

Vorlesung

Grundlagen der Theoretischen Informatik / Einführung in die Theoretische Informatik I

Bernhard Beckert

Institut für Informatik



Sommersemester 2007

Diese Vorlesungsmaterialien basieren ganz wesentlich auf den Folien zu den Vorlesungen von

Katrin Erk (gehalten an der Universität Koblenz-Landau)

Jürgen Dix (gehalten an der TU Clausthal)

Ihnen beiden gilt mein herzlicher Dank.

– Bernhard Beckert, April 2007

Beispiel 22.9

Die Sprache

$$L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$$

wird über finalen Zustand akzeptiert von dem PDA

$$\mathcal{M} = (\{s_0, s_1\}, \{a, b\}, \{Z_0, \underline{A}, A, \underline{B}, B\}, \Delta, s_0, Z_0, \{s_0\})$$

mit ...

Beispiel (Forts.)

Idee:

- auf dem Stack mitzählen, wieviel A -Überhang oder B -Überhang momentan besteht
- Der Stack enthält zu jedem Zeitpunkt
 - entweder nur A/\underline{A} (A -Überhang)
 - oder nur B/\underline{B} (B -Überhang)
 - oder nur das Symbol Z_0 (Gleichstand).
- Das unterste A bzw. B auf dem Stack ist durch einen Unterstrich gekennzeichnet.
So weiß \mathcal{M} , wenn er dies Stacksymbol löscht, daß dann bis zu diesem Moment gleichviel a s wie b s gelesen wurden.

Beispiel (Forts.)

 $(s_0, a, Z_0) \Delta (s_1, \underline{A})$ $(s_1, a, \underline{A}) \Delta (s_1, \underline{AA})$ $(s_1, a, A) \Delta (s_1, AA)$ $(s_1, a, \underline{B}) \Delta (s_0, Z_0)$ $(s_1, a, B) \Delta (s_1, \varepsilon)$ $(s_0, b, Z_0) \Delta (s_1, \underline{B})$ $(s_1, b, \underline{B}) \Delta (s_1, \underline{BB})$ $(s_1, b, B) \Delta (s_1, BB)$ $(s_1, b, \underline{A}) \Delta (s_0, Z_0)$ $(s_1, b, A) \Delta (s_1, \varepsilon)$

Theorem 22.10 (finale Zustände \rightarrow leerer Keller)

Zu jedem PDA \mathcal{M}_1 existiert ein PDA \mathcal{M}_2 mit

$$L_f(\mathcal{M}_1) = L_l(\mathcal{M}_2)$$

Beweisidee

- Wir simulieren die Maschine \mathcal{M}_1 , die über finale Zustände akzeptiert, durch die Maschine \mathcal{M}_2 , die über leeren Keller akzeptiert.
- \mathcal{M}_2 arbeitet wie \mathcal{M}_1 , mit dem Unterschied:
Wenn ein Zustand erreicht wird, der in \mathcal{M}_1 final war, kann \mathcal{M}_2 seinen Keller leeren.

Theorem 22.11 (leerer Keller \rightarrow finale Zustände)

Zu jedem PDA \mathcal{M}_1 existiert ein PDA \mathcal{M}_2 mit

$$L_f(\mathcal{M}_1) = L_f(\mathcal{M}_2)$$

Beweisidee

- Wir simulieren die Maschine \mathcal{M}_1 , die über leeren Keller akzeptiert, durch die Maschine \mathcal{M}_2 , die über finale Zustände akzeptiert.
- \mathcal{M}_2 arbeitet wie \mathcal{M}_1 ,
legt aber ein zusätzliches Symbol ganz unten in den Keller.
Wenn \mathcal{M}_1 seinen Keller geleert hätte (also das neue unterste Symbol sichtbar wird),
kann \mathcal{M}_2 in einen finalen Zustand gehen.

Theorem 22.12 (PDA akzeptieren L_2)

Die Klasse der PDA-akzeptierten Sprachen ist L_2 .

Beweis

Dazu beweisen wir die folgenden zwei Lemmata, die zusammen die Aussage des Satzes ergeben.

Lemma 22.13 (cf-Grammatik \rightarrow PDA)

Zu jeder kontextfreien Grammatik G gibt es einen PDA \mathcal{M} mit

$$L(\mathcal{M}) = L(G)$$

Beweis

O.B.d.A. sei die kontextfreie Grammatik $G = (V, T, R, S)$ in Greibach-Normalform: Alle Grammatikregeln haben die Form

$$A \rightarrow au \quad \text{mit } A \in V, a \in T, u \in V^*$$

Wir konstruieren zu G einen PDA \mathcal{M} , der $L(G)$ akzeptiert.

Beweis (Forts.)

Idee: Der Automat \mathcal{M}

- vollzieht die Grammatikregeln nach, die angewendet worden sein könnten, um das aktuelle Eingabewort zu erzeugen und
- merkt sich das aktuelle Wort in der Ableitung bzw. dessen Rest
- merkt sich auf dem Keller alle Variablen, die im gedachten Ableitungswort noch vorkommen und noch ersetzt werden müssen.
- Die linkeste Variable liegt zuoberst: \mathcal{M} arbeitet mit der Linksableitung.

Beweis (Forts.)

Genauer:

- Erzeugung eines Wortes mit G beginnt beim Startsymbol S .
Deshalb S bei \mathcal{M} in Startkonfiguration oben auf dem Keller.
- Angenommen, G hat 2 Regeln mit S auf der linken Seite:
 $S \rightarrow aA_1A_2$ und $S \rightarrow bB_1B_2$
Angenommen, der erste Buchstabe des Input-Wortes w ist ein a .

Wenn w von G erzeugt wurde, hat G die erste der zwei S -Produktionen angewendet.

Entsprechend: Der Automat \mathcal{M} schiebt A_1A_2 auf den Stack.

Beweis (Forts.)

Genauer:

- Der zweite Buchstabe des Eingabeworts muss durch Anwendung einer Regel $A_1 \rightarrow a_1 \alpha$ erzeugt worden sein.
Angenommen, der zweite Buchstabe des Eingabeworts ist a_1 . Dann müssen die nächsten Buchstaben des Wortes aus den Variablen in α entstehen.
Der Automat entfernt A_1 vom Stack und legt α auf den Stack.
- Wenn es zwei Regeln $A_1 \rightarrow a_1 \alpha_1$ und $A_1 \rightarrow a_1 \alpha_2$ gibt, dann wählt \mathcal{M} indeterminiert eine der Regeln aus.
- Der PDA hat nur einen einzigen Zustand und akzeptiert über den leeren Keller.

Beweis (Forts.)

Formal:

$$\mathcal{M} = (K, \Sigma, \Gamma, \Delta, s_0, Z_0, F)$$

mit

$$K := \{s_0\}$$

$$\Sigma := T$$

$$\Gamma := V$$

$$Z_0 := S$$

$$F := \emptyset$$

$$\Delta := \{((s_0, a, A), (s_0, \alpha)) \mid A \rightarrow a\alpha \in R\}$$

Beweis (Forts.)

Damit gilt (Beweis s. Buch):

Es gibt eine Linksableitung $S \xRightarrow{*}_G x\alpha$ mit $x \in T^*, \alpha \in V^*$

gdw
 \mathcal{M} rechnet $(s_0, x, S) \vdash_{\mathcal{M}}^* (s_0, \varepsilon, \alpha)$

Daraus folgt unmittelbar:

$$L(G) = L_{\ell}(\mathcal{M})$$

Beispiel 22.14

Die Sprache

$$L = \{ww^R \mid w \in \{a,b\}^+\}$$

wird generiert von der GNF-Grammatik $G = (\{S, A, B\}, \{a, b\}, R, S)$ mit

$$R = \{ S \rightarrow aSA \mid bSB \mid aA \mid bB \\ A \rightarrow a \\ B \rightarrow b \}$$

Daraus kann man einen PDA mit den folgenden Regeln konstruieren:

$$(s_0, a, S) \Delta (s_0, SA)$$

$$(s_0, a, S) \Delta (s_0, A)$$

$$(s_0, b, S) \Delta (s_0, SB)$$

$$(s_0, b, S) \Delta (s_0, B)$$

$$(s_0, a, A) \Delta (s_0, \varepsilon)$$

$$(s_0, b, B) \Delta (s_0, \varepsilon)$$