Klausuranmeldung bis 8. 2. 2013

Satz Eine Sprache L ist regulär gdw. der Index von $\sim_L$ endlich ist.

$x \sim_M y$ gdw $\left(\hat{\delta}(z_0, x) = \hat{\delta}(z_0, y) \right)$

$x \sim_L y$ gdw $\forall z \in \Sigma^* \, (xz \in L \text{ gdw. } yz \in L)$

$$\boxed{x \sim_M y} \Rightarrow \boxed{x \sim_L y} \Rightarrow \text{Index}(\sim_L) < \infty$$

Anwendungen:

1) Beweis der Nicht-Regularität von Sprachen
2) Konstruktion von Minimalautomaten

Bsp $L = \{ a^n b^m c^m \mid n, m \geq 0 \} \cup \{ b, c \}^*$

$[ab^2c] = [ab^3c^2] = \{ a^i b^j c^{j-1} \mid i, j \geq 2 \} = \{ w \mid wc \in L, x \in \Sigma^*, x \neq c, wx \notin L \}$

$[ab^3c] = \{ a^i b^j c^{j-2} \mid i \geq 1, j \geq 3 \}$

$[ab^4c] = \ldots$

$\vdots$

$[ab^4c] = \ldots$

$\vdots$

$\Rightarrow$ d.h. der Index von $\sim_L$ muss unendlich sein.

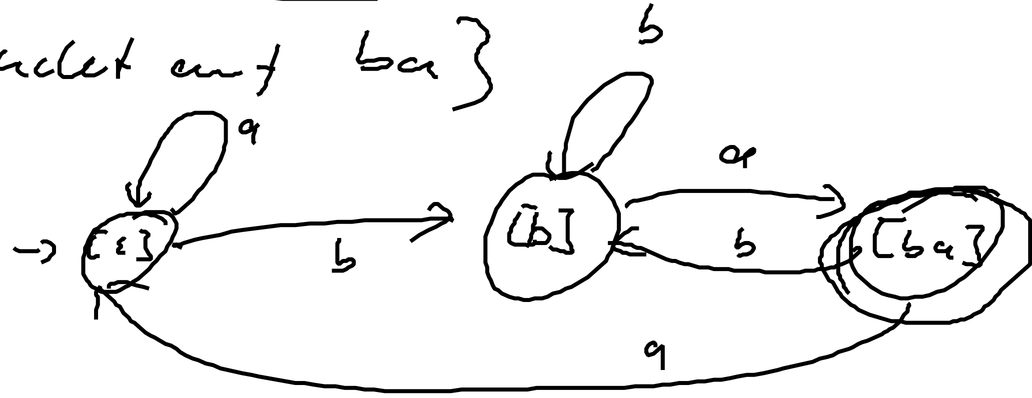
Bsp $L = \{x \in \{a,b\}^* \mid x \text{ endet auf } ba\}$

$\underline{[\varepsilon] = [a]} = [aa] = \dots = \{x \mid x \text{ endet nicht auf } b \text{ und nicht auf } ba\}$

$[b] = [bb] = [ab] = \dots = \underline{\{x \mid x \text{ endet auf } b\}}$

$[ba] = [aba] = [bba] = \dots = \{x \mid x \text{ endet auf } ba\}$

$\Sigma^* = [\varepsilon] \cup [b] \cup [ba]$



$M = (Z, \Sigma, z_0, \delta, E)$

Z = Menge der Äquivalenzklassen $= \Sigma^* / \sim_L$

$z_0 = [\varepsilon]$

$E = \{[x] \mid x \in L\}$

$\delta([\varepsilon], a) = [\overset{\varepsilon}{\cancel{a}}]$

$\delta([\varepsilon], b) = [b]$

$\cancel{\delta([a], a) = [a]}$

$\cancel{\delta([a], b) = [b]}$

$\delta([b], a) = [ba]$

$\underline{\delta([b], b) = [b]}$

$\delta([ba], a) = [\varepsilon]$

$\delta([ba], b) = [b]$

Bem: Der Äquivalenzklassenautomat ist der Minimalautomat (geringste Anzahl von Zuständen)

$$\text{Index}(\sim_L) \le \text{Index}(\sim_M)$$

$\text{Index}(\sim_L)$ ‖ Anzahl der Zustände im Äquivalenzklassenautomat

Alle Automaten mit minimaler Anzahl der Zustände sind strukturell äquivalent. – im Kontext von DFAs

Wann ist ein Automat nicht minimal?

→ Es ex $z, z' \in Z$ mit $z \neq z'$ und $\hat{\delta}(z,x) \in E$ gdw $\hat{\delta}(z',x) \in E$ für alle $x \in \Sigma^*$.

⇒ Wir können diese Zustände verschmelzen.

Algorithmus (Minimalautomaten)

Eingabe: DFA $M = (Z, \Sigma, \delta, z_0, E)$

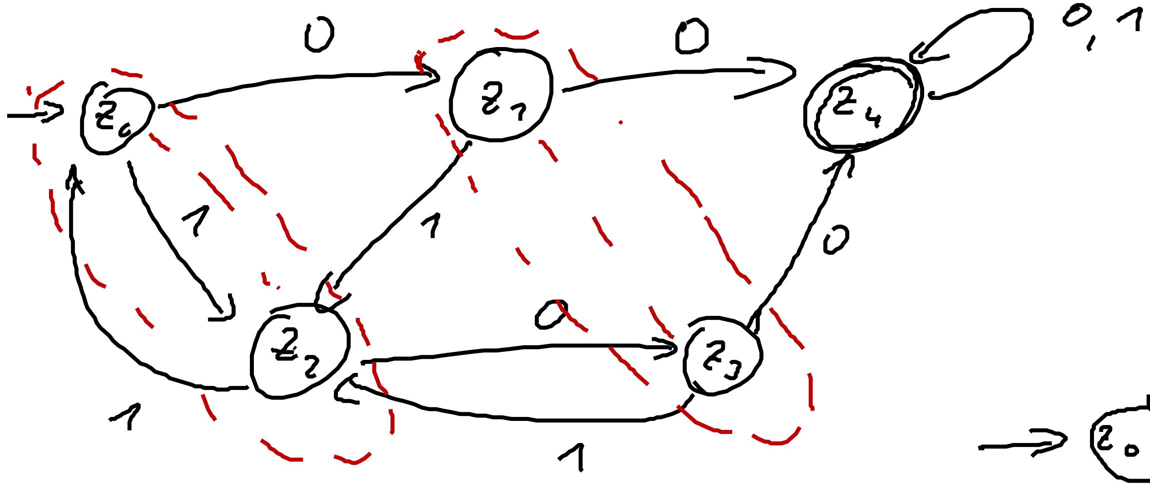
Ausgabe: Paare von zu verschmelzenden Zuständen

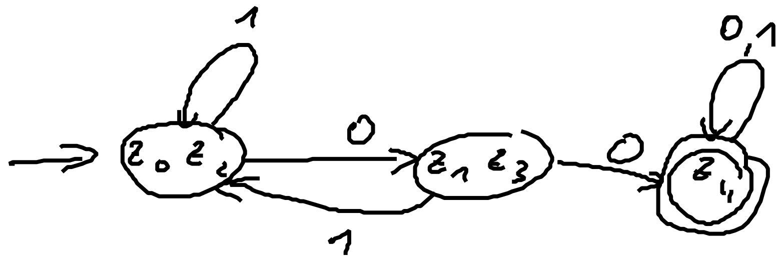
1. Konstruiere Tabelle mit Zustandspaaren $\{z, z'\}$ für $z \neq z'$.
2. Markiere $\{z, z'\}$ falls $z \in E$ und $z' \notin E$ ⇐
3. Für unmarkierte Paare $\{z, z'\}$ und $a \in \Sigma$ teste ob $\{\delta(z,a), \delta(z',a)\}$ markiert ist? ⇐

   ja → markiere
   nein → tue nix
4. Wiederhole 3 für alle Zustandspaare und Eingabezeichen ⇐ bis sich nichts mehr ändert.

⇒ alle unmarkierten Paare sind verschmelzbar.

Bsp





	z_0	z_1	z_2	z_3	z_4
z_0					
z_1	×				
z_2		×			
z_3	×	.	×		
z_4	×	×	×	×	

1.6.6 Abschlusseigenschaften

<u>Def.</u> Sei X eine beliebige Menge und $M \subseteq X$.
Sei op eine binäre (oder unäre) Operation auf X. Dann heißt M abgeschlossen unter op, falls

$$\forall x,y \in M : (x \text{ op } y) \in M$$

(unär: $\forall x : \text{op } x \in M$).

<u>Satz</u> Die regulären Sprachen sind abgeschlossen unter:

- Vereinigung
- Schnitt
- Komplement
- Produkt
- Stern.

<u>Bew</u>: Abschluss unter <u>Vereinigung</u>, <u>Produkt</u> und <u>Stern</u> wg. reg. Ausdrücke.

Komplement:

Gegeben ein DFA $M = (Z, \Sigma, \delta, z_0, E)$, wir konstr. M' mit $T(M') = \overline{T(M)}$:

$$M' = (Z, \Sigma, \delta, z_0, Z - E)$$

gilt nur für DFA!

Schnitt:

Für zwei reg. Spr. L_1 und L_2: $\underline{\underline{L_1 \cap L_2}} = \overline{\overline{L_1 \cap L_2}} = \overline{\overline{L_1} \cup \overline{L_2}}$

Alternativ: Kreuzproduktautomat

Gegeben DFAs $M_1 = (Z_1, \Sigma, \delta_1, z_{01}, E_1)$, $M_2 = (Z_2, \Sigma, \delta_2, z_{02}, E_2)$

Konstr.: $M = (Z_1 \times Z_2, \Sigma, \delta, (z_{01}, z_{02}), E_1 \times E_2)$

$$\delta((z, z'), a) = (\delta_1(z, a), \delta_2(z', a))$$

Offensichtlich: $T(M) = T(M_1) \cap T(M_2)$
