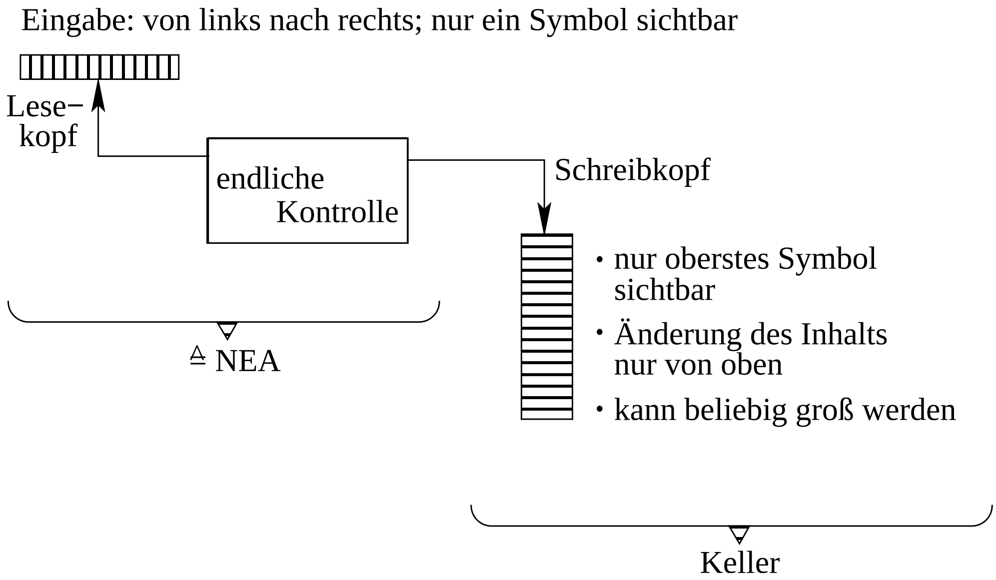
10 Kellerautomaten

Bisher hatten wir kontextfreie Sprachen nur mit Hilfe von Grammatiken charakterisiert. Wir haben gesehen, dass endliche Automaten *nicht* in der Lage sind, alle kontextfreien Sprachen zu akzeptieren.

Um die *Beschreibung von kontextfreien Sprachen* mit Hilfe von endlichen Automaten zu ermöglichen, muss man diese um eine (potentiell unendliche) *Speicherkomponente*, einen sogenannten *Keller*, erweitern.

Die folgende Abbildung zeigt eine schematische Darstellung eines *Kellerautomaten*:



Definition 10.1 (Kellerautomat)
Ein *Kellerautomat (pushdown automaton, kurz PDA)* hat die Form
$\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$, wobei

- Q eine endliche Menge von *Zuständen* ist,
- Σ das *Eingabealphabet* ist,
- Γ das *Kelleralphabet* ist,
- $q_0 \in Q$ der *Anfangszustand* ist,
- $Z_0 \in \Gamma$ das *Kellerstartsymbol* ist und
- $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q$ eine endliche *Übergangsrelation* ist,
- $F \subseteq Q$ eine Menge von Endzuständen ist.

Anschaulich bedeutet die Übergangsrelation:

(q, a, Z, γ, q'): Im Zustand q mit aktuellem Eingabesymbol a und oberstem Kellersymbol Z darf der Automat Z durch γ ersetzen und in den Zustand q' und zum nächsten Eingabesymbol übergehen.

$(q, \varepsilon, Z, \gamma, q')$: wie oben, nur dass das aktuelle Eingabesymbol nicht relevant ist und man nicht zum nächsten Eingabesymbol übergeht (der Lesekopf ändert seine Position nicht).

Den aktuellen *Zustand* einer Kellerautomatenberechnung kann man beschreiben durch

- den *noch zu lesenden Rest* $w \in \Sigma^*$ der Eingabe (Lesekopf steht auf dem ersten Symbol von w)
- den *Zustand* $q \in Q$
- den *Kellerinhalt* $\gamma \in \Gamma^*$ (Schreiblesekopf steht auf dem ersten Symbol von γ)

Definition 10.2
Eine *Konfiguration* von $\mathcal{A}$ hat die Form

$$\mathcal{K} = (q, w, \gamma) \in Q \times \Sigma^* \times \Gamma^*.$$

Die Übergangsrelation ermöglicht die folgenden *Konfigurationsübergänge*:

- $(q, aw, Z\gamma) \vdash (q', w, \beta\gamma)$ falls $(q, a, Z, \beta, q') \in \Delta$
- $(q, w, Z\gamma) \vdash (q', w, \beta\gamma)$ falls $(q, \varepsilon, Z, \beta, q') \in \Delta$
- $\mathcal{K} \vdash^* \mathcal{K}'$ gdw. $\exists n \geq 0\ \exists \mathcal{K}_0, \ldots, \mathcal{K}_n$ mit

$$\mathcal{K} = \mathcal{K}_0 \vdash \mathcal{K}_1 \vdash \ldots \mathcal{K}_n = \mathcal{K}'.$$

Der Automat $\mathcal{A}$ *akzeptiert* das Wort $w \in \Sigma^*$ gdw. $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \gamma)$ mit $q \in F, \gamma \in \Gamma^*$. (Eingabewort ganz gelesen und Endzustand erreicht).

Die von $\mathcal{A}$ *akzeptierte Sprache* ist

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid \mathcal{A} \text{ akzeptiert } w\}.$$

Beispiel 10.3
Ein PDA für $\{a^n b^n \mid n \geq 1\}$.

- $Q = \{q_0, q_1, f\}$,
- $\Gamma = \{Z, Z_0\}$,
- $\Sigma = \{a, b\}$ und
- $\Delta = \{(q_0, a, Z_0, ZZ_0, q_0),$ (erstes a, speichere Z)
  $(q_0, a, Z, ZZ, q_0),$ (weitere a's, speichere Z)
  $(q_0, b, Z, \varepsilon, q_1),$ (erstes b, entnimm Z)
  $(q_1, b, Z, \varepsilon, q_1),$ (weitere b's, entnimm Z)
  $(q_1, \varepsilon, Z_0, \varepsilon, f)\}$ (lösche Z_0, gehe in Endzustand)

Betrachten wir uns einige Konfigurationsübergänge:

1) $(q_0, aabb, Z_0) \vdash (q_0, abb, ZZ_0) \vdash (q_0, bb, ZZZ_0) \vdash (q_1, b, ZZ_0) \vdash (q_1, \varepsilon, Z_0) \vdash (f, \varepsilon, \varepsilon)$
   – akzeptiert
2) $(q_0, aab, Z_0) \vdash^* (q_0, b, ZZZ_0) \vdash (q_1, \varepsilon, ZZ_0)$
   – kein Übergang mehr möglich
3) $(q_0, abb, Z_0) \vdash (q_0, bb, ZZ_0) \vdash (q_1, b, Z_0) \vdash (f, b, \varepsilon)$
   – nicht akzeptiert

Der Kellerautomat aus Beispiel 10.3 ist deterministisch, da es zu jeder Konfiguration höchstens eine Folgekonfiguration gibt.

Definition 10.4
Der PDA $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ heißt *deterministisch*, falls die folgenden beiden Bedingungen erfüllt sind:

1) $\forall q \in Q\ \forall a \in \Sigma \cup \{\varepsilon\}\ \forall Z \in \Gamma$ existiert höchstens ein Übergang der Form $(q, a, Z, \ldots, \ldots) \in \Delta$.
2) Existiert ein Tupel $(q, \varepsilon, Z, \ldots, \ldots) \in \Delta$, so existiert kein Tupel der Form $(q, a, Z, \ldots, \ldots) \in \Delta$ mit $a \in \Sigma$.

Beispiel 10.5
Die Sprache $L = \{w \overleftarrow{w} \mid w \in \{a, b\}^*\}$ (wobei für $w = a_1 \ldots a_n$ gilt $\overleftarrow{w} = a_n \ldots a_1$) wird von einem nichtdeterministischen PDA akzeptiert.

Idee: Der PDA legt die erste Worthälfte im Keller ab und greift darauf *in umgekehrter Reihenfolge* beim Lesen der zweiten Worthälfte zurück. So kann Nichtübereinstimmung festgestellt werden (kein Übergang bei Nichtübereinstimmung).

Nichtdeterminismus ist intuitiv nötig, da man „raten“ muss, wann die Wortmitte erreicht ist (Man kann beweisen, dass kein deterministischer PDA diese Sprache akzeptiert).

$Q = \{q_0, q_1, q_2, f\},\ \Gamma = \{a, b, Z_0\},\ \Sigma = \{a, b\},\ F = \{f\},\ \Delta =$

$\{(q_0,$	$\varepsilon,$	$Z_0,$	$\varepsilon,$	$f),$	akzeptiert ε
$(q_0,$	$a/b,$	$Z_0,$	$a/bZ_0,$	$q_1),$	Lesen
$(q_1,$	$a,$	$b,$	$ab,$	$q_1),$	und
$(q_1,$	$b,$	$a,$	$ba,$	$q_1),$	Speichern
$(q_1,$	$a,$	$a,$	$aa,$	$q_1),$	der ersten
$(q_1,$	$b,$	$b,$	$bb,$	$q_1),$	Worthälfte
$(q_1,$	$b,$	$b,$	$\varepsilon,$	$q_2),$	Lesen der
$(q_1,$	$a,$	$a,$	$\varepsilon,$	$q_2),$	zweiten Hälfte
$(q_2,$	$b,$	$b,$	$\varepsilon,$	$q_2),$	und Vergleichen
$(q_2,$	$a,$	$a,$	$\varepsilon,$	$q_2),$	mit erster
$(q_2,$	$\varepsilon,$	$Z_0,$	$\varepsilon,$	$f)\}$	Übergang zu Endzustand

Für Kellerautomaten gibt es noch einen anderen (äquivalenten) Akzeptanzbegriff:

Definition 10.6

Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ ein PDA ohne ausgezeichnete Endzustandsmenge. Wir sagen: $\mathcal{A}$ *akzeptiert w mit leerem Keller*, falls

$$(q_0, w, z_0) \vdash^* (q, \varepsilon, \varepsilon) \text{ für ein beliebiges } q \in Q.$$

Die durch $\mathcal{A}$ mit leerem Keller akzeptierte Sprache ist

$$N(\mathcal{A}) := \{w \in \Sigma^* \mid \mathcal{A} \text{ akzeptiert mit leerem Keller}\}.$$

Beispiel

Die PDAs aus Beispiel 10.3 und 10.5 akzeptieren die entsprechende Sprache sowohl mit Endzustandsmenge $\{f\}$ als auch mit leerem Keller, denn nur beim Übergang zu f wird Z_0 entfernt und damit der Keller leer.

Satz 10.7

Für PDAs ist Akzeptanz mit Endzuständen äquivalent zu Akzeptanz mit leerem Keller, d. h. für eine formale Sprache L sind äquivalent:

1. $L = L(\mathcal{A})$ *für einen PDA* $\mathcal{A}$.
2. $L = N(\mathcal{B})$ *für einen PDA* $\mathcal{B}$.

Beweis.

„$1 \longrightarrow 2$“: $\mathcal{B}$ arbeitet im Prinzip wie $\mathcal{A}$. Bei Erreichen eines Endzustands leert $\mathcal{B}$ noch den Keller. Zusätzlich muss verhindert werden, dass der Keller leer wird, ohne dass ein Endzustand erreicht war:
$\mathcal{A}$ wird erweitert um

- einen neuen Anfangszustand ${q_0}'$,
- einen neuen Zustand q_1 zum Leeren des Kellers,
- ein neues Startsymbol X_0 (Z_0 ist nichtmehr Startsymbol),

sowie die Übergänge

$$\begin{array}{llllll}
({q_0}', & \varepsilon, & X_0, & Z_0X_0, & q_0), & \\
(q, & \varepsilon, & Z, & \varepsilon, & q_1) & \text{für } q \in F,\ Z \in \Gamma \cup \{X_0\}, \\
(q_1, & \varepsilon, & Z, & \varepsilon, & q_1) & \text{für } Z \in \Gamma \cup \{X_0\}.
\end{array}$$

Beachte:

Da X_0 in den Übergängen von $\mathcal{A}$ nicht vorkommt, kann es nur entfernt werden, wenn vorher ein Endzustand erreicht war.

„$2 \longrightarrow 1$“: $\mathcal{A}$ arbeitet im Prinzip wie $\mathcal{B}$. Zusätzlich muss $\mathcal{A}$ feststellen, wenn der Keller geleert ist und dann in einen Endzustand übergehen. Erkennen des leeren Kellers (bei $\mathcal{B}$) wird durch ein neues Kellerstartsymbol erreicht:
$\mathcal{B}$ wird ergänzt um

- einen neuen Anfangszustand $q_0{}'$,
- einen neuen Zustand f, der einziger Endzustand ist,
- ein neues Kellerstartsymbol X_0 (Z_0 ist nichtmehr Startsymbol),

sowie die Übergänge

$$\begin{array}{llllll} (q_0{}', & \varepsilon, & X_0, & Z_0X_0, & q_0), & \\ (q, & \varepsilon, & X_0, & \varepsilon, & f) & \quad \text{für alle } q \in Q \ (\widehat{=}\ \mathcal{B} \text{ hat seinen Keller geleert}). \end{array}$$

□

Wir werden nun zeigen, dass man mit Kellerautomaten genau die kontextfreien Sprachen akzeptieren kann.

Satz 10.8

Für eine formale Sprache L sind äquivalent:

1) $L = L(G)$ *für eine kontextfreie Grammatik* G.

2) $L = L(\mathcal{A})$ *für einen PDA* $\mathcal{A}$.

Beweis.

„$1 \longrightarrow 2$“: Es sei $G = (N, \Sigma, P, S)$ eine kontextfreie Grammatik. Der zugehörige PDA simuliert Linksableitungen von G, d.h. Ableitungen, bei denen stets das am weitesten links stehende Nichtterminalsymbol zuerst ersetzt wird.

Beachte:

Jedes Wort in $L(G)$ kann durch eine Linksableitung erzeugt werden (da G kontextfrei ist).

Wir definieren dazu $\mathcal{A} = (\{q\}, \Sigma, \Sigma \cup N, q, S, \Delta)$ mit $\Delta :=$

$\{(q, \varepsilon, A, \gamma, q) \mid A \longrightarrow \gamma \in P\} \cup$	(Anwenden einer Produktion auf oberstes Kellersymbol) $(\star)$
$\{(q, a, a, \varepsilon, q) \mid a \in \Sigma\}$	(Entfernen bereits erzeugter Terminalsymbole von der Kellerspitze, wenn sie in der Eingabe vorhanden sind) $(\star\star)$

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 $S \vdash aSa \vdash abSba \vdash abba$ entspricht der *Konfigurationsfolge*

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

Behauptung:
Für $u, v \in \Sigma^*$ und $\alpha \in \{\varepsilon\} \cup N \cdot (\Sigma \cup N)^*$ gilt:

$$S \vdash_G^* u\alpha \text{ mit Linksableitung} \quad \underline{\text{gdw}}. \quad (q, uv, S) \vdash^* (q, v, \alpha).$$

Beachte:
Für $\alpha = \varepsilon = v$ folgt:

$$S \vdash_G^* u \quad \underline{\text{gdw}}. \quad (q, u, S) \vdash^* (q, \varepsilon, \varepsilon)$$

d.h. $L(G) = N(\mathcal{A})$.

Beweis der Behauptung.

„$\Leftarrow$“: Beweis durch Induktion über die Anzahl k der Übergänge mit Transitionen der Form $(\star) : (q, \varepsilon, A, \gamma, q)$.

$k = 0$:
Da im Keller S steht, sind auch keine Transitionen der Form $(\star\star)$: $(q, a, a, \varepsilon, q)$ möglich, d.h. $u = \varepsilon$ und $\alpha = S$. Offenbar gilt $S \vdash^* S$.

$k \to k+1$:

$$(q, \underbrace{u_1 u_2}_{u} v, S) \vdash^* (q, u_2 v, A\alpha') \overset{\text{letzte Trans. }(\star)}{\vdash} (q, u_2 v, \underbrace{\gamma\alpha'}_{u_2\alpha}) \overset{\text{Trans. }(\star\star)}{\vdash^*} (q, v, \alpha)$$

Induktion liefert: $S \vdash^* u_1 A\alpha'$ und außerdem wegen $A \longrightarrow \gamma \in P$:

$$u_1 A\alpha' \vdash u_1\gamma\alpha' = u_1 u_2 \alpha = u\alpha.$$

„$\Rightarrow$“: Beweis durch Induktion über die Länge der Linksableitung

$k = 0$:
Dann ist $u = \varepsilon,\ \alpha = S$ und $(q, v, S) \vdash^0 (q, v, S)$

$k \to k+1$:
$S \vdash_G^{k+1} u\alpha$, d.h.
$S \vdash_G^k u'A\beta \vdash \underbrace{u'\gamma\beta}_{=u\alpha}$ mit $u' \in \Sigma^*$ (da Linksableitung) und $A \longrightarrow \gamma \in P$.

Es sei u'' das längste Anfangsstück von $\gamma\beta$, das in Σ^* liegt. Dann ist $u'u'' = u$ und α ist der Rest von $\gamma\beta$, d.h. $\gamma\beta = u''\alpha$.

Induktion liefert:

$$(q, u' \underbrace{u'' v}_{v'}, S) \vdash^* (q, u''v, A\beta) \vdash (q, u''v, \gamma\beta) = (q, u''v, u''\alpha).$$

Außerdem gibt es Übergänge $(q, u''v, u''\alpha) \vdash^* (q, v, \alpha)$. $\square$

„2 $\longrightarrow$ 1“: Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ ein PDA. Die Nichtterminalsymbole der zugehörigen Grammatik G sind alle Tripel $[p, Z, q] \in Q \times \Gamma \times Q$.

Idee:

Es soll gelten: $[p, Z, q] \vdash_G^* u \in \Sigma^*$ <u>gdw</u>.

- $\mathcal{A}$ kommt *vom Zustand p in den Zustand q*
- durch *Lesen von u* auf dem Eingabeband und
- *Löschen von Z* aus dem Keller

(ohne dabei die Symbole unter Z anzutasten).

Wie realisiert man dies durch geeignete Produktionen?

Betrachte den PDA-Übergang $(p, a, Z, X_1 \dots X_n, p') \in \Delta$. Hier wird a auf dem Eingabeband verbraucht (falls nicht $a = \varepsilon$). Z wird durch $X_1 \dots X_n$ ersetzt.
Um den Kellerinhalt zu erreichen, den man durch einfaches Wegstreichen von Z erhalten würde, muss man also nun noch $X_1 \dots X_n$ löschen. Löschen von X_i kann durch das Nichtterminalsymbol $[p_{i-1}, X_i, p_i]$ (für geeignete Zwischenzustände p_{i-1}, p_i) beschrieben werden.

Formale Definition:

$$\begin{aligned}
G &:= (N, \Sigma, P, S) \text{ mit} \\
N &:= \{S\} \cup \{[p, Z, q] \mid p, q \in Q, Z \in \Gamma\} \\
P &:= \{S \longrightarrow [q_0, Z_0, q] \mid q \in Q\} \cup \\
&\quad\ \{[p, Z, q] \longrightarrow a[p_0, X_1, p_1][p_1, X_2, p_2] \dots [p_{n-1}, X_n, q] \mid \\
&\qquad\quad p, q, p_0, p_1, \dots, p_{n-1} \in Q, \\
&\qquad\quad a \in \Sigma \cup \{\varepsilon\} \text{ und} \\
&\qquad\quad (p, a, Z, X_1 \dots X_n, p_0) \in \Delta \text{ und} \\
&\qquad\quad q = p_0 \text{ falls } n = 0\}
\end{aligned}$$

Beachte:

Für $n = 0$ hat man die Produktion $[p, Z, q] \longrightarrow a$, welche dem Übergang $(p, a, Z, \varepsilon, q) \in \Delta$ entspricht.

Behauptung:

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

$$(\star) \qquad (p, u, Z) \vdash_{\mathcal{A}}^* (q, \varepsilon, \varepsilon) \ \ \underline{\text{gdw}}. \ \ [p, Z, q] \vdash_G^* u$$

Der Beweis dieser Behauptung kann durch Induktion über die Länge der Konfigurationsfolge („$\Rightarrow$“) bzw. über die Länge der Ableitung („$\Leftarrow$“) geführt werden.
Für $p = q_0$ und $Z = Z_0$ folgt daraus:

$$(q_0, u, Z_0) \vdash^* (q, \varepsilon, \varepsilon) \ \ \underline{\text{gdw}}. \ \ S \vdash_G [q_0, Z_0, q] \vdash_G^* u$$

d.h. $u \in N(\mathcal{A})$ <u>gdw</u>. $u \in L(G)$.

□

Beispiel: (vgl. Beispiel 10.3)
Gegeben ist der PDA für $\{a^n b^n \mid n \geq 1\}$.

$$(q_0, ab, Z_0) \overset{(q_0,a,Z_0,ZZ_0,q_0)}{\vdash} (q_0, b, ZZ_0) \overset{(q_0,b,Z,\varepsilon,q_1)}{\vdash} (q_1, \varepsilon, Z_0) \overset{(q_1,\varepsilon,Z_0,\varepsilon,f)}{\vdash} (f, \varepsilon, \varepsilon)$$

entspricht der Ableitung

$$S \vdash_G [q_0, Z_0, f] \vdash_G a[q_0, Z, q_1][q_1, Z_0, f] \vdash_G ab[q_1, Z_0, f] \vdash_G ab.$$

Wegen der gerade gezeigten Äquivalenz zwischen kontextfreien Sprachen und PDA akzeptierbaren Sprachen, kann man Eigenschaften von kontextfreien Sprachen mit Hilfe von Eigenschaften von Kellerautomaten zeigen.

Als Beispiel betrachten wir den Durchschnitt von kontextfreien Sprachen mit regulären Sprachen. Wir wissen: der Durchschnitt zweier kontextfreier Sprachen muss nicht kontextfrei sein. Dahingegen gilt:

Korollar 10.9
Es sei $L \subseteq \Sigma^*$ kontextfrei und $R \subseteq \Sigma^*$ regulär. Dann ist $L \cap R$ kontextfrei.

Beweis.
Es sei

- $L = L(\mathcal{A})$ für einen PDA $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ und
- $R = L(\mathcal{A}')$ für einen DEA $\mathcal{A}' = (Q', \Sigma, q_0', \delta', F')$.

Wir konstruieren den „Produktautomaten“ von $\mathcal{A}$ und $\mathcal{A}$', der ein PDA für $L \cap R$ ist:

$$\mathcal{B} := \ (Q \times Q', \Sigma, \Gamma, (q_0, q_0{}'), Z_0, \Delta', F \times F')$$

mit

$$\begin{aligned}\Delta' = \ & \{((p,p'), a, Z, \gamma, (q,q')) \mid a \in \Sigma, (p, a, Z, \gamma, q) \in \Delta \text{ und } \delta(p', a) = q'\} \ \cup \\ & \{((p,p'), \varepsilon, Z, \gamma, (q,p')) \mid (p, \varepsilon, Z, \gamma, q) \in \Delta\}\end{aligned}$$

Man zeigt nun leicht (durch Induktion über k):

$$((p,p'), uv, \gamma) \vdash^k_{\mathcal{B}} ((q,q'), v, \beta) \qquad \text{g}\underline{\text{dw}}. \qquad (p, uv, \gamma) \vdash^k_{\mathcal{A}} (q, v, \beta) \text{ und } p' \xrightarrow{u}_{\mathcal{A}'} q'.$$

□

Bemerkung 10.10.
Ist $\mathcal{A}$ in dem Beweis ein deterministischer PDA, so ist der konstruierte Produktautomat $\mathcal{B}$ auch deterministisch, wenn man $\mathcal{A}$' o. E. deterministisch wählt. Man kann diese Abschlusseigenschaften von kontextfreien Sprachen u. a. dazu verwenden, für gewisse Sprachen nachzuweisen, dass sie <u>nicht</u> kontextfrei sind.

Beispiel 10.11

Die Sprache $L = \{ww \mid w \in \{a,b\}^*\}$ ist nicht kontextfrei.

Beweis.

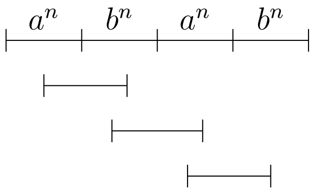
Angenommen, L ist kontextfrei. Dann ist mit Korollar 10.9 auch

$$L' := L \cap a^+b^+a^+b^+ = \{a^ib^ja^ib^j \mid i,j \geq 0\}$$

kontextfrei.

Es sei nun zu L' die Zahl n wie im Pumping Lemma. Wir betrachten $z = a^ib^ja^ib^j \in L'$. Es gibt eine Zerlegung $z = uvwxy$ mit $|vwx| \leq n$ und $vx \neq \varepsilon$, sodass $uwy \in L'$.

Wegen $|vwx| \leq n$ enthält aber vwx höchstens 2 der aufeinanderfolgenden a^n/b^n-Blöcke:



1. Fall
2. Fall
3. Fall

In allen 3 Fällen kann uwy kein Element von L' sein. □

Für endliche Automaten hatten wir gesehen, dass bereits die deterministischen endlichen Automaten alle erkennbaren Sprachen akzeptieren. Dazu haben wir gezeigt, dass man zu jedem NEA einen äquivalenten DEA konstruieren kann. Für Kellerautomaten ist dies nicht möglich.

Definition 10.12

Eine Sprache heißt *deterministisch kontextfrei* (dkf), falls sie von einem deterministischen Kellerautomaten (DPDA) akzeptiert wird.

Wir werden zeigen, dass die Sprache

$$L = \{w\overleftarrow{w} \mid w \in \{a,b\}^*\}$$

nicht dkf ist. Wir haben bereits gesehen (Beispiel 10.5), dass L kontextfrei ist. Zunächst benötigen wir aber eine Definition und ein Lemma.

Definition 10.13

Zu einer Sprache $L \subseteq \Sigma^*$ sei

$$Min(L) := \{w \in L \mid \text{kein echtes Präfix von } w \text{ liegt in } L\}.$$

Lemma 10.14

Ist L dkf, so auch $Min(L)$.

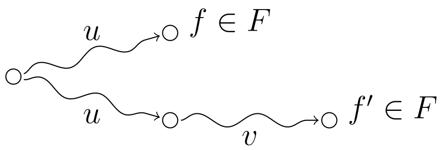
Beweis.
Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ ein DPDA für L, d. h. $L = L(\mathcal{A})$ (Akzeptanz mit Endzustand).

Wir ändern $\mathcal{A}$ so ab, dass gilt: wurde das erste Mal ein Endzustand erreicht, so kann danach keiner mehr erreicht werden.

$$\begin{aligned}
\mathcal{A}' &:= (Q', \Gamma, q_0, Z_0, \Delta', F) \quad \text{mit} \\
Q' &:= \ Q \cup \{\hat{q}\} \\
\Delta' &:= (\Delta \setminus F \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q) \\
&\qquad \cup \ \{(q, a, Z, Z, \hat{q}) \mid q \in F, a \in \Sigma, Z \in \Gamma\} \\
&\qquad \cup \ \{(\hat{q}, a, Z, Z, \hat{q}) \mid a \in \Sigma, Z \in \Gamma\}
\end{aligned}$$

Man sieht leicht: $L(\mathcal{A}') = \mathit{Min}(L)$ und $\mathcal{A}'$ ist deterministisch. □

Beachte:
Bei einem nicht-det. PDA funktioniert die Konstruktion nicht, da es für Eingabe u verschiedene Konfigurationsfolgen geben kann.



Satz 10.15
Die Sprache $L := \{w\overleftarrow{w} \mid w \in \{a,b\}^*$ *ist nicht deterministisch kontextfrei.*

Beweis.
Wäre L dkf, so mit Bemerkung 10.10 und Lemma 10.14 auch

$$L' := \mathit{Min}(L) \cap (ab)^+(ba)^+(ab)^+(ba)^+.$$

Es ist

$$L' = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i > j > 0\}$$

(für $i \le j$ wäre $(ab)^i(ba)^i \in L$ echtes Präfix).

Mit Hilfe des Pumping Lemmas zeigt man, dass L' nicht kf.

Angenommen doch. Dann sei n die Zahl aus dem Pumping Lemma für L'. Dann hat

$$z = (ab)^{n+1}(ba)^n(ab)^n(ba)^{n+1}$$

eine Zerlegung $z = uvwxy$ mit $|vwx| \le n$, $vx \ne \varepsilon$ und $uv^iwx^iy \in L$ für alle $i \ge 0$.

1. Fall: vwx im Mittelteil $(ba)^n(ab)^n$ Aufpumpen liefert ein Wort mit „$i \le j$“, d. h. $\notin \mathit{Min}(L)$.

2. Fall: vwx im linken Teil $(ab)^{n+1}(ba)^n$ oder im rechten Teil $(ab)^n(ba)^{n+1}$ Aufpumpen liefert ein Wort, das nicht in L ist.

$\square$

Deterministisch kontextfreie Sprachen sind für die *Syntaxanalyse* von Programmen interessant, da für sie das Wortproblem <u>linear</u> (im Gegensatz zu kubisch beim CYK-Algorithmus für allgemeine kontextfreie Sprachen) entscheidbar ist.

Dies ist wichtig im Compilerbau (siehe z. B. Ingo Wegener: Theoretische Informatik, S. 199-216).

Im Gegensatz zu den kf Sprachen ist die Klasse der dkf Sprachen unter Komplement abgeschlossen.

Satz 10.16
Ist L deterministisch kontextfrei, so auch $\overline{L}$.

(ohne Beweis)

Beachte:
Für einen DPDA, der L mit Endzuständen aus F akzeptiert, genügt es nicht, als neue Endzustandsmenge einfach $Q \setminus F$ zu nehmen, um einen Automaten für $\overline{L}$ zu erhalten. Der Grund ist, dass es für den DPDA zwei verschiedene Gründe geben kann, warum er ein Wort w nicht akzeptiert:

i) wenn w ganz gelesen ist, so ist der Automat nicht in einem Endzustand
ii) das Wort w wird nicht ganz gelesen

Man muss den DPDA zunächst in einen DPDA umwandeln, bei dem der zweite Grund nie auftritt (siehe z. B. Ingo Wegener: Theoretische Informatik, S. 193-195, für Details).