

4.7 Kellerautomaten

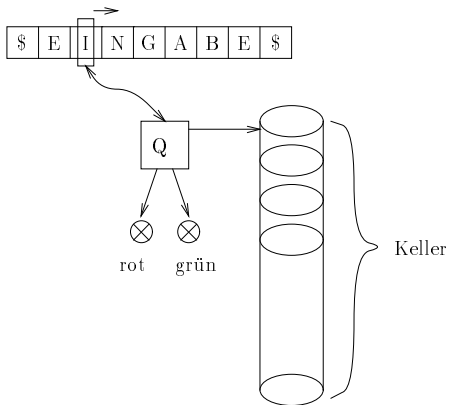
In der Literatur findet man häufig auch die Bezeichnungen **Stack-Automat** oder **Pushdown-Automat**. Kellerautomaten sind, wenn nichts anderes gesagt wird, **nichtdeterministisch**.

Definition 78

Ein NPDA = PDA (= Nichtdeterministischer Pushdown-Automat) besteht aus:

Q	endliche Zustandsmenge
Σ	endliches Eingabealphabet
Δ	endliches Stackalphabet
$q_0 \in Q$	Anfangszustand
$Z_0 \in \Delta$	Initialisierung des Stack
δ	Übergangsrelation Fkt. $Q \times (\Sigma \cup \{\epsilon\}) \times \Delta \rightarrow 2^{Q \times \Delta^*}$ wobei $ \delta(q, a, Z) + \delta(q, \epsilon, Z) < \infty \quad \forall q, a, Z$
$F \subseteq Q$	akzeptierende Zustände

Der Kellerautomat



Konfiguration:

Tupel (q, w, α) mit

$$\begin{aligned} q &\in Q \\ w &\in \Sigma^* \\ \alpha &\in \Delta^* \end{aligned}$$

Schritt:

$$(q, w_0 w', Z \alpha') \rightarrow (q', w', Z_1 \dots Z_r \alpha')$$

wenn $(q', Z_1 \dots Z_r) \in \delta(q, w_0, Z)$

bzw.:

$$(q, w, Z \alpha') \rightarrow (q', w, Z_1 \dots Z_r \alpha')$$

wenn $(q', Z_1 \dots Z_r) \in \delta(q, \epsilon, Z)$

Definition 79

- ① Ein NPDA A akzeptiert $w \in \Sigma^*$ durch leeren Stack, falls

$$(q_0, w, Z_0) \rightarrow^* (q, \epsilon, \epsilon) \text{ f\"ur ein } q \in Q .$$

- ② Ein NPDA A akzeptiert $w \in \Sigma^*$ durch akzeptierenden Zustand, falls

$$(q_0, w, Z_0) \rightarrow^* (q, \epsilon, \alpha) \text{ f\"ur ein } q \in F, \alpha \in \Delta^* .$$

- ③ Ein NPDA heit deterministisch (DPDA), falls

$$|\delta(q, a, Z)| + |\delta(q, \epsilon, Z)| \leq 1 \quad \forall (q, a, Z) \in Q \times \Sigma \times \Delta .$$

Beispiel 80

Der PDA mit

$$\begin{aligned}\delta(q_0, a, *) &= \{(q_0, a *)\} && \text{für } a \in \{0, 1\}, * \in \{0, 1, Z_0\} \\ \delta(q_0, \#, *) &= \{(q_1, *)\} && \text{für } * \in \{0, 1, Z_0\} \\ \delta(q_1, 0, 0) &= \{(q_1, \epsilon)\} \\ \delta(q_1, 1, 1) &= \{(q_1, \epsilon)\} \\ \delta(q_1, \epsilon, Z_0) &= \{(q_1, \epsilon)\} && \text{akzeptiert mit leerem Stack}\end{aligned}$$

die Sprache

$$L = \{w\#w^R; w \in \{0, 1\}^*\}.$$

Beispiel 80

Der PDA mit

$$\begin{aligned}\delta(q_0, a, *) &= \{(q_0, a *)\} && \text{für } a \in \{0, 1\}, * \in \{0, 1, Z_0\} \\ \delta(q_0, \#, *) &= \{(q_1, *)\} && \text{für } * \in \{0, 1, Z_0\} \\ \delta(q_1, 0, 0) &= \{(q_1, \epsilon)\} \\ \delta(q_1, 1, 1) &= \{(q_1, \epsilon)\} \\ \delta(q_1, \epsilon, Z_0) &= \{(q_a, \epsilon)\} && \text{akzeptiert mit akzeptierendem} \\ &&& \text{Zustand } (F = \{q_a\}) \\ &&& \text{(und leerem Stack)}\end{aligned}$$

die Sprache

$$L = \{w\#w^R; w \in \{0, 1\}^*\}.$$

Satz 81

Sei $A_1 = (Q, \Sigma, \Delta, q_0, Z_0, \delta, F)$ ein NPDA, der mit Endzustand akzeptiert. Dann kann in linearer Zeit ein NPDA $A_2 = (Q', \Sigma, \Delta', q'_0, Z'_0, \delta')$ konstruiert werden, der $L(A_1)$ mit leerem Stack akzeptiert.

Beweis:

A_2 simuliert A_1 . Sobald A_1 in einen akzeptierenden Endzustand gerät, rät A_2 , ob die Eingabe zu Ende gelesen ist. Falls A_2 dies meint, wird der Keller geleert.

Um zu verhindern, dass bei der Simulation von A_1 der Keller leer wird, ohne dass A_1 akzeptiert, führen wir ein neues Kellersymbol Z'_0 ein:

$$\begin{aligned}Q' &= Q \cup \{\bar{q}, q'_0\} \\ \Delta' &= \Delta \cup \{Z'_0\}\end{aligned}$$

und wir **erweitern** δ zu δ' gemäß

$$\delta'(q'_0, \epsilon, Z'_0) = \{(q_0, Z_0 Z'_0)\}$$

$$\delta'(q, \epsilon, Z) \cup = \{(\bar{q}, \epsilon)\} \quad \text{für } q \in F, Z \in \Delta'$$

$$\delta'(\bar{q}, \epsilon, Z) = \{(\bar{q}, \epsilon)\} \quad \text{für } Z \in \Delta'$$



Bemerkung:

Akzeptieren mit leerem Keller bedeutet, dass der NPDA akzeptiert, falls der Keller leer ist **und** die Eingabe gelesen ist.

Bemerkung:

Akzeptieren mit leerem Keller bedeutet, dass der NPDA akzeptiert, falls der Keller leer ist **und** die Eingabe gelesen ist, bzw. dass, falls der Keller leer ist, der NPDA **die bisher gelesene Eingabe** akzeptiert.

Satz 82

Sei $A_1 = (Q, \Sigma, \Delta, q_0, Z_0, \delta)$ ein NPDA, der mit leerem Keller akzeptiert. Dann kann in linearer Zeit ein NPDA $A_2 = (Q', \Sigma, \Delta', q'_0, Z'_0, \delta', F)$ konstruiert werden, welcher $L(A_1)$ mit akzeptierendem Endzustand akzeptiert.

Beweis:

A_2 simuliert A_1 . Am Anfang steht ein neues Kellersymbol auf dem Stack. Sobald bei der Simulation von A_1 dieses auf dem Stack vorgefunden wird, weiß man, dass A_1 seinen Stack leerräumt hat und folglich akzeptiert. Folglich geht A_2 in einen akzeptierenden Endzustand und hält:

$$Q' = Q \cup \{q'_0, q_f\}$$

$$\Delta' = \Delta \cup \{Z'_0\}$$

$$F = \{q_f\}$$

$$\delta'(q'_0, \epsilon, Z'_0) = \{(q_0, Z_0 Z'_0)\}$$

$$\delta'(q, a, Z) = \delta(q, a, Z) \text{ für } q \in Q, a \in \Sigma \cup \{\epsilon\}, Z \in \Delta$$

$$\delta'(q, \epsilon, Z'_0) = \{(q_f, Z'_0)\} \text{ für } q \in Q$$



4.8 Kellerautomaten und kontextfreie Sprachen

Satz 83

Sei G eine CFG in Greibach-Normalform. Dann kann in linearer Zeit ein NPDA A konstruiert werden (welcher mit leerem Stack akzeptiert), so dass

$$L(A) = L(G) .$$

Beweis:

Sei o.B.d.A. $\epsilon \notin L(G)$.

Der Automat startet mit S auf dem Stack. Er sieht sich in jedem Schritt das oberste Stacksymbol A an und überprüft, ob es in G eine Produktion gibt, deren linke Seite A ist und deren rechte Seite mit dem Terminal beginnt, welches unter dem Lesekopf steht.

Sei also $G = (V, T, P, S)$.

Konstruiere NPDA $A = (Q, \Sigma, \Delta, q_0, Z_0, \delta)$ mit

$$Q := \{q_0\}$$

$$\Delta := V$$

$$\Sigma := T$$

$$Z_0 := S$$

$$\delta(q_0, a, A) \ni (q_0, \alpha) \quad \text{für } (A \rightarrow a\alpha) \in P .$$

Beweis (Forts.):

Zu zeigen ist nun: $L(A) = L(G)$.

Hilfsbehauptung:

$S \rightarrow_G^* w_1 \dots w_i A_1 \dots A_m$ mit $w_j \in T, A_j \in V$ per Linksableitung

$$\Leftrightarrow (q_0, w_1 \dots w_i, Z_0) \rightarrow_A^* (q_0, \epsilon, A_1 \dots A_m)$$

Der Beweis erfolgt durch Induktion über die Anzahl der Schritte in der Linksableitung.

Beweis (Forts.):

Induktionsanfang ($i = 0$):

$$S \rightarrow_G^* S \quad \Leftrightarrow \quad (q_0, \epsilon, Z_0) \rightarrow_A^* (q_0, \epsilon, Z_0)$$

Beweis (Forts.):

Induktionsschritt $((i - 1) \mapsto i)$:

$$\begin{aligned} S &\rightarrow_G^* w_1 \dots w_i A_1 \dots A_m \\ \Leftrightarrow S &\rightarrow_G^* w_1 \dots w_{i-1} A' A_v \dots A_m \quad v \in \{1, \dots, m + 1\} \\ &\rightarrow_G w_1 \dots w_i A_1 \dots A_m \\ &\text{(also } (A' \rightarrow w_i A_1 \dots A_{v-1}) \in P) \end{aligned}$$

gemäß Induktionsvoraussetzung

$$\begin{aligned} \Leftrightarrow (q_0, w_1 \dots w_{i-1}, Z_0) &\rightarrow_A^* (q_0, \epsilon, A' A_v \dots A_m) \\ \Leftrightarrow (q_0, w_1 \dots w_{i-1} w_i, Z_0) &\rightarrow_A^* (q_0, w_i, A' A_v \dots A_m) \\ &\rightarrow_A (q_0, \epsilon, A_1 \dots A_m) \\ &\text{da } (A' \rightarrow w_i A_1 \dots A_{v-1}) \in P) \\ \Leftrightarrow (q_0, w_1 \dots w_i, Z_0) &\rightarrow_A^* (q_0, \epsilon, A_1 \dots A_m) \end{aligned}$$

Beweis (Forts.):

Aus der Hilfsbehauptung folgt

$$L(A) = L(G) .$$



Satz 84

Sei $A = (Q, \Sigma, \Delta, q_0, Z_0, \delta)$ ein NPDA, der mit leerem Keller akzeptiert. Dann ist $L(A)$ kontextfrei.

Beweis:

Wir definieren:

$$G = (V, T, P, S)$$

$$T := \Sigma$$

$$V := Q \times \Delta \times Q \cup \{S\} \quad \text{wobei wir die Tupel mit } [, ,] \text{ notieren}$$

$$P \ni S \rightarrow [q_0, Z_0, q] \text{ f\"ur } q \in Q$$

$$P \ni [q, Z, q_m] \rightarrow a[p, Z_1, q_1][q_1, Z_2, q_2] \cdots [q_{m-1}, Z_m, q_m]$$

$$\text{f\"ur } \delta(q, a, Z) \ni (p, Z_1 \cdots Z_m), \forall q_1, \dots, q_m \in Q,$$

$$\text{mit } a \in \Sigma \cup \{\epsilon\}.$$

Idee: Aus $[p, X, q]$ sollen sich alle die W\"orтер ableiten lassen, die der NPDA A lesen kann, wenn er im Zustand p mit (lediglich) X auf dem Stack startet und im Zustand q mit leerem Stack endet.

Beweis (Forts.):

Hilfsbehauptung:

$$[p, X, q] \rightarrow_G^* w \Leftrightarrow (p, w, X) \rightarrow_A^* (q, \epsilon, \epsilon).$$

„ \Rightarrow “: Induktion über die Länge l der Ableitung.

Induktionsanfang ($l = 1$):

$$\begin{aligned} & [p, X, q] \rightarrow_G w \\ \Rightarrow & \delta(p, w, X) \ni (q, \epsilon) \\ \Rightarrow & (p, w, X) \rightarrow_A (q, \epsilon, \epsilon) \end{aligned}$$

Beweis (Forts.):

Induktionsschritt $((l - 1) \mapsto l)$:

Gelte

$$\begin{aligned} [p, X, q_{m+1}] &\rightarrow_G a[q_1, X_1, q_2][q_2, X_2, q_3] \cdots [q_m, X_m, q_{m+1}] \\ &\rightarrow_G^* aw^{(1)} \cdots w^{(m)} = w \end{aligned}$$

mit $(q_1, X_1 \cdots X_m) \in \delta(p, a, X)$, $[q_i, X_i, q_{i+1}] \rightarrow_G^{l_i} w^{(i)}$ und $\sum l_i < l$.

Dann gilt gemäß Induktionsvoraussetzung

$$\begin{aligned} \Rightarrow & (q_i, w^{(i)}, X_i) \rightarrow_A^{l_i} (q_{i+1}, \epsilon, \epsilon) \quad \forall i \in \{1, \dots, m\} \\ \Rightarrow & (q, \underbrace{aw^{(1)} \cdots w^{(m)}}_{=w}, X) \rightarrow_A (q_1, w^{(1)} \cdots w^{(m)}, X_1 \cdots X_m) \\ & \rightarrow_A^{<l} (q_{m+1}, \epsilon, \epsilon) . \end{aligned}$$

Beweis (Forts.):

„ \Leftarrow “: Induktion über die Länge l einer Rechnung des NPDA's A

Induktionsanfang ($l = 1$):

$$\begin{aligned} & (p, w, X) \rightarrow_A (q, \epsilon, \epsilon) \\ \Rightarrow & (q, \epsilon) \in \delta(p, w, X) \quad (\text{also } |w| \leq 1) \\ \Rightarrow & ([p, X, q] \rightarrow w) \in P. \end{aligned}$$

Beweis (Forts.):

Induktionsschritt $((l - 1) \mapsto l)$:

Sei

$$\begin{aligned}(p, w, X) &\rightarrow_A (q_1, w', X_1 \cdots X_m) \\ &\rightarrow_A^* (q, \epsilon, \epsilon)\end{aligned}$$

eine Rechnung von A der Länge l , mit $w = ew'$ und $e \in \Sigma \cup \{\epsilon\}$.

Nach Definition gibt es

$$([p, X, q] \rightarrow e[q_1 X_1 q_2] \cdots [q_m X_m q_{m+1}]) \in P \quad \text{mit } q_{m+1} = q$$

und eine Zerlegung $w' = w^{(1)} \cdots w^{(m)}$, so dass $w^{(1)} \cdots w^{(i)}$ der von A zu dem Zeitpunkt verarbeitete Teilstring von w' ist, wenn X_{i+1} zum ersten Mal oberstes Stacksymbol (bzw., für $i = m$, der Stack leer) wird.

Beweis (Forts.):

Gemäß Induktionsvoraussetzung gilt also

$$(q_i, w^{(i)}, X_i) \rightarrow_A^{l_i} (q_{i+1}, \epsilon, \epsilon) \quad \text{mit } \sum l_i < l \text{ und} \\ [q_i, X_i, q_{i+1}] \rightarrow_G^* w^{(i)} .$$

Also folgt:

$$[p, X, q] \rightarrow_G e[q_1, X_1, q_2] \cdots [q_m, X_m, q_{m+1}] \quad \text{mit } q_{m+1} = q \\ \rightarrow_G^{\leq l} ew^{(1)} \cdots w^{(m)} = w$$

Aus der Hilfsbehauptung folgt der Satz. □

Satz 85

Folgende Aussagen sind äquivalent:

- L wird von einer *kontextfreien Grammatik* erzeugt.
- L wird von einem *NPDA* akzeptiert.

Beweis:

Folgt aus den vorhergehenden Sätzen. □

Bemerkung

Für die Praxis (z.B. Syntaxanalyse von Programmen) sind polynomiale Algorithmen wie CYK noch zu langsam. Für Teilklassen von CFLs sind schnellere Algorithmen bekannt, z.B.



Jay Earley:

An Efficient Context-free Parsing Algorithm.

Communications of the ACM, **13**(2):94–102, 1970

4.9 Deterministische Kellerautomaten

Wir haben bereits definiert:

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

$$|\delta(q, a, Z)| + |\delta(q, \epsilon, Z)| \leq 1 \quad \forall (q, a, Z) \in Q \times \Sigma \times \Delta .$$

Die von einem DPDA, der mit **leerem Keller akzeptiert**, erkannte Sprache genügt der **Fano-Bedingung**, d.h. kein Wort in der Sprache ist echtes Präfix eines anderen Wortes in der Sprache.

Festlegung:

Da wir an einem weniger eingeschränkten Maschinenmodell interessiert sind, legen wir fest, dass ein DPDA stets mit **akzeptierenden Zuständen** akzeptiert.

Definition 86

Ein DPDA ist in **Normalform**, falls gilt:

- 1 $(q', \alpha) = \delta(q, e, X)$ für $e \in \Sigma \cup \{\epsilon\}$, $q, q' \in Q$, $X \in \Delta$
 $\Rightarrow \alpha \in \{\epsilon, X, YX\}$ für $Y \in \Delta$.
- 2 Der Automat liest jede Eingabe vollständig.

Satz 87

Zu jedem DPDA $A = (Q, \Sigma, \Delta, q_0, Z_0, \delta, F)$ kann ein äquivalenter DPDA $A' = (Q', \Sigma, \Delta', q'_0, Z'_0, \delta', F')$ in Normalform konstruiert werden.

Beweis:

Erste Schritte der Konstruktion:

- 1 Werden von A in einem Schritt mehr als zwei Symbole auf dem Stack abgelegt, wird dies von A' durch eine Folge von Schritten mit je 2 Stacksymbolen ersetzt.
- 2 Werden zwei oder ein Stacksymbol abgelegt und dabei das oberste Stacksymbol X geändert, entfernen wir zunächst in einem eigenen Schritt das oberste Stacksymbol und pushen dann die gewünschten Symbole. (Das „Merken“ erfolgt in der Zustandsmenge Q' .)
- 3 Wir vervollständigen δ' mittels eines Fangzustandes. Es könnte hier noch sein, dass der DPDA eine Schleife aus lauter ϵ -Übergängen ausführt.

Beweis (Forts.):

Hilfsbehauptung:

Der DPDA A führt ab einer bestimmten Konfiguration (q, ϵ, β) unendlich viele direkt aufeinander folgende ϵ -Übergänge genau dann aus, wenn

$$\begin{array}{lll} (q, \epsilon, \beta) & \rightarrow^* & (q', \epsilon, X\beta') \quad \text{und} \\ (q', \epsilon, X) & \rightarrow^+ & (q', \epsilon, X\alpha) \quad \text{für } q, q' \in Q \\ & & X \in \Delta, \alpha, \beta, \beta' \in \Delta^* \end{array}$$

„ \Leftarrow “: klar

Beweis (Forts.):

„ \Rightarrow “: Betrachte eine unendlich lange Folge von ϵ -Übergängen.

Sei $n := |Q| \cdot |\Delta| + |\beta| + 1$.

Wird die Stackhöhe n nie erreicht, so muss sich sogar eine Konfiguration des DPDA's wiederholen. Daraus folgt die Behauptung.

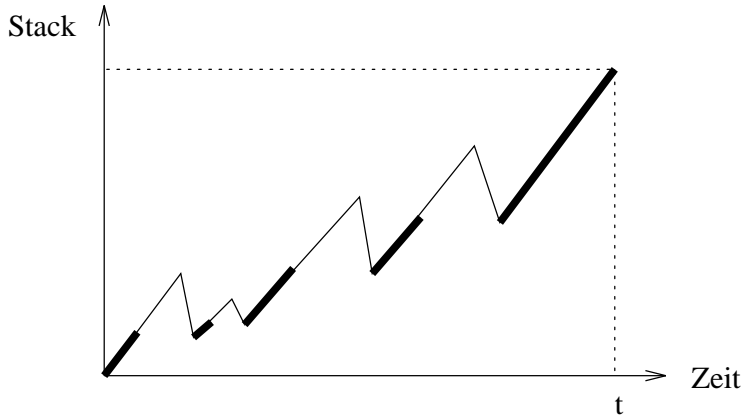
Beweis (Forts.):

Ansonsten wird jede Stackhöhe $|\beta|, \dots, n$ mindestens einmal erreicht (wegen der Normalform ist die Höhendifferenz pro Schritt $\in \{-1, 0, 1\}$).

Betrachte den Zeitpunkt t , in dem die Stackhöhe zum erstenmal n ist. Markiere für jedes $i \in \{|\beta|, \dots, n\}$ den Zeitpunkt t_i , wo zum letzten Mal (vor Zeitpunkt t) die Stackhöhe $= i$ ist. Zu diesen Zeitpunkten t_i betrachte die Paare $(q, X) \in Q \times \Delta$, wobei q der Zustand des DPDA's und X das oberste Kellersymbol des DPDA's zu diesem Zeitpunkt ist.

Da es mehr als $|\Delta| \cdot |Q|$ markierte Paare gibt, taucht ein markiertes Paar (q', X) doppelt auf. Für dieses gilt dann $(q', \epsilon, X) \rightarrow^+ (q', \epsilon, X\alpha)$.

Beweis (Forts.):



Beweis (Forts.):

Das gleiche Argument gilt, falls sich die Stackhöhe um $> |Q| \cdot |\Delta|$ erhöht.

Damit lassen sich alle Paare (q', X) finden, für die gilt:

$$(q', \epsilon, X) \rightarrow^+ (q', \epsilon, X\alpha), \alpha \in \Delta^*.$$

Da der DPDA nicht endlos weiterlaufen soll, ersetzen wir $\delta(q', \epsilon, X)$ durch einen ϵ -Übergang in einen neuen Zustand q'' (der genau dann akzeptierend ist, wenn in der Schleife $(q', \epsilon, X) \rightarrow^+ (q', \epsilon, X\alpha)$ ein akzeptierender Zustand auftritt) und einen ϵ -Übergang von q'' in den nichtakzeptierenden Fangzustand. □

Satz 88

*Die Klasse der deterministischen kontextfreien Sprachen (also der von DPDA's **erkannten** Sprachen) [DCFL] ist unter Komplement abgeschlossen.*

Beweis:

Folgt aus der obigen Konstruktion!

