SS 2005

Einführung in die Informatik IV

Ernst W. Mayr

Fakultät für Informatik
TU München

http://www14.in.tum.de/lehre/2005SS/info4/index.html.de

20. Mai 2005

Korollar 79

Sei G eine kontextfreie Grammatik. Es gibt einen Algorithmus, der eine zu G äquivalente Grammatik in Greibach-Normalform konstruiert, deren rechte Seiten jeweils höchstens zwei Variablen enthalten.

Beweis:

Klar! $\square$

Korollar 79

Sei G eine kontextfreie Grammatik. Es gibt einen Algorithmus, der eine zu G äquivalente Grammatik in Greibach-Normalform konstruiert, deren rechte Seiten jeweils höchstens zwei Variablen enthalten.

Beweis:

Klar! $\square$

4.7 Kellerautomaten

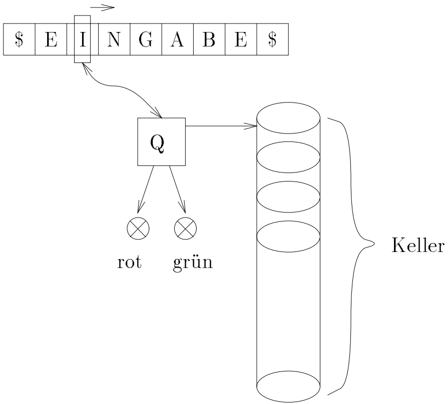
In der Literatur findet man häufig auch die Bezeichnungen Stack-Automat oder Pushdown-Automat. Kellerautomaten sind, wenn nichts anderes gesagt wird, nichtdeterministisch.

Definition 80

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. $Q \times (\Sigma \cup \{\epsilon\}) \times \Delta \to 2^{Q \times \Delta^*}$ 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

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

Konfiguration:

Tupel (q, w, α) mit

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

Schritt:

$(q, w_0 w', Z\alpha') \rightarrow (q', w', Z_1 \dots Z_r \alpha')$
gdw $\quad (q', Z_1 \dots Z_r) \in \delta(q, w_0, Z)$
bzw.:
$(q, w, Z\alpha') \rightarrow (q', w, Z_1 \dots Z_r \alpha')$
gdw $\quad (q', Z_1 \dots Z_r) \in \delta(q, \epsilon, Z)$

Definition 81

1. Ein NPDA A akzeptiert $w \in \Sigma^*$ durch leeren Stack, falls
$$(q_0, w, Z_0) \rightarrow^* (q, \epsilon, \epsilon) \text{ für ein } q \in Q \; .$$
2. Ein NPDA A akzeptiert $w \in \Sigma^*$ durch akzeptierenden Zustand, falls
$$(q_0, w, Z_0) \rightarrow^* (q, \epsilon, \alpha) \text{ für ein } q \in F, \alpha \in \Delta^* \; .$$
3. Ein NPDA heißt deterministisch (DPDA), falls
$$|\delta(q, a, Z)| + |\delta(q, \epsilon, Z)| \leq 1 \qquad \forall (q, a, Z) \in Q \times \Sigma \times \Delta \; .$$

Definition 81

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

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

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

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

3. Ein NPDA heißt deterministisch (DPDA), falls

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

Definition 81

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

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

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

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

3. Ein NPDA heißt deterministisch (DPDA), falls

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

Definition 81

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

$$(q_0, w, Z_0) \to^* (q, \epsilon, \epsilon) \text{ für ein } q \in Q\ .$$

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

$$(q_0, w, Z_0) \to^* (q, \epsilon, \alpha) \text{ für ein } q \in F,\ \alpha \in \Delta^*\ .$$

3. Ein NPDA heißt deterministisch (DPDA), falls

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

Beispiel 82

Der PDA mit

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

die Sprache

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

Beispiel 82

Der PDA mit

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

die Sprache

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

Satz 83

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

$$\begin{array}{ll} \delta'(q'_0, \epsilon, Z'_0) = \{(q_0, Z_0 Z'_0)\} & \\ \delta'(q, \epsilon, Z) \cup= \{(\bar{q}, \epsilon)\} & \text{für } q \in F,\ Z \in \Delta' \\ \delta'(\bar{q}, \epsilon, Z) = \{(\bar{q}, \epsilon)\} & \text{für } Z \in \Delta' \end{array}$$

$\square$

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

$$\begin{aligned} &\delta'(q_0', \epsilon, Z_0') = \{(q_0, Z_0 Z_0')\} && \\ &\delta'(q, \epsilon, Z) \mathrel{\cup=} \{(\bar{q}, \epsilon)\} && \text{für } q \in F,\ Z \in \Delta' \\ &\delta'(\bar{q}, \epsilon, Z) = \{(\bar{q}, \epsilon)\} && \text{für } Z \in \Delta' \end{aligned}$$

$\square$

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

$$\begin{aligned} &\delta'(q_0', \epsilon, Z_0') = \{(q_0, Z_0 Z_0')\} && \\ &\delta'(q, \epsilon, Z) \cup= \{(\bar{q}, \epsilon)\} && \text{für } q \in F,\ Z \in \Delta' \\ &\delta'(\bar{q}, \epsilon, Z) = \{(\bar{q}, \epsilon)\} && \text{für } Z \in \Delta' \end{aligned}$$

$\square$

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 84

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 leergeräumt hat und folglich akzeptiert. Folglich geht A_2 in einen akzeptierenden Endzustand und hält:

$$\begin{aligned} Q' &= Q \cup \{q_0', q_f\} \\ \Delta' &= \Delta \cup \{Z_0'\} \\ F &= \{q_f\} \end{aligned}$$

$$\begin{aligned} \delta'(q_0', \epsilon, Z_0') &= \{(q_0, Z_0 Z_0')\} \\ \Delta'(q, a, Z) &= \delta(q, a, Z) \text{ für } q \in Q,\ a \in \Sigma \cup \{\epsilon\},\ Z \in \Delta \\ \delta'(q, \epsilon, Z_0') &= \{(q_f, Z_0')\} \text{ für } q \in Q \end{aligned}$$

$\square$

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 leergeräumt hat und folglich akzeptiert. Folglich geht A_2 in einen akzeptierenden Endzustand und hält:

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

$$\begin{aligned} \delta'(q'_0, \epsilon, Z'_0) &= \{(q_0, Z_0 Z'_0)\} \\ \Delta'(q, a, Z) &= \delta(q, a, Z) \text{ für } q \in Q,\ a \in \Sigma \cup \{\epsilon\},\ Z \in \Delta \\ \delta'(q, \epsilon, Z'_0) &= \{(q_f, Z'_0)\} \text{ für } q \in Q \end{aligned}$$

$\square$

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 leergeräumt hat und folglich akzeptiert. Folglich geht A_2 in einen akzeptierenden Endzustand und hält:

$$
\begin{array}{lcl}
Q' & = & Q \cup \{q_0', q_f\} \\
\Delta' & = & \Delta \cup \{Z_0'\} \\
F & = & \{q_f\}
\end{array}
$$

$$
\begin{array}{lcl}
\delta'(q_0', \epsilon, Z_0') & = & \{(q_0, Z_0 Z_0')\} \\
\Delta'(q, a, Z) & = & \delta(q, a, Z) \text{ für } q \in Q,\ a \in \Sigma \cup \{\epsilon\},\ Z \in \Delta \\
\delta'(q, \epsilon, Z_0') & = & \{(q_f, Z_0')\} \text{ für } q \in Q
\end{array}
$$

□

Satz 85

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.

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

$$
\begin{aligned}
Q &:= \{q_0\} \\
\Delta &:= V \\
\Sigma &:= T \\
Z_0 &:= S
\end{aligned}
$$

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

Beweis:

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

Beweis:

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

Hilfsbehauptung:

$$S \rightarrow^*_G w_1 \dots w_i A_1 \dots A_m \text{ mit } w_j \in T, A_j \in V \text{ per Linksableitung}$$

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

Beweis:

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

Hilfsbehauptung:

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

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

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

Beweis:

Induktionsanfang ($i = 0$):

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

Beweis:

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

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

Beweis:

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

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

gemäß Induktionsvoraussetzung

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

Beweis:

Aus der Hilfsbehauptung folgt

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

$\square$