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

23. Mai 2005

Satz 86

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:

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

$$T := \Sigma$$

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

$$P \ni S \to [q_0, Z_0, q] \text{ für } q \in Q$$

Beweis:

Wir definieren:

$$
\begin{aligned}
&G = (V, T, P, S) \\
&T := \Sigma \\
&V := Q \times \Delta \times Q \cup \{S\} \qquad \text{wobei wir die Tupel mit } [\,,\,,\,] \text{ notieren} \\
&P \ni S \to [q_0, Z_0, q] \text{ für } q \in Q \\
&P \ni [q, Z, q_m] \to a[p, Z_1, q_1][q_1, Z_2, q_2] \cdots [q_{m-1}, Z_m, q_m] \\
&\qquad \text{für } \delta(q, a, Z) \ni (p, Z_1 \cdots Z_m), \forall q_1, \ldots, q_m \in Q, \\
&\qquad\qquad \text{mit } a \in \Sigma \cup \{\epsilon\}\,.
\end{aligned}
$$

Beweis:

Wir definieren:

$$
\begin{aligned}
&G = (V, T, P, S)\\
&T := \Sigma\\
&V := Q \times \Delta \times Q \cup \{S\} \qquad \text{wobei wir die Tupel mit } [,,] \text{ notieren}\\
&P \ni S \rightarrow [q_0, Z_0, q] \text{ für } q \in Q\\
&P \ni [q, Z, q_m] \rightarrow a[p, Z_1, q_1][q_1, Z_2, q_2] \cdots [q_{m-1}, Z_m, q_m]\\
&\qquad\qquad \text{für } \delta(q, a, Z) \ni (p, Z_1 \cdots Z_m), \forall q_1, \ldots, q_m \in Q,\\
&\qquad\qquad\qquad \text{mit } a \in \Sigma \cup \{\epsilon\}\,.
\end{aligned}
$$

Idee: Aus $[p, X, q]$ sollen sich alle die Wörter 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:

Hilfsbehauptung:

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

Beweis:

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 ($l = 1$):

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

Beweis:

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

Gelte

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

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 $\sum l_i < l$.

Beweis:

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

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

Beweis:

„$\Leftarrow$": Induktion über die Länge l einer Rechnung des NPDAs A

Induktionsanfang ($l = 1$):

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

Beweis:

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

$$(p, w, X) \rightarrow_A (q_1, w', X_1 \cdots X_m)$$
$$\rightarrow_A^* (q, \epsilon, \epsilon)$$

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

Beweis:

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

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

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

Nach Definition gibt es

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

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:

Gemäß Induktionsvoraussetzung gilt also

$$(q_i, w^{(i)}, X_i) \to_A^{l_i} (q_{i+1}, \epsilon, \epsilon) \qquad \text{mit } \sum l_i < l \text{ und}$$

$$[q_i, X_i, q_{i+1}] \to_G^* w^{(i)} \ .$$

Beweis:

Gemäß Induktionsvoraussetzung gilt also

$$(q_i, w^{(i)}, X_i) \to_A^{l_i} (q_{i+1}, \epsilon, \epsilon) \qquad \text{mit } \sum l_i < l \text{ und}$$

$$[q_i, X_i, q_{i+1}] \to_G^* w^{(i)} \ .$$

Also folgt:

$$\begin{aligned}[p, X, q] &\to_G e[q_1, X_1, q_2] \cdots [q_m, X_m, q_{m+1}] \quad \text{mit } q_{m+1} = q \\ &\to_G^{<l} ew^{(1)} \cdots w^{(m)} = w\end{aligned}$$

Beweis:

Gemäß Induktionsvoraussetzung gilt also

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

Also folgt:

$$\begin{aligned}[p, X, q] &\to_G e[q_1, X_1, q_2] \cdots [q_m, X_m, q_{m+1}] \quad \text{mit } q_{m+1} = q \\ &\to_G^{<l} ew^{(1)} \cdots w^{(m)} = w\end{aligned}$$

Aus der Hilfsbehauptung folgt der Satz. □

Satz 87

Folgende Aussagen sind äquivalent:

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

Beweis:

Folgt aus den vorhergehenden Sätzen. □

Satz 87

Folgende Aussagen sind äquivalent:

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

Beweis:

Folgt aus den vorhergehenden Sätzen. □

Satz 87

Folgende Aussagen sind äquivalent:

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

Beweis:

Folgt aus den vorhergehenden Sätzen. □

Satz 87

Folgende Aussagen sind äquivalent:

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

Beweis:

Folgt aus den vorhergehenden Sätzen. $\square$

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

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

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.

4.9 Deterministische Kellerautomaten

Wir haben bereits definiert:
Ein PDA heißt deterministisch (DPDA), falls

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

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.

4.9 Deterministische Kellerautomaten

Wir haben bereits definiert:
Ein PDA heißt deterministisch (DPDA), falls

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

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 88

Ein DPDA ist in Normalform, falls gilt:

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

Satz 89

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.

Definition 88

Ein DPDA ist in Normalform, falls gilt:

1. $(q', \alpha) = \delta(q, e, X)$ für $e \in \Sigma \cup \{\epsilon\}$, $q, q' \in Q$, $X \in \Delta$

$$\Rightarrow \alpha \in \{\epsilon, X, YX\} \text{ für } Y \in \Delta\ .$$

2. Der Automat liest jede Eingabe vollständig.

Satz 89

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.

Definition 88

Ein DPDA ist in Normalform, falls gilt:

1. $(q', \alpha) = \delta(q, e, X)$ für $e \in \Sigma \cup \{\epsilon\}$, $q, q' \in Q$, $X \in \Delta$

$$\Rightarrow \alpha \in \{\epsilon, X, YX\} \text{ für } Y \in \Delta\,.$$

2. Der Automat liest jede Eingabe vollständig.

Satz 89

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.

Definition 88

Ein DPDA ist in Normalform, falls gilt:

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

Satz 89

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:

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:

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:

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:

Hilfsbehauptung:

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

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

Beweis:

Hilfsbehauptung:

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

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

„$\Leftarrow$“: klar

Beweis:

„$\Rightarrow$“: Betrachte eine unendlich lange Folge von ϵ-Übergängen.
Sei $n := |Q| \cdot |\Delta| + |\beta| + 1$.

Beweis:

„$\Rightarrow$“: 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.

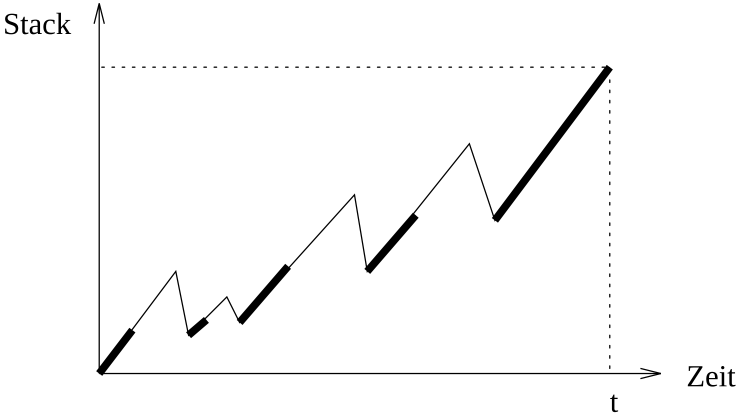
Beweis:

„$\Rightarrow$“: Betrachte eine unendlich lange Folge von ϵ-Übergängen.
Sei $n := |Q| \cdot |\Delta| + |\beta| + 1$.

Ansonsten wird jede Stackhöhe $|\beta|, \ldots, n$ 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 $i \in \{|\beta|, \ldots, n\}$ den Zeitpunkt t_i, wo zum letzen 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:

„$\Rightarrow$“: Betrachte eine unendlich lange Folge von ϵ-Übergängen.
Sei $n := |Q| \cdot |\Delta| + |\beta| + 1$.



Beweis:

„$\Rightarrow$“: Betrachte eine unendlich lange Folge von ϵ-Übergängen.
Sei $n := |Q| \cdot |\Delta| + |\beta| + 1$.

Ansonsten wird jede Stackhöhe $|\beta|, \ldots, n$ 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 $i \in \{|\beta|, \ldots, n\}$ den Zeitpunkt t_i, wo zum letzen 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:

„$\Rightarrow$“: Betrachte eine unendlich lange Folge von ϵ-Übergängen.
Sei $n := |Q| \cdot |\Delta| + |\beta| + 1$.

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

Beweis:

„$\Rightarrow$“: Betrachte eine unendlich lange Folge von ϵ-Übergängen.
Sei $n := |Q| \cdot |\Delta| + |\beta| + 1$.

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

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

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 $(q', \epsilon, X) \to^{+} (q', \epsilon, X\alpha)$ ein akzeptierender Zustand auftritt) und einen ϵ-Übergang von q'' in den nichtakzeptierenden Fangzustand. $\square$

Satz 90

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

Beweis:

Folgt aus der obigen Konstruktion! □

Satz 90

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

Beweis:

Folgt aus der obigen Konstruktion! □