Automaten und Formale Sprachen
14. Vorlesung

Prof. Dr. Dietrich Kuske

FG Automaten und Logik, TU Ilmenau

Wintersemester 2022/23

letzte Vorlesung:

- Schnitt und Komplement kontextfreier Sprachen i.a. nicht kontextfrei

heute:

- Schnitt einer kontextfreien und einer regulären Sprache ist kontextfrei (dazu: Kellerautomaten mit Endzuständen)
- wird L von einem deterministischen PDA akzeptiert, so gilt dies auch für $\Sigma^* \setminus L$

PDAs mit Endzuständen

Definition

Ein Kellerautomat mit Endzuständen oder PDAE ist ein 7-Tupel $M = (Z, \Sigma, \Gamma, \iota, \delta, \#, E)$, wobei

- $(Z, \Sigma, \Gamma, \iota, \delta, \#)$ ein PDA und
- $E \subseteq Z$ eine Menge von Endzuständen ist.

Definition

Sei $M = (Z, \Sigma, \Gamma, \iota, \delta, \#, E)$ ein PDAE. Die von M akzeptierte Sprache ist

$$L(M) = \left\{ w \in \Sigma^* \;\middle|\; \text{es gibt } e \in E \text{ und } \gamma \in \Gamma^* \text{ mit } (\iota, w, \#) \vdash^* (e, \varepsilon, \gamma) \right\}.$$

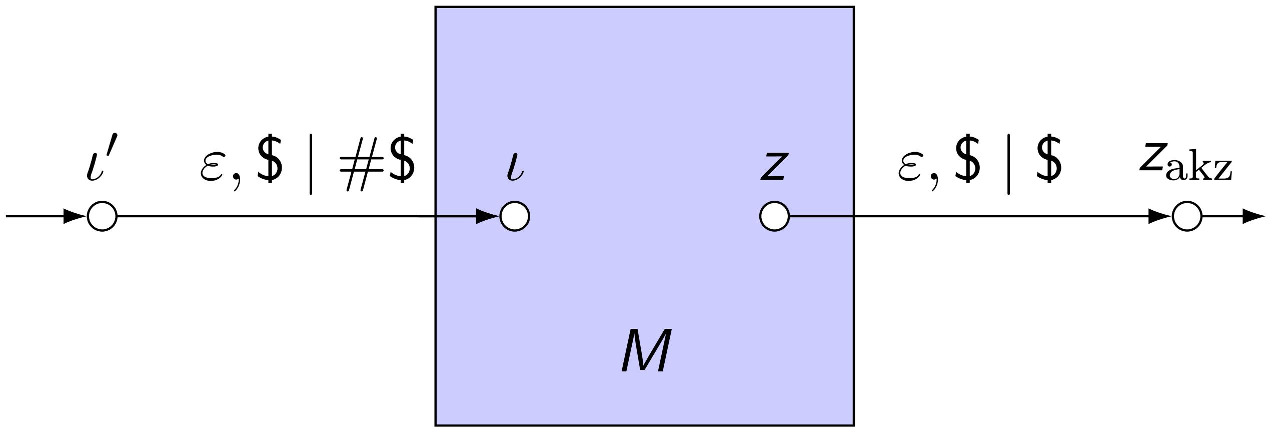
PDAEs akzeptieren also so, wie es NFAs tun: Der Inhalt des Kellers nach dem kompletten Lesen der Eingabe ist irrelevant, es kommt nur auf den erreichten Zustand an.

Lemma

Jede kontextfreie Sprache wird von einem PDAE akzeptiert.

Beweisidee: Sei $M = (Z, \Sigma, \Gamma, \iota, \delta, \#)$ ein PDA.

- Führe ein neues Kellerinitialisierungszeichen \$ ein und ersetze im 1. Schritt einer Berechnung mithilfe einer ε-Transition \$ durch $\#\$$.
- Simuliere dann den PDA M.
- Wenn (nur) \$ auf dem Keller steht, wechsele mit einer ε-Transition in einen akzeptierenden Zustand.



etwas genauer: Wir betrachten den folgenden PDAE:

$$M' = \big(Z \uplus \{\iota', z_{\mathrm{akz}}\}, \Sigma, \Gamma \uplus \{\$\}, \iota', \delta', \$, \{z_{\mathrm{akz}}\}\big)$$

wobei für $z \in Z \uplus \{\iota', z_{\mathrm{akz}}\}$, $a \in \Sigma$ und $A \in \Gamma \cup \{\$\}$ gilt:

$$\delta'(z, a, A) \quad = \quad \begin{cases} \delta(z, a, A) & \text{falls } z \in Z, A \in \Gamma \\ \emptyset & \text{sonst} \end{cases}$$

$$\delta'(z, \varepsilon, A) \quad = \quad \begin{cases} \delta(z, \varepsilon, A) & \text{falls } z \in Z, A \in \Gamma \\ \big\{(z_{\mathrm{akz}}, \$)\big\} & \text{falls } z \in Z, A = \$ \\ \big\{(\iota, \#\$)\big\} & \text{falls } z = \iota', A = \$ \\ \emptyset & \text{sonst.} \end{cases}$$

Dann haben die akzeptierenden Berechnungen von M' folgende Form:

$$(\iota', w, \$) \vdash_{M'} (\iota, w, \#\$) \vdash^*_{M'} (z, \varepsilon, \$) \vdash_{M'} (z_{\mathrm{akz}}, \varepsilon, \$)$$

wobei

$$(\iota, w, \#) \vdash^*_{M} (z, \varepsilon, \varepsilon)$$

gilt. Also haben wir

$$\begin{aligned} w \in L(M) \quad &\Longleftrightarrow \quad (\iota, w, \#) \vdash^*_{M} (z, \varepsilon, \varepsilon) \\ &\Longleftrightarrow \quad (\iota', w, \$) \vdash^*_{M'} (z_{\mathrm{akz}}, \varepsilon, \$) \\ &\Longleftrightarrow \quad w \in L(M') \end{aligned}$$

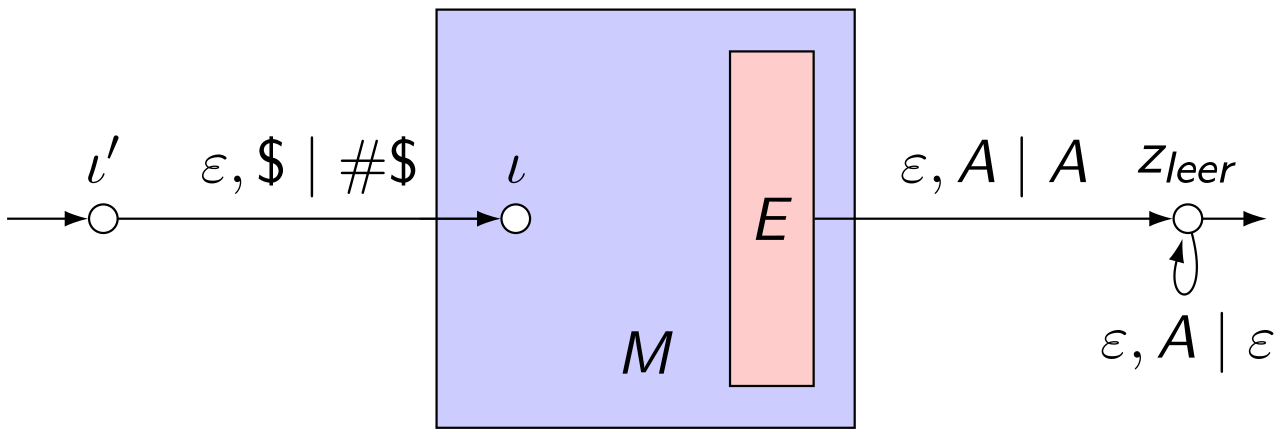
□

Lemma

Ist M ein PDAE, so wird $L(M)$ von einem PDA akzeptiert, ist also kontextfrei.

Beweisidee:

- Führe ein neues Kellerinitialisierungszeichen \$ ein und ersetze im 1. Schritt einer Berechnung mithilfe einer ε-Transition \$ durch $\#\$$.
- Simuliere dann den PDAE M.
- Aus jedem Endzustand kann mittels ε-Transition in einen Zustand gewechselt werden, der den Keller leert.



genauer: Sei $M = (Z, \Sigma, \Gamma, \iota, \delta, \#, E)$ ein PDAE. Betrachte den PDA

$$M' = \left(Z \uplus \{\iota', z_{leer}\}, \Sigma, \Gamma \uplus \{\$\}, \iota', \delta', \$\right)$$

wobei für $z \in Z \cup \{\iota', z_{leer}\}$, $a \in \Sigma$ und $A \in \Gamma$ gilt:

$$\begin{aligned}
\delta'(z, a, A) &= \begin{cases} \delta(z, a, A) & \text{falls } z \in Z, A \in \Gamma \\ \emptyset & \text{sonst} \end{cases} \\
\delta'(z, \varepsilon, A) &= \begin{cases} \left\{(\iota, \#\$)\right\} & \text{falls } z = \iota', A = \$ \\ \delta(z, \varepsilon, A) & \text{falls } z \in Z \setminus E \\ \delta(z, \varepsilon, A) \cup \left\{(z_{leer}, A)\right\} & \text{falls } z \in E \\ \left\{(z_{leer}, \varepsilon)\right\} & \text{falls } z = z_{leer} \end{cases}
\end{aligned}$$

Dann haben die akzeptierenden Berechnungen von M' folgende Form:

$$(\iota', w, \$) \vdash_{M'} (\iota, w, \#\$) \vdash^*_{M'} (z, \varepsilon, \gamma\$) \vdash_{M'} (z_{leer}, \varepsilon, \gamma\$) \vdash^*_{M'} (z_{leer}, \varepsilon, \varepsilon)$$

wobei

$$(\iota, w, \#) \vdash^*_{M} (z, \varepsilon, \gamma) \text{ und } z \in E$$

gelten. Also haben wir

$$\begin{aligned} w \in L(M) \quad &\iff \quad (\iota, w, \#) \vdash^*_{M} (z, \varepsilon, \gamma) \text{ mit } z \in E \\ &\iff \quad (\iota', w, \$) \vdash^*_{M'} (z_{leer}, \varepsilon, \varepsilon) \\ &\iff \quad w \in L(M') \end{aligned}$$

$\square$

Diese zwei Lemmata zeigen

Satz

Eine Sprache L ist genau dann kontextfrei, wenn sie von einem PDAE akzeptiert wird.

Wir werden diesen Satz nun verwenden, um zu zeigen, daß der Schnitt einer kontextfreien und einer <u>regulären</u> Sprache wieder kontextfrei ist (der Schnitt zweiter kontextfreier Sprachen ist i.a. nicht kontextfrei, siehe Folie ca. 13.28).

Satz

Seien L eine kontextfreie und R eine reguläre Sprache. Dann ist $L \cap R$ kontextfrei.

Beweis: (analog der Kreuzprodukt-Konstruktion für NFAs)

Sei $M = (Z_1, \Sigma, \Gamma, \iota_1, \delta_1, \#, E_1)$ ein PDAE mit $L(M) = L$.

Sei $A = (Z_2, \Sigma, \iota_2, \delta_2, E_2)$ ein DFA mit $L(A) = R$.

Konstruktion eines PDAE M' für $L \cap R$:

$$M' = \left(Z_1 \times Z_2, \Sigma, \Gamma, (\iota_1, \iota_2), \delta, \#, E_1 \times E_2\right).$$

Hierbei ist die Überführungsfunktion δ wie folgt definiert:

$$\big((z_1', z_2'), B_1 \dots B_k\big) \in \delta\big((z_1, z_2), a, A\big) \text{ gdw.}$$
$$(z_1', B_1 \dots B_k) \in \delta_1(z_1, a, A) \text{ und } z_2' = \widehat{\delta_2}(z_2, a)$$

Man zeigt dann per Induktion über die Länge der Ableitung:

Für alle $w \in \Sigma^*$, $\gamma, \gamma' \in \Gamma^*$, $z_1, z_1' \in Z_1$, $z_2, z_2' \in Z_2$ gilt

$$\begin{aligned} & \big((z_1, z_2), w, \gamma\big) \vdash_{M'}^* \big((z_1', z_2'), \varepsilon, \gamma'\big) \\ \iff \quad & (z_1, w, \gamma) \vdash_M^* (z_1', \varepsilon, \gamma') \text{ und } z_2' = \widehat{\delta_2}(z_2, w) \end{aligned}$$

Mit $(z_1, z_2) = (\iota_1, \iota_2)$ und $(z_1', z_2') \in E_1 \times E_2$ erhält man hieraus

$$w \in L(M) \cap L(A) = L \cap R \iff w \in L(M')\,.$$

$\square$

Beispiel

$$L = \{u_1 3 \ldots u_{r-1} 3 u_r 2 v_1 3 \ldots v_{s-1} 3 v_s \mid \begin{array}{l} u_1, \ldots, u_r, v_1, \ldots, v_s \in \{0,1\}^* \\ \forall j \le s \, \exists i \le r : v_j = u_i \} \end{array}$$

ist nicht kontextfrei.

Beweisidee: Wäre L kontextfrei, so auch $L \cap \{0,1\}^*\{2\}\{0,1\}^* = \{w2w \mid w \in \{0,1\}^*\}$. Das ist aber nicht der Fall (vgl. Folie F-L3-nicht-kontextfrei). □

Hintergrund: Viele Programmiersprachen haben die „Kontextbedingung“, daß eine Variable vor der Benutzung deklariert werden muß.

Diese Eigenschaft kann nicht in einer kontextfreien Grammatik ausgedrückt werden.

Deterministisch kontextfreie Sprachen

Definition

Ein deterministischer Kellerautomat oder DPDA ist ein PDAE $M = (Z, \Sigma, \Gamma, \iota, \delta, \#, E)$, so daß für alle $z \in Z$, $a \in \Sigma$ und $A \in \Gamma$ gilt

$$\left|\delta(z, a, A)\right| + \left|\delta(z, \varepsilon, A)\right| \leq 1\,.$$

Ein PDAE ist also ein DPDA, wenn für alle Konfigurationen gilt

$$\left.\begin{array}{l}(z, w, \gamma) \vdash (z_1, w_1, \gamma_1) \text{ und} \\ (z, w, \gamma) \vdash (z_2, w_2, \gamma_2)\end{array}\right\} \implies (z_1, w_1, \gamma_1) = (z_2, w_2, \gamma_2)$$

Definition

Eine Sprache L ist deterministisch kontextfrei, wenn es einen deterministischen Kellerautomaten M gibt mit $L(M) = L$.

Ziel

Ist $L \subseteq \Sigma^*$ deterministisch kontextfrei, so auch $\Sigma^* \setminus L$.

Beweisidee: vertausche die akzeptierenden und die nichtakzeptierenden Zustände.

Problem: es kann Wörter w geben, die vom DPDA M nicht vollständig gelesen werden. Diese würden auch vom „Komplementärautomaten“ M' nicht bis zum Ende gelesen werden. Also $w \notin L(M)$ und $w \notin L(M')$, d.h. $L(M')$ ist nicht das Komplement von $L(M)$.
Dies kann aus drei Gründen geschehen:

- Der Keller ist leer, bevor das Eingabewort vollständig gelesen wurde.
- Es gibt keine passende Anweisung (d.h. M gerät in eine Konfiguration $(z, aw', A\gamma)$ mit $\delta(z, a, A) = \delta(z, \varepsilon, A) = \emptyset$).
- M gerät in eine endlose Folge von ε-Transitionen, ohne daß das Wort bereits zu Ende gelesen wurde.

vorläufiges Ziel

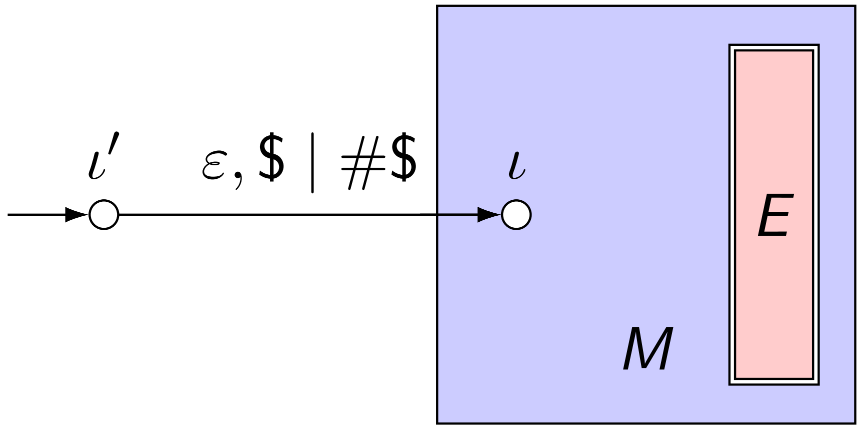
Konstruiere aus DPDA M einen äquivalenten DPDA M', so daß

1. M' den Keller niemals leert,
2. M' niemals blockiert, d.h. für alle $(z, a, A) \in Z \times \Sigma \times \Gamma$ gilt $\delta(z, a, A) \cup \delta(z, \varepsilon, A) \neq \emptyset$ und
3. M' keine endlosen Folgen von ε-Transitionen erlaubt.

Lemma

Sei $M = (Z, \Sigma, \Gamma, \iota, \delta, \#, E)$ ein DPDA. Dann existiert ein DPDA M_1 mit $L(M) = L(M_1)$, so daß M_1 den Keller nie vollständig leert.

Beweis: Füge neues Kellerinitialisierungssymbol ein, wird dieses gesehen, so blockiere:



Wir setzen also $M_1 = \big(Z \cup \{\iota'\}, \Sigma, \Gamma \cup \{\$\}, \iota', \delta_1, \$, E\big)$ mit

$$\delta_1(z,a,A) = \begin{cases} \delta(z,a,A) & \text{falls } z \in Z, a \in \Sigma \cup \{\varepsilon\}, A \in \Gamma \\ \{(\iota, \#\$)\} & \text{falls } z = \iota', a = \varepsilon, A = \$ \\ \emptyset & \text{sonst} \end{cases}$$

Dann gelten (Beweis per Induktion über die Länge der Berechnung):

- $(\iota', w, \$) \vdash_{M_1}^{*} (z, w', \gamma) \implies \gamma \in \Gamma^*\$$
  insbes. hat M_1 immer ein nichtleeres Wort aus $\Gamma^*\$$ auf dem Keller
- $(\iota, w, \#) \vdash_{M}^{*} (z, w', \gamma) \iff (\iota', w, \$) \vdash_{M_1}^{+} (z, w', \gamma\$)$
  d.h. M_1 simuliert M

Also: M_1 ist DPDA mit $L(M) = L(M_1)$, der den Keller nie leert. $\square$

Lemma

Zu jedem DPDA M existiert ein DPDA M' mit $L(M) = L(M')$, so daß M' jedes Wort w bis zum Ende liest.

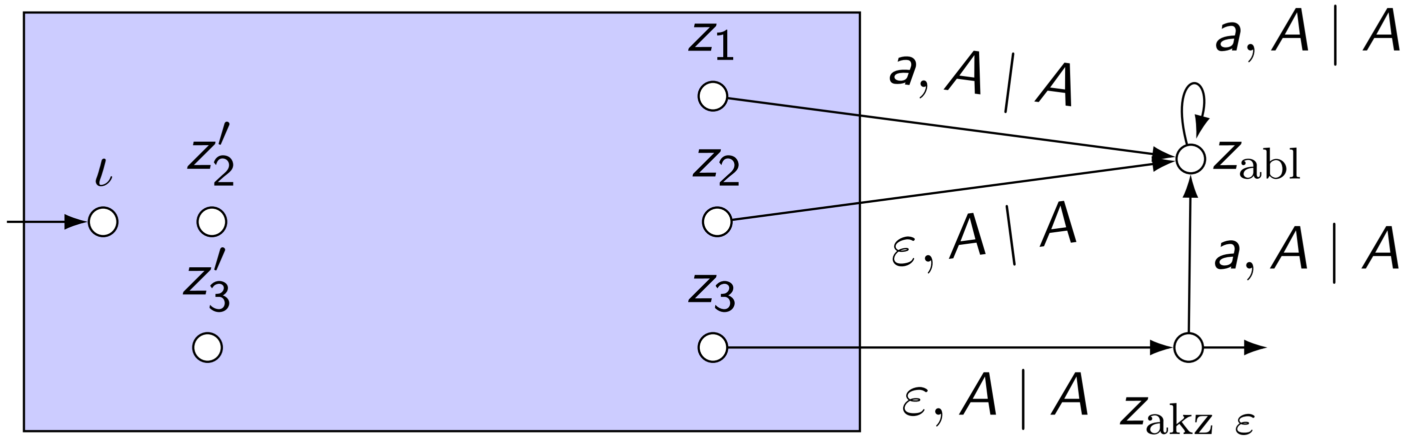
Beweis: Sei $M = (Z, \Sigma, \Gamma, \iota, \delta, \#, E)$. O.E. leert M den Keller nie.

Ist $w \in \Sigma^*$, so kann es zwei Gründe geben, warum M dieses Wort nicht bis zum Ende liest:

- Es gibt keine passende Anweisung (d.h. M gerät in eine Konfiguration $(z, aw', A\gamma)$ mit $\delta(z, a, A) = \delta(z, \varepsilon, A) = \emptyset$).
- M gerät in eine endlose Folge von ε-Transitionen, ohne daß das Wort bereits zu Ende gelesen wurde.

Lösung: Füge neue Zustände z_{abl} und $z_{\mathrm{akz}\ \varepsilon}$ hinzu.

- Wenn sich neuer DPDA M' in Zustand z_{abl} befindet, so liest er Wort zu Ende und akzeptiert nicht.
- Wenn sich M' in Zustand $z_{\mathrm{akz}\ \varepsilon}$ befindet, so liest er Wort zu Ende. Ist das restliche Wort ε, so akzeptiert er, ist es nicht ε, so wechselt er in z_{abl} (und akzeptiert also nicht).



- $\delta(z_1, \varepsilon, A) = \delta(z_1, a, A) = \emptyset$
- Die Konfiguration (z_2, ε, A) ist Beginn einer unendlichen Berechnung, die nie einen Endzustand aus E trifft.
- Die Konfiguration (z_3, ε, A) ist Beginn einer unendlichen Berechnung, die einen Endzustand aus E enthält.

Der neue DPDA M' wechselt wie folgt in einen der neuen Zustände $z_{\mathrm{akz}\ \varepsilon}$ bzw. z_{abl}:

- $\delta'(z, a, A) = \{(z_{\mathrm{abl}}, A)\}$ falls $\delta(z, a, A) = \delta(z, \varepsilon, A) = \emptyset$
- $\delta'(z, \varepsilon, A) = \{(z_{\mathrm{akz}\ \varepsilon}, A)\}$ falls es unendliche Folge von Konfigurationen $(z_i, \varepsilon, \gamma_i)$ gibt mit
  - $(z, \varepsilon, A) = (z_1, \varepsilon, \gamma_1)$,
  - $(z_i, \varepsilon, \gamma_i) \vdash (z_{i+1}, \varepsilon, \gamma_{i+1})$ für alle $i \geq 1$ und
  - $z_i \in E$ für ein $i \geq 1$.
- $\delta'(z, \varepsilon, A) = \{(z_{\mathrm{abl}}, A)\}$ falls es unendliche Folge von Konfigurationen $(z_i, \varepsilon, \gamma_i)$ gibt mit
  - $(z, \varepsilon, A) = (z_1, \varepsilon, \gamma_1)$,
  - $(z_i, \varepsilon, \gamma_i) \vdash (z_{i+1}, \varepsilon, \gamma_{i+1})$ für alle $i \geq 1$ und
  - $z_i \notin E$ für alle $i \geq 1$.

Dann liest M' jedes Wort w bis zum Ende und akzeptiert genau dann, wenn M akzeptiert. $\square$

Bemerkung

Der Beweis zeigt die Existenz von M'. Ob wir $\delta'(z, \varepsilon, A) = \{(z_{\mathrm{akz}\ \varepsilon}, A)\}$ oder $\delta'(z, \varepsilon, A) = \{(z_{\mathrm{abl}}, A)\}$ setzen müssen, haben wir nicht algorithmisch festgestellt.

Ein vollständiger Beweis incl. einer algorithmischen Konstruktion von M' steht im Buch Dexter Kozen: Automata and Computability. Springer 1997, Seiten 176 ff. (das ich uneingeschränkt empfehle).

Jedenfalls ist damit das vorläufige Ziel von Folie 14.16 erreicht, wir haben einen äquivalenten DPDA konstruiert, der den Keller nie leert, seine Eingabe vollständig liest und keine unendlichen Folgen von ε-Transitionen hat.

Satz

Ist $L \subseteq \Sigma^*$ deterministisch kontextfrei, so auch $\Sigma^* \setminus L$.

Beweis: Sei $M = (Z, \Sigma, \Gamma, \iota, \delta, \#, E)$ DPDA mit $L(M) = L$. O.E. liest M jedes Wort bis zum Ende (insbes. gibt es keine unendlichen Folgen von ε-Transitionen). Dann kann es $w \in \Sigma^*$, $e \in E$ und $z \in Z \setminus E$ geben mit

$$(\iota, w, \#) \vdash^* (e, \varepsilon, \gamma) \vdash^+ (z, \varepsilon, \gamma')\,.$$

Setzt man $M' = (Z, \Sigma, \Gamma, \iota, \delta, \#, Z \setminus E)$, so gilt also

$$w \in L(M) \cap L(M')\,,$$

d.h. $L(M')$ ist nicht das Komplement von $L(M)$.

Lösung: M' merkt sich, ob während der letzten Folge von ε-Transitionen nur Zustände aus $Z \setminus E$ gesehen wurden.

Definiere zunächst eine Funktion $f\colon Z \to \{1,2\}$ gemäß

$$f(z) = \begin{cases} 1 & \text{falls } z \notin E \\ 2 & \text{falls } z \in E \end{cases}$$

Wir konstruieren DPDA M' mit Zustandsmenge $Z' = Z \times \{1,2,3,4\}$ so, daß gilt: Wenn

$$(z_0, w, \gamma_0) \vdash_M (z_1, w, \gamma_1) \vdash_M (z_2, w, \gamma_2) \vdash_M \cdots \vdash_M (z_n, w, \gamma_n) \qquad (2)$$

maximale Berechnung von M aus ε-Transitionen ist, dann erhalten wir eine maximale Berechnung von M' aus ε-Transitionen

$$\begin{aligned} \big((z_0, i_0), w, \gamma_0\big) \quad &\vdash_{M'} \quad \big((z_1, i_1), w, \gamma_1\big) \vdash_{M'} \big((z_2, i_2), w, \gamma_2\big) \vdash_{M'} \cdots \\ &\vdash_{M'} \quad \big((z_n, i_n), w, \gamma_n\big) \\ &\vdash_{M'} \quad \big((z_n, i_n + 2), w, \gamma_n\big) \end{aligned}$$

mit $i_j = \max\big\{f(z_0), f(z_1), \cdots, f(z_j)\big\}$ für alle $1 \leq j \leq n$.

Definiere hierfür die Überführungsfunktion
$\delta'\colon Z' \times \big(\Sigma \cup \{\varepsilon\}\big) \times \Gamma \to \mathcal{P}_e(Z' \times \Gamma^*)$ wie folgt:

$$\delta'\big((z,i),\varepsilon,A\big) = \begin{cases} \Big\{\big((z',j),\gamma\big)\Big\} & \text{falls } i \leq 2, j = \max\big\{i, f(z')\big\} \\ & \text{und } \delta(z,\varepsilon,A) = \big\{(z',\gamma)\big\} \\ \Big\{\big((z,i+2),A\big)\Big\} & \text{falls } i \leq 2, \delta(z,\varepsilon,A) = \emptyset \\ \emptyset & \text{falls } i > 2 \end{cases}$$

$$\delta'\big((z,i),a,A\big) = \begin{cases} \Big\{\Big(\big(z', f(z')\big),\gamma\Big)\Big\} & \text{falls } i > 2, \delta(z,a,A) = \big\{(z',\gamma)\big\} \\ \emptyset & \text{sonst} \end{cases}$$

Dann ist $M' = \Big(Z', \Sigma, \Gamma, \big(\iota, f(\iota)\big), \#, Z \times \{3\}\Big)$ ein DPDA.

Die maximalen Berechnungen

$$(\iota, w, \#) \vdash_M^* (z, \varepsilon, \gamma)$$

entsprechen den maximalen Berechnungen

$$\Big((\iota, f(\iota)), w, \#\Big) \vdash_{M'}^* \big((z_n, i), \varepsilon, \gamma_n\big) \,.$$

Diese erfüllen $i \in \{3, 4\}$ mit $i = 3$ gdw. wenn unter den letzten ε-Transitionen kein Zustand aus E war.

Daher gilt $\Sigma^* \setminus L(M) = L(M')$. $\square$

Der Beweis zeigt die Existenz von M' (da wir nur die Existenz des DPDA, der jedes Wort bis zum Ende liest, gezeigt haben).

Da aber dieser DPDA algorithmisch konstruiert werden kann (siehe z.B. Kozens Buch), gilt sogar:

Satz

Aus einem DPDA M kann ein DPDA M' berechnet werden mit $L(M') = \Sigma^* \setminus L(M)$.

Zusammenfassung 14. Vorlesung

in dieser Vorlesung neu

- der Schnitt einer kontextfreien und einer regulären Sprache ist kontextfrei
- Wird L von einem deterministischen PDA akzeptiert, so auch das Komplement

kommende Vorlesung

- Algorithmen (vgl. Vorlesung 8)
- Fragestunde