

$$L_n = \{ u1v : u, v \in \{0, 1\}^*, |v| = n-1 \}$$

“das  $n$ -letzte Bit ist eine 1”

Es gibt einen NFA für  $L_n$  mit  $n+1$  Zuständen.

Aber: jeder totale DFA für  $L_n$  hat  $\geq 2^n$  Zustände.

$$\text{Nerode-Index} = |\{0, 1\}^*/L_n| \geq 2^n$$



Anzahl an Zuständen im Minimalautomaten für  $L_n$   
 $\leq$  Anzahl an Zuständen in jedem totalen DFA für  $L_n$

$$L_n = \{ u1v : u, v \in \{0, 1\}^*, |v| = n-1 \}$$

“das  $n$ -letzte Bit ist eine 1”

Behauptung:  $|\{0, 1\}^*/L_n| \geq 2^n$

Für alle  $x, y \in \{0, 1\}^n$  mit  $x \neq y$  gilt:  $x \not\sim_{L_n} y$

sei  $x = a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n$

$y = b_1 \dots b_{i-1} 0 b_{i+1} \dots b_n$

Dann gilt:  $x1^{i-1} \in L_n$ , aber  $y1^{i-1} \notin L_n$

Daher gilt: es gibt mindestens  $2^n \sim_{L_n}$ -Äquivalenzklassen

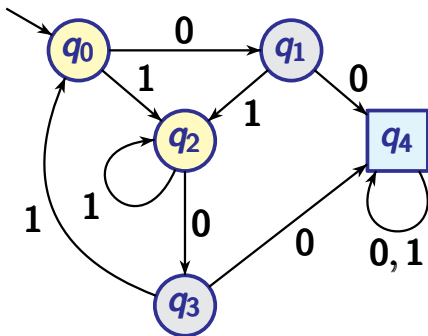
Sei  $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$  ein DFA, wobei  $\delta$  total (ohne unerreichbare Zustände)

Äquivalenzrelation  $\equiv$  auf dem Zustandsraum  $Q$ :

$$\begin{aligned} q \equiv p \quad \text{gdw} \quad & \text{für alle Wörter } z \in \Sigma^* \text{ gilt:} \\ & \delta(q, z) \in F \Leftrightarrow \delta(p, z) \in F \\ \text{gdw} \quad & \mathcal{L}(q) = \mathcal{L}(p) \end{aligned}$$

für Zustand  $s \in Q$ :

$$\mathcal{L}(s) \stackrel{\text{def}}{=} \mathcal{L}(\mathcal{M}^s), \quad \text{wobei } \mathcal{M}^s = (Q, \Sigma, \delta, s, F)$$



Äquivalenz  $\equiv$  mit  
drei Äquivalenzklassen

$\{q_0, q_2\}$

$\{q_1, q_3\}$

$\{q_4\}$

$$\mathcal{L}(q_4) = \{0, 1\}^*$$

$$\mathcal{L}(q_1) = \mathcal{L}(q_3) \hat{=} (11^*0)^*0(0+1)^*$$

$$\mathcal{L}(q_0) = \mathcal{L}(q_2) \hat{=} 1^*0(11^*0)^*0(0+1)^*$$

Sei  $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$  ein totaler DFA.

Äquivalenzrelation  $\equiv$  auf dem Zustandsraum  $Q$ :

$q \equiv p$  gdw für alle Wörter  $z \in \Sigma^*$  gilt:

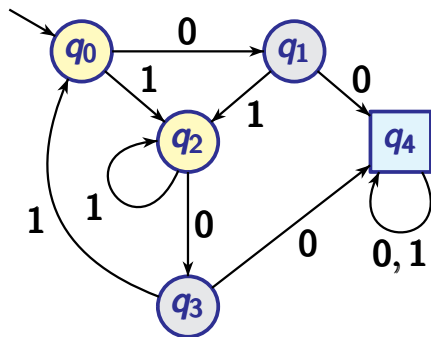
$$\delta(q, z) \in F \Leftrightarrow \delta(p, z) \in F$$

Quotienten-DFA

$$\mathcal{M}/\equiv \stackrel{\text{def}}{=} (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F')$$

wobei  $F' = \{ [q]_{\equiv} : q \in F \}$

$$\delta'([q]_{\equiv}, a) = [\delta(q, a)]_{\equiv}$$

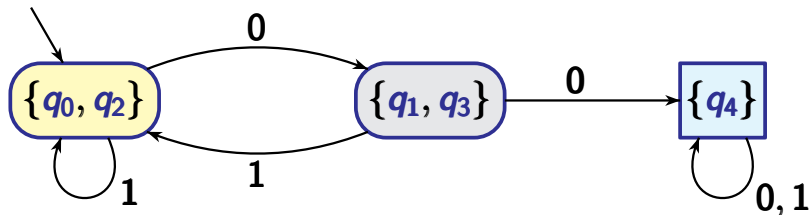


Äquivalenz  $\equiv$  mit  
drei Äquivalenzklassen

$\{q_0, q_2\}$

$\{q_1, q_3\}$

$\{q_4\}$

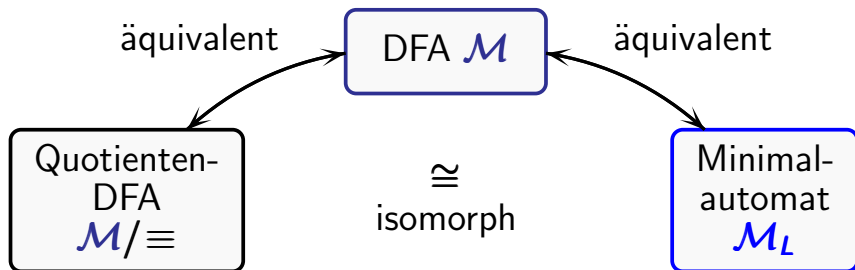


Sei  $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$  ein DFA und  $L = \mathcal{L}(\mathcal{M})$ .

(1)  $\mathcal{M}$  und  $\mathcal{M}/\equiv$  sind äquivalent, d.h.,

$$\mathcal{L}(\mathcal{M}/\equiv) = \mathcal{L}(\mathcal{M})$$

(2) Ist  $\delta$  total und sind alle Zustände  $q \in Q$  von  $q_0$  erreichbar, so sind  $\mathcal{M}/\equiv$  und  $\mathcal{M}_L$  isomorph.



Sei  $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$  ein totaler DFA.

Behauptung:  $\mathcal{L}(\mathcal{M}/\equiv) = \mathcal{L}(\mathcal{M})$

Beweis: zeige zunächst, dass folgende Aussagen gelten:

- (i) für alle  $p \in Q$  gilt:  $p \in F$  gdw  $[p]_{\equiv} \in F'$
- (ii) Ist  $q_0 q_1 \dots q_n$  der Lauf für  $z \in \Sigma^*$  in  $\mathcal{M}$ , so ist

$$[q_0]_{\equiv} [q_1]_{\equiv} \dots [q_n]_{\equiv}$$

der Lauf für  $z$  in  $\mathcal{M}/\equiv$ .

$$\begin{aligned} z \in \mathcal{L}(\mathcal{M}) &\iff q_n \in F \\ &\stackrel{(i)}{\iff} [q_n]_{\equiv} \in F' \stackrel{(ii)}{\iff} z \in \mathcal{L}(\mathcal{M}/\equiv) \end{aligned}$$

gegeben: totaler DFA  $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$

$\mathcal{R}_0 := \{ \langle p, q \rangle : \{p, q\} \subseteq F \text{ oder } \{p, q\} \cap F = \emptyset \};$

$i := 0;$

REPEAT

IF es gibt ein Zustandspaar  $\langle p, q \rangle \in \mathcal{R}_i$  und  $a \in \Sigma$   
mit  $\langle \delta(p, a), \delta(q, a) \rangle \notin \mathcal{R}_i$

THEN wähle solches Zustandspaar  $\langle p, q \rangle \in \mathcal{R}_i$  ;

$\mathcal{R}_{i+1} := \mathcal{R}_i \setminus \{ \langle p, q \rangle \}; \quad i := i+1$

FI

UNTIL  $\mathcal{R}_i = \mathcal{R}_{i-1} \leftarrow p \equiv q$  für alle  $\langle p, q \rangle \in \mathcal{R}_i$

## Table filling algorithm (Pseudo-Code)

2371

gegeben: DFA  $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$  mit  $Q = \{q_0, \dots, q_n\}$

---

Erstelle eine Tabelle für alle ungeordneten Paare  $\langle q_i, q_k \rangle$ .

Markiere alle Paare  $\langle q_i, q_k \rangle$  mit  $\{q_i, q_k\} \cap F \neq \emptyset$   
und  $\{q_i, q_k\} \setminus F \neq \emptyset$

WHILE es gibt ein unmarkiertes Paar  $\langle q_i, q_k \rangle$  und  $a \in \Sigma$ ,  
so dass  $\langle \delta(q_i, a), \delta(q_k, a) \rangle$  markiert ist

DO

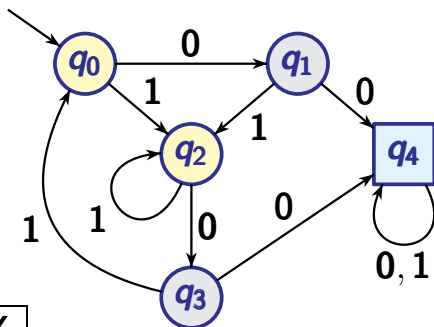
wähle ein solches Paar  $\langle q_i, q_k \rangle$  und markiere es

OD

Bilde maximale Mengen von paarweise unmarkierten  
Zuständen.

# Beispiel: table filling algorithm

2376



$q_1$	X			
$q_2$		X		
$q_3$	X		X	
$q_4$	X	X	X	X
	$q_0$	$q_1$	$q_2$	$q_3$

keine weiteren  
Markierungen möglich

$$q_0 \equiv q_2$$

$$q_1 \equiv q_3$$

Charakterisierungen regulärer Sprachen (Typ 3):

Sei  $L \subseteq \Sigma^*$  eine Sprache. Dann gilt:

$L$  ist regulär

gdw  $L = \mathcal{L}(G)$  für eine reguläre Grammatik  $G$

gdw  $L = \mathcal{L}(M)$  für einen NFA  $M$

gdw  $L = \mathcal{L}(M)$  für einen DFA  $M$

gdw  $L = \mathcal{L}(\alpha)$  für einen regulären Ausdruck  $\alpha$

gdw der Nerode-Index von  $L$  ist endlich

Die Größe einer Grammatik  $G$  ist definiert durch:

$$|G| = \# \text{ Variablen} + \sum \text{ Länge der Regeln}$$

Beispiel: CFG  $G$  mit den Variablen  $S, A, B$   
und den Regeln

$S \rightarrow \varepsilon$		$aAc$	← Länge 5
$A \rightarrow aAc$		$B$	← Länge 6
$B \rightarrow b$			← Länge 2

hat die Größe  $3 + 13 = 16$

## Zur Erinnerung: CFG $\rightsquigarrow$ $\varepsilon$ -freie CFG

35A

gegeben: CFG  $G = (V, \Sigma, \mathcal{P}, S)$  mit  $\varepsilon \notin \mathcal{L}(G)$

gesucht: äquivalente  $\varepsilon$ -freie CFG

1. berechne  $V_\varepsilon = \{A \in V : A \Rightarrow^* \varepsilon\}$
2. entferne alle  $\varepsilon$ -Regeln
3. solange es eine Regel  $B \rightarrow xAy$  gibt, so dass

$$A \in V_\varepsilon, |xy| \geq 1, B \not\rightarrow xy,$$

wähle eine solche und füge die Regel  $B \rightarrow xy$  ein



exponentielles Wachstum

der Größe der Grammatik ist möglich