Vorlesung

Grundlagen der Theoretischen Informatik / Einführung in die Theoretische Informatik I

Bernhard Beckert

Institut für Informatik

UNIVERSITÄT KOBLENZ · LANDAU

Sommersemester 2007

Dank

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

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Lemma 5.15 (PDA $\longrightarrow$ cf-Grammatik)

Zu jedem Push-Down-Automaten $\mathcal{M}$ gibt es eine kontextfreie Grammatik G mit

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

Beweis

Sei $\mathcal{M}$ ein PDA, der eine Sprache L **über leeren Keller** akzeptiert.
Wir konstruieren aus dem Regelsatz von $\mathcal{M}$ eine kontextfreie Grammatik, die L erzeugt.

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Lemma 5.15 (PDA $\longrightarrow$ cf-Grammatik)

Zu jedem Push-Down-Automaten $\mathcal{M}$ gibt es eine kontextfreie Grammatik G mit

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

Beweis

Sei $\mathcal{M}$ ein PDA, der eine Sprache L **über leeren Keller** akzeptiert.
Wir konstruieren aus dem Regelsatz von $\mathcal{M}$ eine kontextfreie Grammatik, die L erzeugt.

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

Idee:

Die Variablen der Grammatik sind 3-Tupel der Form

$$[q, A, p]$$

Bedeutung:

Grammatik kann Wort x aus Variablen $[q, A, p]$ ableiten

gdw

$\mathcal{M}$ kann vom Zustand q in den Zustand p übergehen,
dabei A vom Keller entfernen (sonst den Keller unveränder lassen) und das Wort x lesen:

$$\left([q, A, p] \Longrightarrow^* x\right) \quad \underline{\text{gdw}} \quad \left((q, x, A\gamma) \vdash^* (p, \varepsilon, \gamma)\right)$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

Idee:

Die Variablen der Grammatik sind 3-Tupel der Form

$$[q, A, p]$$

Bedeutung:

Grammatik kann Wort x aus Variablen $[q, A, p]$ ableiten

gdw

$\mathcal{M}$ kann vom Zustand q in den Zustand p übergehen,
dabei A vom Keller entfernen (sonst den Keller unveränder lassen) und das Wort x lesen:

$$\left([q, A, p] \Longrightarrow^* x\right) \quad \underline{\text{gdw}} \quad \left((q, x, A\gamma) \vdash^* (p, \varepsilon, \gamma)\right)$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

Idee:

Die Variablen der Grammatik sind 3-Tupel der Form

$$[q, A, p]$$

Bedeutung:

Grammatik kann Wort x aus Variablen $[q, A, p]$ ableiten

<u>gdw</u>

$\mathcal{M}$ kann vom Zustand q in den Zustand p übergehen,
dabei A vom Keller entfernen (sonst den Keller unveränder lassen) und das Wort x lesen:

$$\left([q, A, p] \Longrightarrow^* x\right) \quad \underline{\text{gdw}} \quad \left((q, x, A\gamma) \vdash^* (p, \varepsilon, \gamma)\right)$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

Formale Konstruktion:

Sei

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

ein PDA.

Daraus konstruiert man die Grammatik

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

mit

$$V := \{[q, A, p] \mid q, p \in K,\ A \in \Gamma\} \cup \{S\}$$
$$T := \Sigma$$

und …

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

…folgenden Regeln in R:

- $S \rightarrow [s_0, Z_0, q]$ für alle $q \in K$,
- $[q, A, q_{m+1}] \rightarrow a\,[q_1, B_1, q_2][q_2, B_2, q_3]\dots[q_m, B_m, q_{m+1}]$
  für jeden Δ-Übergang $(q, a, A)\,\Delta\,(q_1, B_1 \dots B_m)$ und
  für jede beliebige Kombination $q_2, \dots, q_{m+1} \in K$,
- $[q, A, q_1] \rightarrow a$
  für jeden Δ-Übergang $(q, a, A)\;\Delta\;(q_1, \varepsilon)$

Dabei ist $a \in \Sigma \cup \{\varepsilon\}$.

Siehe Buch für Beweis, dass die Konstruktion das gewünschte Ergebnis liefert:

$$\left([q, A, p] \Longrightarrow^* x\right) \;\underline{gdw}\; \left((q, x, A) \vdash^* (p, \varepsilon, \varepsilon)\right)$$

woraus sofort $L_\ell(\mathcal{M}) = L(G)$ *folgt.*

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

…folgenden Regeln in R:

- $S \to [s_0, Z_0, q]$ für alle $q \in K$,
- $[q, A, q_{m+1}] \to a\,[q_1, B_1, q_2][q_2, B_2, q_3] \ldots [q_m, B_m, q_{m+1}]$
  für jeden Δ-Übergang $(q, a, A)\,\Delta\,(q_1, B_1 \ldots B_m)$ und
  für jede beliebige Kombination $q_2, \ldots, q_{m+1} \in K$,
- $[q, A, q_1] \to a$
  für jeden Δ-Übergang $(q, a, A)\ \Delta\ (q_1, \varepsilon)$

Dabei ist $a \in \Sigma \cup \{\varepsilon\}$.

Siehe Buch für Beweis, dass die Konstruktion das gewünschte Ergebnis liefert:

$$\left([q, A, p] \Longrightarrow^* x\right)\ \underline{gdw}\ \left((q, x, A) \vdash^* (p, \varepsilon, \varepsilon)\right)$$

woraus sofort $L_\ell(\mathcal{M}) = L(G)$ *folgt.*

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

...folgenden Regeln in R:

- $S \longrightarrow [s_0, Z_0, q]$ für alle $q \in K$,
- $[q, A, q_{m+1}] \longrightarrow a\,[q_1, B_1, q_2][q_2, B_2, q_3] \dots [q_m, B_m, q_{m+1}]$
  für jeden Δ-Übergang $(q, a, A)\,\Delta\,(q_1, B_1 \dots B_m)$ und
  für jede beliebige Kombination $q_2, \dots, q_{m+1} \in K$,
- $[q, A, q_1] \longrightarrow a$
  für jeden Δ-Übergang $(q, a, A)\ \Delta\ (q_1, \varepsilon)$

Dabei ist $a \in \Sigma \cup \{\varepsilon\}$.

Siehe Buch für Beweis, dass die Konstruktion das gewünschte Ergebnis liefert:

$$\left([q, A, p] \Longrightarrow^* x\right)\ \underline{gdw}\ \left((q, x, A) \vdash^* (p, \varepsilon, \varepsilon)\right)$$

woraus sofort $L_\ell(\mathcal{M}) = L(G)$ *folgt.*

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

…folgenden Regeln in R:

- $S \longrightarrow [s_0, Z_0, q]$ für alle $q \in K$,
- $[q, A, q_{m+1}] \longrightarrow a\,[q_1, B_1, q_2][q_2, B_2, q_3] \ldots [q_m, B_m, q_{m+1}]$
  für jeden Δ-Übergang $(q, a, A)\,\Delta\,(q_1, B_1 \ldots B_m)$ und
  für jede beliebige Kombination $q_2, \ldots, q_{m+1} \in K$,
- $[q, A, q_1] \longrightarrow a$
  für jeden Δ-Übergang $(q, a, A)\;\Delta\;(q_1, \varepsilon)$

Dabei ist $a \in \Sigma \cup \{\varepsilon\}$.

Siehe Buch für Beweis, dass die Konstruktion das gewünschte Ergebnis liefert:

$$\left([q, A, p] \Longrightarrow^* x\right) \ \underline{gdw} \ \left((q, x, A) \vdash^* (p, \varepsilon, \varepsilon)\right)$$

woraus sofort $L_\ell(\mathcal{M}) = L(G)$ *folgt.*

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

…folgenden Regeln in R:

- $S \to [s_0, Z_0, q]$ für alle $q \in K$,
- $[q, A, q_{m+1}] \to a\,[q_1, B_1, q_2][q_2, B_2, q_3] \dots [q_m, B_m, q_{m+1}]$
  für jeden Δ-Übergang $(q, a, A)\,\Delta\,(q_1, B_1 \dots B_m)$ und
  für jede beliebige Kombination $q_2, \dots, q_{m+1} \in K$,
- $[q, A, q_1] \to a$
  für jeden Δ-Übergang $(q, a, A)\ \Delta\ (q_1, \varepsilon)$

Dabei ist $a \in \Sigma \cup \{\varepsilon\}$.

Siehe Buch für Beweis, dass die Konstruktion das gewünschte Ergebnis liefert:

$$\left([q, A, p] \Longrightarrow^* x\right)\ \underline{gdw}\ \left((q, x, A) \vdash^* (p, \varepsilon, \varepsilon)\right)$$

woraus sofort $L_\ell(\mathcal{M}) = L(G)$ *folgt.*

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beweis (Forts.)

…folgenden Regeln in R:

- $S \to [s_0, Z_0, q]$ für alle $q \in K$,
- $[q, A, q_{m+1}] \to a\,[q_1, B_1, q_2][q_2, B_2, q_3]\dots[q_m, B_m, q_{m+1}]$
  für jeden Δ-Übergang $(q, a, A)\,\Delta\,(q_1, B_1 \dots B_m)$ und
  für jede beliebige Kombination $q_2, \dots, q_{m+1} \in K$,
- $[q, A, q_1] \to a$
  für jeden Δ-Übergang $(q, a, A)\ \Delta\ (q_1, \varepsilon)$

Dabei ist $a \in \Sigma \cup \{\varepsilon\}$.

Siehe Buch für Beweis, dass die Konstruktion das gewünschte Ergebnis liefert:

$$\left([q, A, p] \Longrightarrow^* x\right)\ \underline{gdw}\ \left((q, x, A) \vdash^* (p, \varepsilon, \varepsilon)\right)$$

woraus sofort $L_\ell(\mathcal{M}) = L(G)$ *folgt.*

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel 5.16

Sprache:

$$L_{ab} = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

L_{ab} wird über leeren Keller akzeptiert von dem PDA

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

mit den Regeln

1. $(s_0, \varepsilon, Z_0) \;\Delta\; (s_0, \varepsilon)$
2. $(s_0, a, Z_0) \;\Delta\; (s_0, A)$
3. $(s_0, a, A) \;\Delta\; (s_0, AA)$
4. $(s_0, b, A) \;\Delta\; (s_1, \varepsilon)$
5. $(s_1, b, A) \;\Delta\; (s_1, \varepsilon)$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel 5.16

Sprache:

$$L_{ab} = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

L_{ab} wird über leeren Keller akzeptiert von dem PDA

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

mit den Regeln

$$\begin{array}{ll}
1. & (s_0, \varepsilon, Z_0)\ \Delta\ (s_0, \varepsilon) \\
2. & (s_0, a, Z_0)\ \Delta\ (s_0, A) \\
3. & (s_0, a, A)\ \Delta\ (s_0, AA) \\
4. & (s_0, b, A)\ \Delta\ (s_1, \varepsilon) \\
5. & (s_1, b, A)\ \Delta\ (s_1, \varepsilon)
\end{array}$$

Beispiel 5.16

Sprache:

$$L_{ab} = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

L_{ab} wird über leeren Keller akzeptiert von dem PDA

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

mit den Regeln

$$\begin{array}{ll}
1. & (s_0, \varepsilon, Z_0)\ \Delta\ (s_0, \varepsilon) \\
2. & (s_0, a, Z_0)\ \Delta\ (s_0, A) \\
3. & (s_0, a, A)\ \Delta\ (s_0, AA) \\
4. & (s_0, b, A)\ \Delta\ (s_1, \varepsilon) \\
5. & (s_1, b, A)\ \Delta\ (s_1, \varepsilon)
\end{array}$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Die Transformation ergibt folgende Grammatik-Regeln:

$$
\begin{array}{lrcl}
 & S & \to & [s_0, Z_0, s_0] \mid [s_0, Z_0, s_1] \\
1. & [s_0, Z_0, s_0] & \to & \varepsilon \\
2. & [s_0, Z_0, s_0] & \to & a[s_0, A, s_0] \\
 & [s_0, Z_0, s_1] & \to & a[s_0, A, s_1] \\
3. & [s_0, A, s_0] & \to & a[s_0, A, s_0][s_0, A, s_0] \\
 & [s_0, A, s_0] & \to & a[s_0, A, s_1][s_1, A, s_0] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_0][s_0, A, s_1] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_1][s_1, A, s_1] \\
4. & [s_0, A, s_1] & \to & b \\
5. & [s_1, A, s_1] & \to & b
\end{array}
$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Die Transformation ergibt folgende Grammatik-Regeln:

$$
\begin{array}{lll}
 & S & \rightarrow [s_0, Z_0, s_0] \mid [s_0, Z_0, s_1] \\
1. & [s_0, Z_0, s_0] & \rightarrow \varepsilon \\
2. & [s_0, Z_0, s_0] & \rightarrow a[s_0, A, s_0] \\
 & [s_0, Z_0, s_1] & \rightarrow a[s_0, A, s_1] \\
3. & [s_0, A, s_0] & \rightarrow a[s_0, A, s_0][s_0, A, s_0] \\
 & [s_0, A, s_0] & \rightarrow a[s_0, A, s_1][s_1, A, s_0] \\
 & [s_0, A, s_1] & \rightarrow a[s_0, A, s_0][s_0, A, s_1] \\
 & [s_0, A, s_1] & \rightarrow a[s_0, A, s_1][s_1, A, s_1] \\
4. & [s_0, A, s_1] & \rightarrow b \\
5. & [s_1, A, s_1] & \rightarrow b
\end{array}
$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Die Transformation ergibt folgende Grammatik-Regeln:

$$
\begin{array}{lrcl}
 & S & \to & [s_0, Z_0, s_0] \mid [s_0, Z_0, s_1] \\
1. & [s_0, Z_0, s_0] & \to & \varepsilon \\
2. & [s_0, Z_0, s_0] & \to & a[s_0, A, s_0] \\
 & [s_0, Z_0, s_1] & \to & a[s_0, A, s_1] \\
3. & [s_0, A, s_0] & \to & a[s_0, A, s_0][s_0, A, s_0] \\
 & [s_0, A, s_0] & \to & a[s_0, A, s_1][s_1, A, s_0] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_0][s_0, A, s_1] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_1][s_1, A, s_1] \\
4. & [s_0, A, s_1] & \to & b \\
5. & [s_1, A, s_1] & \to & b
\end{array}
$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Die Transformation ergibt folgende Grammatik-Regeln:

$$
\begin{array}{lrcl}
 & S & \to & [s_0, Z_0, s_0] \mid [s_0, Z_0, s_1] \\
1. & [s_0, Z_0, s_0] & \to & \varepsilon \\
2. & [s_0, Z_0, s_0] & \to & a[s_0, A, s_0] \\
 & [s_0, Z_0, s_1] & \to & a[s_0, A, s_1] \\
3. & [s_0, A, s_0] & \to & a[s_0, A, s_0][s_0, A, s_0] \\
 & [s_0, A, s_0] & \to & a[s_0, A, s_1][s_1, A, s_0] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_0][s_0, A, s_1] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_1][s_1, A, s_1] \\
4. & [s_0, A, s_1] & \to & b \\
5. & [s_1, A, s_1] & \to & b
\end{array}
$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Die Transformation ergibt folgende Grammatik-Regeln:

$$
\begin{array}{lrcl}
 & S & \to & [s_0, Z_0, s_0] \mid [s_0, Z_0, s_1] \\
1. & [s_0, Z_0, s_0] & \to & \varepsilon \\
2. & [s_0, Z_0, s_0] & \to & a[s_0, A, s_0] \\
 & [s_0, Z_0, s_1] & \to & a[s_0, A, s_1] \\
3. & [s_0, A, s_0] & \to & a[s_0, A, s_0][s_0, A, s_0] \\
 & [s_0, A, s_0] & \to & a[s_0, A, s_1][s_1, A, s_0] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_0][s_0, A, s_1] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_1][s_1, A, s_1] \\
4. & [s_0, A, s_1] & \to & b \\
5. & [s_1, A, s_1] & \to & b
\end{array}
$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Die Transformation ergibt folgende Grammatik-Regeln:

$$
\begin{array}{lrcl}
 & S & \to & [s_0, Z_0, s_0] \mid [s_0, Z_0, s_1] \\
1. & [s_0, Z_0, s_0] & \to & \varepsilon \\
2. & [s_0, Z_0, s_0] & \to & a[s_0, A, s_0] \\
 & [s_0, Z_0, s_1] & \to & a[s_0, A, s_1] \\
3. & [s_0, A, s_0] & \to & a[s_0, A, s_0][s_0, A, s_0] \\
 & [s_0, A, s_0] & \to & a[s_0, A, s_1][s_1, A, s_0] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_0][s_0, A, s_1] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_1][s_1, A, s_1] \\
4. & [s_0, A, s_1] & \to & b \\
5. & [s_1, A, s_1] & \to & b
\end{array}
$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Die Transformation ergibt folgende Grammatik-Regeln:

$$
\begin{array}{lrcl}
 & S & \to & [s_0, Z_0, s_0] \mid [s_0, Z_0, s_1] \\
1. & [s_0, Z_0, s_0] & \to & \varepsilon \\
2. & [s_0, Z_0, s_0] & \to & a[s_0, A, s_0] \\
 & [s_0, Z_0, s_1] & \to & a[s_0, A, s_1] \\
3. & [s_0, A, s_0] & \to & a[s_0, A, s_0][s_0, A, s_0] \\
 & [s_0, A, s_0] & \to & a[s_0, A, s_1][s_1, A, s_0] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_0][s_0, A, s_1] \\
 & [s_0, A, s_1] & \to & a[s_0, A, s_1][s_1, A, s_1] \\
4. & [s_0, A, s_1] & \to & b \\
5. & [s_1, A, s_1] & \to & b
\end{array}
$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Lesbarer haben wir damit folgende Grammatik:

$$
\begin{aligned}
S &\longrightarrow A \mid B \\
A &\longrightarrow aC \mid \varepsilon \\
B &\longrightarrow aD \\
C &\longrightarrow aCC \mid aDE \\
D &\longrightarrow aCD \mid aDF \mid b \\
F &\longrightarrow b
\end{aligned}
$$

Man sieht jetzt:

- Variable E ist nutzlos, und damit auch die Variable C.
- Die Grammatik enthält Kettenproduktionen und nullbare Variablen.

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Lesbarer haben wir damit folgende Grammatik:

$$
\begin{aligned}
S &\longrightarrow A \mid B \\
A &\longrightarrow aC \mid \varepsilon \\
B &\longrightarrow aD \\
C &\longrightarrow aCC \mid aDE \\
D &\longrightarrow aCD \mid aDF \mid b \\
F &\longrightarrow b
\end{aligned}
$$

Man sieht jetzt:

- Variable E ist nutzlos, und damit auch die Variable C.
- Die Grammatik enthält Kettenproduktionen und nullbare Variablen.

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Lesbarer haben wir damit folgende Grammatik:

$$
\begin{aligned}
S &\longrightarrow A \mid B \\
A &\longrightarrow aC \mid \varepsilon \\
B &\longrightarrow aD \\
C &\longrightarrow aCC \mid aDE \\
D &\longrightarrow aCD \mid aDF \mid b \\
F &\longrightarrow b
\end{aligned}
$$

Man sieht jetzt:

- Variable E ist nutzlos, und damit auch die Variable C.
- Die Grammatik enthält Kettenproduktionen und nullbare Variablen.

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Nach Entfernung der überflüssigen Elemente:

$$\begin{aligned} S &\to \varepsilon \mid aD \\ D &\to aDF \mid b \\ F &\to b \end{aligned}$$

Mit dieser Grammatik kann man z.B. folgende Ableitung ausführen:

$$S \Longrightarrow aD \Longrightarrow aaDF \Longrightarrow aaaDFF \Longrightarrow aaabFF \\ \Longrightarrow aaabbF \Longrightarrow aaabbb$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Nach Entfernung der überflüssigen Elemente:

$$\begin{aligned} S &\to \varepsilon \mid aD \\ D &\to aDF \mid b \\ F &\to b \end{aligned}$$

Mit dieser Grammatik kann man z.B. folgende Ableitung ausführen:

$$\begin{aligned} &S \Longrightarrow aD \Longrightarrow aaDF \Longrightarrow aaaDFF \Longrightarrow aaabFF \\ &\Longrightarrow aaabbF \Longrightarrow aaabbb \end{aligned}$$

Gleichmächtigkeit: PDAs und kontextfreie Grammatiken

Beispiel (Forts.)

Nach Entfernung der überflüssigen Elemente:

$$\begin{aligned} S &\to \varepsilon \mid aD \\ D &\to aDF \mid b \\ F &\to b \end{aligned}$$

Mit dieser Grammatik kann man z.B. folgende Ableitung ausführen:

$$\begin{aligned} &S \Longrightarrow aD \Longrightarrow aaDF \Longrightarrow aaaDFF \Longrightarrow aaabFF \\ &\Longrightarrow aaabbF \Longrightarrow aaabbb \end{aligned}$$

Teil I

Teil I

Abschlusseigenschaften

Theorem 6.1 (Abschlusseigenschaften von $\mathbf{L}_2$)

$\mathbf{L}_2$ *ist abgeschlossen gegen:*

- *Vereinigung* $\cup$
- *Konkatenation* $\circ$
- *Kleene-Stern* *

Beweis

Seien

$$G_i = (V_i, T_i, R_i, S_i) \qquad (i \in \{1,2\})$$

zwei cf-Grammatiken mit $V_1 \cap V_2 = \emptyset$.
Sei

$$L_i = L(G_i)$$

Abschlusseigenschaften

Theorem 6.1 (Abschlusseigenschaften von $\mathbf{L}_2$)

$\mathbf{L}_2$ *ist abgeschlossen gegen:*

- *Vereinigung* $\cup$
- *Konkatenation* $\circ$
- *Kleene-Stern* *

Beweis

Seien

$$G_i = (V_i, T_i, R_i, S_i) \qquad (i \in \{1,2\})$$

zwei cf-Grammatiken mit $V_1 \cap V_2 = \emptyset$.

Sei

$$L_i = L(G_i)$$

Abschlusseigenschaften

Theorem 6.1 (Abschlusseigenschaften von $\mathbf{L}_2$)

$\mathbf{L}_2$ *ist abgeschlossen gegen:*

- *Vereinigung* $\cup$
- *Konkatenation* $\circ$
- *Kleene-Stern* *

Beweis

Seien

$$G_i = (V_i, T_i, R_i, S_i) \qquad (i \in \{1, 2\})$$

zwei cf-Grammatiken mit $V_1 \cap V_2 = \emptyset$.
Sei

$$L_i = L(G_i)$$

Abschlusseigenschaften

Beweis (Forts.)

zu $\cup$:

$$(V_1 \cup V_2 \cup \{S_{neu}\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{S_{neu} \to S_1 \mid S_2\}, S_{neu})$$

erzeugt gerade $L_1 \cup L_2$

zu $\circ$:

$$(V_1 \cup V_2 \cup \{S_{neu}\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{S_{neu} \to S_1 S_2\}, S_{neu})$$

erzeugt gerade $L_1 \circ L_2$

zu *:

$$(V_1 \cup \{S_{neu}\}, T_1, R_1 \cup \{S_{neu} \to S_1 S_{neu} \mid \varepsilon\}, S_{neu})$$

erzeugt gerade L_1^*. □

Abschlusseigenschaften

Beweis (Forts.)

zu $\cup$:

$$(V_1 \cup V_2 \cup \{S_{neu}\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{S_{neu} \to S_1 \mid S_2\}, S_{neu})$$

erzeugt gerade $L_1 \cup L_2$

zu $\circ$:

$$(V_1 \cup V_2 \cup \{S_{neu}\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{S_{neu} \to S_1 S_2\}, S_{neu})$$

erzeugt gerade $L_1 \circ L_2$

zu *:

$$(V_1 \cup \{S_{neu}\}, T_1, R_1 \cup \{S_{neu} \to S_1 S_{neu} \mid \varepsilon\}, S_{neu})$$

erzeugt gerade L_1^*. □

Beweis (Forts.)

zu $\cup$:

$$(V_1 \cup V_2 \cup \{S_{neu}\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{S_{neu} \to S_1 \mid S_2\}, S_{neu})$$

erzeugt gerade $L_1 \cup L_2$

zu $\circ$:

$$(V_1 \cup V_2 \cup \{S_{neu}\}, T_1 \cup T_2, R_1 \cup R_2 \cup \{S_{neu} \to S_1 S_2\}, S_{neu})$$

erzeugt gerade $L_1 \circ L_2$

zu *:

$$(V_1 \cup \{S_{neu}\}, T_1, R_1 \cup \{S_{neu} \to S_1 S_{neu} \mid \varepsilon\}, S_{neu})$$

erzeugt gerade L_1^*. □

Abschlusseigenschaften

Theorem 6.2 (Abschlusseigenschaften von $\mathbf{L}_2$)

$\mathbf{L}_2$ *ist **nicht** abgeschlossen gegen:*

- *Durchschnitt* $\cap$
- *Komplement* $\neg$

Beweis

Zu „$\cap$“:

$$L_1 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\}$$
$$L_2 = \{a^m b^n c^n \mid n, m \in \mathbb{N}\}$$

wird erzeugt von $G_i = (\{S, S', T\}, \{a, b, c\}, R_i, S)$ mit

$$R_1 = \{ \quad S \to S'T, \quad S' \to aS'b \mid ab, \quad T \to cT \mid c \}$$

$$R_2 = \{ \quad S \to TS', \quad S' \to bS'c \mid bc, \quad T \to aT \mid a \}$$

Sowohl L_1 als auch L_2 sind cf, **nicht** aber $L_1 \cap L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Abschlusseigenschaften

Beweis

Zu „∩“:

$$L_1 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\}$$
$$L_2 = \{a^m b^n c^n \mid n, m \in \mathbb{N}\}$$

wird erzeugt von $G_i = (\{S, S', T\}, \{a, b, c\}, R_i, S)$ mit

$$R_1 = \{ \quad S \rightarrow S'T, \quad S' \rightarrow aS'b \mid ab, \quad T \rightarrow cT \mid c\}$$

$$R_2 = \{ \quad S \rightarrow TS', \quad S' \rightarrow bS'c \mid bc, \quad T \rightarrow aT \mid a\}$$

Sowohl L_1 als auch L_2 sind cf, **nicht** aber $L_1 \cap L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Abschlusseigenschaften

Beweis

Zu „$\cap$“:

$$L_1 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\}$$
$$L_2 = \{a^m b^n c^n \mid n, m \in \mathbb{N}\}$$

wird erzeugt von $G_i = (\{S, S', T\}, \{a, b, c\}, R_i, S)$ mit

$$R_1 = \{ \quad \begin{aligned} S &\rightarrow S'T \\ S' &\rightarrow aS'b \mid ab \\ T &\rightarrow cT \mid c\} \end{aligned}$$

$$R_2 = \{ \quad \begin{aligned} S &\rightarrow TS' \\ S' &\rightarrow bS'c \mid bc \\ T &\rightarrow aT \mid a\} \end{aligned}$$

Sowohl L_1 als auch L_2 sind cf, **nicht** aber $L_1 \cap L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Abschlusseigenschaften

Beweis

Zu „¬“:

Angenommen, $\mathbf{L}_2$ wäre abgeschlossen gegen ¬.

Wegen

$$L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$$

wäre $\mathbf{L}_2$ dann auch abgeschlossen gegen ∩ – **Widerspruch** □

Abschlusseigenschaften

Beweis

Zu „$\neg$“:

Angenommen, $\mathbf{L}_2$ wäre abgeschlossen gegen $\neg$.

Wegen

$$L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$$

wäre $\mathbf{L}_2$ dann auch abgeschlossen gegen $\cap$ – **Widerspruch** □

Abschlusseigenschaften

Beweis

Zu „¬“:

Angenommen, $\mathbf{L}_2$ wäre abgeschlossen gegen $\neg$.

Wegen

$$L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$$

wäre $\mathbf{L}_2$ dann auch abgeschlossen gegen $\cap$ – **Widerspruch** □

Teil I

Teil I

Wortproblem

Problem

Gegeben: eine cf-Grammatik G, so daß $L(G)$ eine Sprache ist über Σ, und ein Wort $w \in \Sigma^*$

Frage: Ist $w \in L(G)$?

Wortproblem

Problem

Gegeben: eine cf-Grammatik G, so daß $L(G)$ eine Sprache ist über Σ, und ein Wort $w \in \Sigma^*$

Frage: Ist $w \in L(G)$?

Wortproblem

Lösung des Wortproblems für L_3

Gegeben eine rechtslineare Grammatik G, so daß $L(G)$ eine Sprache ist über Σ, und ein Wort $w \in \Sigma^*$.

- Konstruiere aus G einen ε-NDEA A_1.
- Konstruiere aus A_1 einen NDEA A_2.
- Konstruiere aus A_2 einen DEA A_3.
- Probiere aus, ob A_3 das Wort w akzeptiert.
  Dazu braucht der Automat A_3 genau $|w|$ Schritte.

Wortproblem

Lösung des Wortproblems für L_3

Gegeben eine rechtslineare Grammatik G, so daß $L(G)$ eine Sprache ist über Σ, und ein Wort $w \in \Sigma^*$.

- Konstruiere aus G einen ε-NDEA A_1.
- Konstruiere aus A_1 einen NDEA A_2.
- Konstruiere aus A_2 einen DEA A_3.
- Probiere aus, ob A_3 das Wort w akzeptiert.
  Dazu braucht der Automat A_3 genau $|w|$ Schritte.

Wortproblem

Lösung des Wortproblems für L_3

Gegeben eine rechtslineare Grammatik G, so daß $L(G)$ eine Sprache ist über Σ, und ein Wort $w \in \Sigma^*$.

- Konstruiere aus G einen ε-NDEA A_1.
- Konstruiere aus A_1 einen NDEA A_2.
- Konstruiere aus A_2 einen DEA A_3.
- Probiere aus, ob A_3 das Wort w akzeptiert.
  Dazu braucht der Automat A_3 genau $|w|$ Schritte.

Wortproblem

Lösung des Wortproblems für L_3

Gegeben eine rechtslineare Grammatik G, so daß $L(G)$ eine Sprache ist über Σ, und ein Wort $w \in \Sigma^*$.

- Konstruiere aus G einen ε-NDEA A_1.
- Konstruiere aus A_1 einen NDEA A_2.
- Konstruiere aus A_2 einen DEA A_3.
- Probiere aus, ob A_3 das Wort w akzeptiert.
  Dazu braucht der Automat A_3 genau $|w|$ Schritte.

Wortproblem

Das Wortproblem für L_2

- Zu jeder cf-Grammatik G kann man einen PDA konstruieren
- Aber ein Pushdown-Automat kann ε-Übergänge machen, in denen er das Wort nicht weiter liest.
- **Wie kann man dann garantieren, daß der Automat in endlich vielen Schritten das Wort w zu Ende gelesen hat?**
- Deshalb: verwende anderes Verfahren:
  Cocke-Younger-Kasami-Algorithmus (CYK-Algorithmus)
  Auch: Chart-Parsing

Wortproblem

Das Wortproblem für L_2

- Zu jeder cf-Grammatik G kann man einen PDA konstruieren
- Aber ein Pushdown-Automat kann ε-Übergänge machen, in denen er das Wort nicht weiter liest.
- **Wie kann man dann garantieren, daß der Automat in endlich vielen Schritten das Wort w zu Ende gelesen hat?**
- Deshalb: verwende anderes Verfahren:
  Cocke-Younger-Kasami-Algorithmus (CYK-Algorithmus)
  Auch: Chart-Parsing

Wortproblem

Das Wortproblem für L_2

- Zu jeder cf-Grammatik G kann man einen PDA konstruieren
- Aber ein Pushdown-Automat kann ε-Übergänge machen, in denen er das Wort nicht weiter liest.
- **Wie kann man dann garantieren, daß der Automat in endlich vielen Schritten das Wort w zu Ende gelesen hat?**
- Deshalb: verwende anderes Verfahren:
  Cocke-Younger-Kasami-Algorithmus (CYK-Algorithmus)
  Auch: Chart-Parsing

Wortproblem

Das Wortproblem für L_2

- Zu jeder cf-Grammatik G kann man einen PDA konstruieren
- Aber ein Pushdown-Automat kann ε-Übergänge machen, in denen er das Wort nicht weiter liest.
- **Wie kann man dann garantieren, daß der Automat in endlich vielen Schritten das Wort w zu Ende gelesen hat?**
- Deshalb: verwende anderes Verfahren:
  Cocke-Younger-Kasami-Algorithmus (CYK-Algorithmus)
  Auch: Chart-Parsing

Chart-Parsing

Gegeben: Ein Wort

$$w = a_1 \dots a_n$$

Idee

- Prinzip der dynamischen Programmierung
- 1.: Ermittle woraus sich die einstelligen Teilworte ableiten lassen
  2.: Ermittle woraus sich die zweistelligen Teilworte ableiten lassen
  $\dots$
  n.: Ermittle woraus sich die n-stelligen Teilworte (w selbst) ableiten lassen

Chart-Parsing

Gegeben: Ein Wort

$$w = a_1 \dots a_n$$

Idee

- Prinzip der dynamischen Programmierung
- 1.: Ermittle woraus sich die einstelligen Teilworte ableiten lassen
  2.: Ermittle woraus sich die zweistelligen Teilworte ableiten lassen
  $\dots$
  n.: Ermittle woraus sich die n-stelligen Teilworte (w selbst) ableiten lassen