, \mathcal{K}_n$ existieren mit
$$\mathcal{K}_0 = \mathcal{K},\ \mathcal{K}_n = \mathcal{K}' \text{ und } \mathcal{K}_i \vdash_{\mathcal{A}} \mathcal{K}_{i+1} \text{ für } 0 \le i < n.$$
- Der Automat $\mathcal{A}$ *akzeptiert* das Wort $w \in \Sigma^* \Leftrightarrow$
$$(q_s, w, Z_s) \vdash_{\mathcal{A}}^* (q_f, \varepsilon, \beta) \quad \text{für ein } q_f \in F \text{ und } \beta \in \Gamma^*.$$

Beispiel

- Ein PDA für $\{a^n b^n \mid n \geq 1\}$.

Erinnerung: **Übergangsrelation** $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q$.



- $(q_s, aabb, Z_s) \vdash_{\mathcal{A}} (q_s, abb, ZZ_s) \vdash_{\mathcal{A}} (q_s, bb, ZZZ_s) \vdash_{\mathcal{A}} (q_1, b, ZZ_s) \vdash_{\mathcal{A}} (q_1, \varepsilon, Z_s) \vdash_{\mathcal{A}} (f, \varepsilon, \varepsilon)$ – *akzeptiert*.
- $(q_s, aab, Z_s) \vdash^*_{\mathcal{A}} (q_s, b, ZZZ_s) \vdash_{\mathcal{A}} (q_1, \varepsilon, ZZ_s)$ – *nicht akzeptiert*.
- $(q_s, abb, Z_s) \vdash_{\mathcal{A}} (q_s, bb, ZZ_s) \vdash_{\mathcal{A}} (q_1, b, Z_s) \vdash_{\mathcal{A}} (f, b, \varepsilon)$ – *nicht akzeptiert* (weil das Wort nicht zu Ende gelesen wurde – „b“ in (f, b, ε))

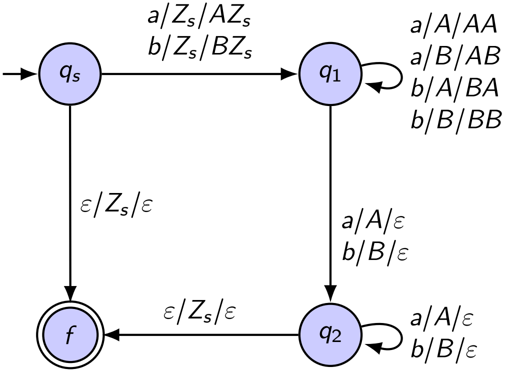
Beispiel fortgesetzt

Formal:

- $Q = \{q_s, q_1, f\}$
- $\Gamma = \{Z, Z_s\}$
- $\Sigma = \{a, b\}$
- $$\begin{aligned}\Delta = \{&(q_s, & a, && Z_s, && ZZ_s, && q_s),\\ &(q_s, & a, && Z, && ZZ, && q_s),\\ &(q_s, & b, && Z, && \varepsilon, && q_1),\\ &(q_1, & b, && Z, && \varepsilon, && q_1),\\ &(q_1, & \varepsilon, && Z_s, && \varepsilon, && f)\}\end{aligned}$$
- $F = \{f\}$

Weiteres Beispiel

- Ein PDA für $L = \{ww^R \mid w \in \{a,b\}^*\}$ (wobei für $w = a_1 \cdots a_n$ gilt $w^R = a_n \cdots a_1$).



Akzeptanz per leerem Keller

Akzeptanz per leerem Keller

- Ein *PDA mit Akzeptanz per leerem Keller* ist ein Tupel

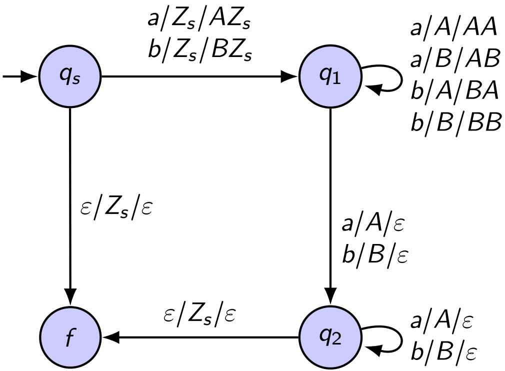
$$\mathcal{A} = (Q, \Sigma, \Gamma, q_s, Z_s, \Delta),$$

  wobei alle Komponenten wie für PDAs sind.
- Ein solcher PDA akzeptiert ein Eingabewort $w \in \Sigma^*$ gdw. $(q_s, w, Z_s) \vdash_{\mathcal{A}}^* (q, \varepsilon, \varepsilon)$ für ein beliebiges $q \in Q$.

- Konfiguration $(q, \varepsilon, \varepsilon)$ bedeutet, dass sich $\mathcal{A}$ in einem beliebigen Zustand befinden kann, das Eingabewort zu Ende gelesen haben muss und den *Keller vollständig geleert* haben muss (einschließlich des Kellerstartsymbols Z_s).

Beispiel wiederholt

- Ein PDA mit Akzeptanz per leerem Keller für $L = \{ww^R \mid w \in \{a,b\}^*\}$ (wobei für $w = a_1 \cdots a_n$ gilt $w^R = a_n \cdots a_1$).



Akzeptanz per leerem Keller

Satz

Jede Sprache L, die von einem PDA (mit akzeptierenden Zuständen) akzeptiert wird, wird auch von einem PDA mit Akzeptanz per leerem Keller akzeptiert und umgekehrt.

Beweis

Beweisidee:

„$\Rightarrow$“ Erstelle aus $\mathcal{A}$ einen Automaten $\mathcal{B}$:

- $\mathcal{B}$ arbeitet fast wie $\mathcal{A}$.
- Leere den Keller beim Erreichen eines Endzustandes.
- Verhindere das der Keller *zu früh* leer wird (sonst bleibt der Automat stecken)

„$\Leftarrow$“ Erstelle aus $\mathcal{B}$ einen Automaten $\mathcal{A}$:

- $\mathcal{A}$ arbeitet fast wie $\mathcal{B}$.
- Wenn ein leerer Keller ereicht ist, gehe zu Endzustand über.

Konstruktion: PDA mit akz. Zust. → akz. per leerem Keller

Beweis: „$\Rightarrow$“:

- Sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ ein PDA mit akzeptierenden Zuständen.
- Neuer Automat $\mathcal{B} = (Q \cup \{q_0', q_1'\}, \Sigma, \Gamma \cup \{Z_0'\}, q_0', Z_0', \Delta')$:
- Erweitere $\mathcal{A}$ um:
  - einen neuen Anfangszustand q_0'
  - einen neuen Zustand q_1' zum Leeren des Kellers
  - ein neues Kellerstartsymbol Z_0' (Z_0 ist nicht mehr Kellerstartsymbol).

  sowie die Übergänge

$$\begin{aligned}\Delta' = \Delta &\cup \{(q_0', \varepsilon, Z_0', Z_0 Z_0', q_0)\} && \text{(Von neuem Startzustand zu Altem)}\\ &\cup \{(q, \varepsilon, Z, \varepsilon, q_1') \mid q \in F, Z \in \Gamma \cup \{Z_0'\}\} && \text{(von Endzustand zum Leeren des Kellers)}\\ &\cup \{(q_1, \varepsilon, Z, \varepsilon, q_1') \mid \text{für } Z \in \Gamma \cup \{Z_0'\}\} && \text{(Leeren des Kellers).}\end{aligned}$$

- Keller wird leer genau dann, wenn in $\mathcal{A}$ ein Endzustand erreicht wird.

Konstruktion: PDA akz. per leerem Keller → PDA akz. Zust.

„$\Leftarrow$“:

- Sei $\mathcal{B} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ ein PDA mit Akzeptanz per leerem Keller.
- Neuer Automat $\mathcal{A} = (Q \cup \{q_0', f\}, \Sigma, \Gamma \cup \{Z_0'\}, q_0', Z_0', \Delta', \{f\})$:
- Erweitere $\mathcal{B}$ um:
  - einen neuen Anfangszustand q_0'
  - einen neuen Zustand f, der einziger Endzustand ist
  - ein neues Kellerstartsymbol Z_0' (Z_0 ist nichtmehr Kellerstartsymbol)

  sowie die Übergänge

$$\begin{aligned}\Delta' = \Delta &\cup \{(q_0', \varepsilon, Z_0', Z_0 Z_0', q_0)\} \\ &\cup \{(q, \varepsilon, Z_0', \varepsilon, f) \mid q \in Q\}\end{aligned}$$

- Endzustand wird erreicht genau dann, wenn Keller von $\mathcal{B}$ leer wird.

Kontextfreie Sprachen und Kellerautomaten

Satz

Für jede formale Sprache L sind äquivalent:

1) L wird von einer kontextfreien Grammatik erzeugt.
2) L wird von einem PDA akzeptiert.

Linksableitungen

Definition Linksableitung

Sei G eine kontextfreie Grammatik. Eine Ableitung

$$S = w_0 \vdash_G w_1 \vdash_G w_2 \vdash_G \cdots \vdash_G w_n$$

heißt *Linksableitung*, wenn sich w_{i+1} aus w_i (für alle $i < n$) durch *Anwendung einer Regel auf das am weitesten links stehende Nichtterminal in* w_i ergibt.

Lemma

Für jede kontextfreie Grammatik G gilt:

$$L(G) = \{w \in \Sigma^* \mid w \text{ kann in } G \text{ mit Linksableitung erzeugt werden}\}.$$

Linksableitungen Beweis

Beweisidee:

- Wenn $w \in L(G)$, dann gibt es eine Ableitung von w in G.
- Diese kann als Ableitungsbaum dargestellt werden.
- Aus dem Ableitungsbaum lässt sich eine Linksableitung von w ablesen.

Kontextfreie Sprachen und Kellerautomaten

Satz 1) ⇒ 2)

Sei L formale Sprache.

1) Wenn L von einer kontextfreien Grammatik erzeugt wird, dann
2) wird L von einem PDA akzeptiert.

Beweis.

- Es sei $G = (N, \Sigma, P, S)$ eine kontextfreie Grammatik.
- Wir konstruieren einen PDA $\mathcal{A}$ *mit Akzeptanz per leerem Keller* mit $L(\mathcal{A}) = L(G)$.
- Idee: $\mathcal{A}$ soll die Linksableitungen von G auf dem Keller simulieren.

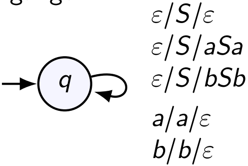
Beweis fortgesetzt

- Dazu definieren wir $\mathcal{A} = (Q, \Sigma, \Gamma, q_s, Z_s, \Delta)$ mit
  - $Q = \{q\}$, $\Gamma = \Sigma \cup N$, $q_s = q$, $Z_s = S$,
  - Δ besteht aus folgenden Übergängen:
  - Übergänge zum Anwenden von Produktionen auf das oberste Kellersymbol:
    - ▷ für jede Regel $A \to \gamma$ den Übergang $(q, \varepsilon, A, \gamma, q)$
  - Übergänge zum Entfernen bereits erzeugter Terminalsymbole von der Kellerspitze, wenn sie in der Eingabe vorhanden sind:
    - ▷ für jedes Terminalsymbol $a \in \Sigma$ den Übergang $(q, a, a, \varepsilon, q)$.

Beispiel

- $P = \{S \longrightarrow \varepsilon,\ S \longrightarrow aSa,\ S \longrightarrow bSb\}$ liefert die Übergänge:

$$
\begin{array}{llllll}
(q, & \varepsilon, & S, & \varepsilon, & q), & (S \longrightarrow \varepsilon) \\
(q, & \varepsilon, & S, & aSa, & q), & (S \longrightarrow aSa) \\
(q, & \varepsilon, & S, & bSb, & q), & (S \longrightarrow bSb) \\
(q, & a, & a, & \varepsilon, & q), & (a \text{ entfernen}) \\
(q, & b, & b, & \varepsilon, & q) & (b \text{ entfernen})
\end{array}
$$



- Die Ableitung $S \vdash_G aSa \vdash_G abSba \vdash_G abba$ entspricht der Konfigurationsfolge

$$
\begin{array}{llll}
(q, abba, S) \vdash_{\mathcal{A}} (q, abba, aSa) & \vdash_{\mathcal{A}} (q, bba, Sa) & \vdash_{\mathcal{A}} (q, bba, bSba) \vdash_{\mathcal{A}} \\
(q, ba, Sba) \vdash_{\mathcal{A}} (q, ba, ba) & \vdash_{\mathcal{A}} (q, a, a) & \vdash_{\mathcal{A}} (q, \varepsilon, \varepsilon)
\end{array}
$$

Beweis fortgesetzt

Lemma

Für alle $w \in \Sigma^*$ gilt:

$$S \vdash_G^* w \text{ mittels Linksableitung gdw. } (q, w, S) \vdash_{\mathcal{A}}^* (q, \varepsilon, \varepsilon).$$

Beweisidee:

- Wandle Linksableitung in akzeptierende Konfigurationsfolge um und umgekehrt.
- Jede Anwendung einer Produktion in G entspricht dabei genau
  - einer Anwendung eines Übergangs in $\mathcal{A}$ vom Typ 1,
  - gefolgt von Übergängen vom Typ 2, um je ein Nichtterminal a vom Keller zu löschen und gleichzeitig a in w zu lesen,

  und umgekehrt.

Kontextfreie Sprachen und Kellerautomaten

- Aus dem Lemma folgt: $L(G) = L(\mathcal{A})$ und somit

Satz 1) $\Rightarrow$ 2)

Sei L formale Sprache.

1) Wenn L von einer kontextfreien Grammatik erzeugt wird, dann
2) wird L von einem PDA akzeptiert.

Kontextfreie Sprachen und Kellerautomaten

Satz 2) $\Rightarrow$ 1)

Sei L formale Sprache.

1) Wenn L von einer kontextfreien Grammatik erzeugt wird, dann
2) wird L von einem PDA akzeptiert.

Beweis.

- Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ ein PDA mit Akzeptanz per leerem Keller.
- Wir möchten eine kontextfreie Grammatik G konstruieren mit $L(G) = L(\mathcal{A})$.

Beweis

- Die Nichtterminalsymbole von G sind alle Tripel

$$[p, Z, q] \in Q \times \Gamma \times Q.$$

- Idee: Es soll gelten: $[p, Z, q] \vdash_G^* u \in \Sigma^* \Leftrightarrow$
  - $\mathcal{A}$ erreicht *vom Zustand p aus den Zustand q* (in beliebig vielen Schritten)
  - durch *Lesen von u* auf dem Eingabeband und
  - *Löschen von Z* aus dem Keller ohne dabei die Symbole unter Z anzutasten.
- Dann gilt $w \in L(\mathcal{A}) \quad \Leftrightarrow \quad [q_0, Z_0, q] \vdash_G^* w$ für ein $q \in Q$.

Beweis (fortgesetzt)

- Die Produktionen werden diese Bedeutung der Symbole $[p, Z, q]$ garantieren.
- $\mathcal{A}$ erreicht von p aus q unter Lesen der Eingabe u und Löschen des Stacksymbols Z $\Leftrightarrow$
  - $u = a$ (für $a \in \Sigma \cup \{\varepsilon\}$) und es gibt Übergang $(p, a, Z, \varepsilon, q) \in \Delta$ oder
  - $u = av_1 \dots v_n$ und es gibt Übergang $(p, a, Z, X_1 \dots X_n, p_0) \in \Delta$ und
    - ▷ Zustände $p_1, \dots, p_{n-1}$, so dass
    - ▷ $[p_0, X_1, p_1]$ unter Lesen von v_1;
    - ▷ $[p_1, X_2, p_2]$ unter Lesen von v_2;
    - ▷ ...
    - ▷ $[p_{n-1}, X_n, q]$ unter Lesen von v_n.

Beweis (fortgesetzt)

- Unsere Grammatik enthält alle Produktionen (für $a \in \Sigma \cup \{\varepsilon\}$)

$$[p, Z, q] \longrightarrow a$$

für

$$(p, a, Z, \varepsilon, q) \in \Delta$$

- und alle Produktionen

$$[p, Z, q] \longrightarrow a[p_0, X_1, p_1] \cdots [p_{n-1}, X_n, q]$$

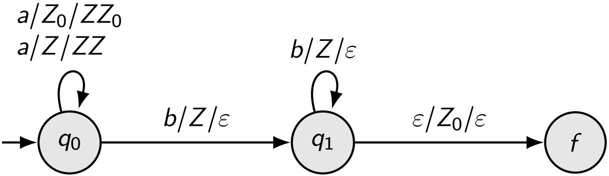
für $(p, a, Z, X_1 \dots X_n, p_0) \in \Delta$ und *alle möglichen Zustandsfolgen* $p_1, \dots, p_{n-1}$.

Beweis (fortgesetzt)

- Formal:

$$
\begin{aligned}
G &:= (N, \Sigma, P, S) \text{ mit} \\
N &:= \{S\} \cup \{[p, Z, q] \mid p, q \in Q, Z \in \Gamma\} \\
P &:= \{S \longrightarrow [q_0, Z_0, q] \mid q \in Q\} \cup \\
&\quad\ \{[p, Z, q] \longrightarrow a \mid (p, a, Z, \varepsilon, q) \in \Delta \text{ mit } a \in \Sigma \cup \{\varepsilon\}\ \} \cup \\
&\quad\ \{[p, Z, q] \longrightarrow a[p_0, X_1, p_1][p_1, X_2, p_2] \ldots [p_{n-1}, X_n, q] \mid \\
&\qquad\qquad (p, a, Z, X_1 \ldots X_n, p_0) \in \Delta,\ \ q \in Q, \\
&\qquad\qquad a \in \Sigma \cup \{\varepsilon\}, \\
&\qquad\qquad p_1, \ldots, p_{n-1} \in Q, \\
&\qquad\qquad n \geq 1\}
\end{aligned}
$$

Beispiel



- PDA mit Akzeptanz per leerem Keller für $\{a^n b^n \ : \ n \geq 1\}$.
- Produktionen für $(q_0, b, Z, \varepsilon, q_1)$: $[q_0, Z, q_1] \longrightarrow b$.
- Produktionen für $(q_1, b, Z, \varepsilon, q_1)$: $[q_1, Z, q_1] \longrightarrow b$.
- Produktionen für $(q_1, \varepsilon, Z_0, \varepsilon, f)$: $[q_1, Z_0, f] \longrightarrow \varepsilon$.

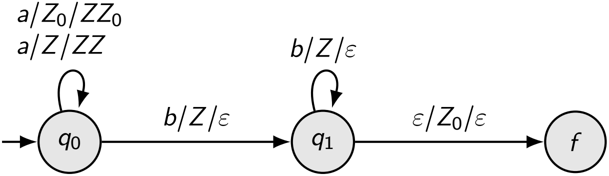
Beispiel (fortgesetzt)

Erinnerung:

$$\{[p,Z,q] \longrightarrow a[p_0, X_1, p_1][p_1, X_2, p_2] \ldots [p_{n-1}, X_n, q] \mid (p, a, Z, X_1 \ldots X_n, p_0) \in \Delta,\ q \in Q, a \in \Sigma \cup \{\varepsilon\}, p_1, \ldots, p_{n-1} \in Q, n \geq 1\}$$

- Produktionen für (q_0, a, Z_0, ZZ_0, q_0) :
  - $[q_0, Z_0, q_0] \longrightarrow a[q_0, Z, q_0][q_0, Z_0, q_0]$
  - $[q_0, Z_0, q_0] \longrightarrow a[q_0, Z, q_1][q_1, Z_0, q_0]$
  - $[q_0, Z_0, q_0] \longrightarrow a[q_0, Z, f][f, Z_0, q_0]$
  - $[q_0, Z_0, q_1] \longrightarrow a[q_0, Z, q_0][q_0, Z_0, q_1]$
  - $[q_0, Z_0, q_1] \longrightarrow a[q_0, Z, q_1][q_1, Z_0, q_1]$
  - $[q_1, Z_0, q_0] \longrightarrow a[q_0, Z, f][f, Z_0, q_1]$
  - $[q_0, Z_0, f] \longrightarrow a[q_0, Z, q_0][q_0, Z_0, f]$
  - $[q_0, Z_0, f] \longrightarrow a[q_0, Z, q_1][q_1, Z_0, f]$
  - $[q_0, Z_0, f] \longrightarrow a[q_0, Z, f][f, Z_0, f]$

Beispiel (fortgesetzt)



- Berechnung von $\mathcal{A}$ auf Wort ab:
  - $(q_0, ab, Z_0) \vdash_{\mathcal{A}} (q_0, b, ZZ_0) \vdash_{\mathcal{A}} (q_1, \varepsilon, Z_0) \vdash_{\mathcal{A}} (f, \varepsilon, \varepsilon)$.
- Zugehörige Ableitung in G:
  - $S \vdash_G [q_0, Z_0, f] \vdash_G a[q_0, Z, q_1][q_1, Z_0, f] \vdash_G ab[q_1, Z_0, f] \vdash_G ab$.

Beweis Korrektheit der Konstruktion

Behauptung

Für alle $p, q \in Q, u \in \Sigma^*, Z \in \Gamma, \gamma \in \Gamma^*$ gilt:

$$[p, Z, q] \vdash_G^* u \Leftrightarrow (p, u, Z) \vdash_{\mathcal{A}}^* (q, \varepsilon, \varepsilon)$$

Beweis.

- $\Rightarrow$ Per Induktion über die Länge der Ableitung.
- $\Leftarrow$ per Induktion über die Länge der Konfigurationsfolge.

- Für $p = q_0$ und $Z = Z_0$ folgt daraus:

$$S \vdash_G [q_0, Z_0, q] \vdash_G^* u \Leftrightarrow S(q_0, u, Z_0) \vdash_{\mathcal{A}}^* (q, \varepsilon, \varepsilon),$$

d. h. $u \in L(G) \Leftrightarrow u \in L(\mathcal{A})$.

Korollar

- Wir nennen zwei PDAs $\mathcal{A}$ und $\mathcal{A}'$ *äquivalent*, wenn $L(\mathcal{A}) = L(\mathcal{A}')$.

Korollar

Zu jedem PDA $\mathcal{A}$ gibt es einen PDA $\mathcal{A}'$ mit Akzeptanz per leerem Keller, so dass $L(\mathcal{A}) = L(\mathcal{A}')$ und $\mathcal{A}'$ nur einen Zustand hat.

Beweis.

- Gegeben einen PDA $\mathcal{A}$, wende erst die Konstruktion „$2 \Rightarrow 1$" an: $\mathcal{A} \to G$
- Wende dann die Konstruktion „$1 \Rightarrow 2$" an: $G \to \mathcal{A}'$.
- $L(\mathcal{A}') = L(\mathcal{A})$ und $\mathcal{A}'$ hat nur einen Zustand.

Abschluss unter Schnitt

- Die kontextfreien Sprachen sind nicht unter Schnitt abgeschlossen.
- Beispiel:
  - $L_1 = \{a^i b^i c^j \ : \ i, j \geq 0\}$,
  - $L_2 = \{a^i b^j c^j \ : \ i, j \geq 0\}$,
  - $L_1 \cap L_2 = \{a^i b^i c^i \ : \ i \geq 0\}$ ist nicht kontextfrei.
  - Beweis mit Pumping Lemma.
- Aber:

Satz

Es sei $L \subseteq \Sigma^*$ kontextfrei und $R \subseteq \Sigma^*$ regulär. Dann ist $L \cap R$ kontextfrei.

Beweisidee: Nutze beide Automaten synchron (Produktkonstruktion)

Beweis

- Sei $L = L(\mathcal{A})$ für einen PDA $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ (also mit akzeptierenden Zuständen).
- Sei $R = L(\mathcal{A}')$ für einen DEA $\mathcal{A}' = (Q', \Sigma, q_0', \delta', F')$.
- Produktkonstruktion für $L \cap R$:

$$
\begin{aligned}
\mathcal{B} &:= (Q \times Q', \Sigma, \Gamma, (q_0, q_0'), Z_0, \Delta', F \times F') \text{ mit} \\
\Delta' &:= \{((p,p'), a, Z, \gamma, (q,q')) \;:\; (p,a,Z,\gamma,q) \in \Delta,\ \delta(p',a) = q'\} \cup \\
&\quad\ \{((p,p'), \varepsilon, Z, \gamma, (q,p')) \;:\; (p,\varepsilon,Z,\gamma,q) \in \Delta, p' \in Q'\}
\end{aligned}
$$

- Induktion über k: $((p,p'), uv, \gamma) \vdash_{\mathcal{B}}^{k} ((q,q'), v, \beta) \Leftrightarrow (p, uv, \gamma) \vdash_{\mathcal{A}}^{k} (q, v, \beta)$ und $p' \xrightarrow{u}_{\mathcal{A}'} q'$.
- Für zwei PDAs funktioniert eine solche Produktkonstruktion nicht, da die beiden PDAs den Keller im Allgemeinen nicht synchron nutzen.