Minimierung von DFAs

Für jeden DFA gibt es einen eindeutigen, äquivalenten DFA mit minimaler Anzahl an Zuständen. (Für NFAs leider nicht)

Äquivalenz von Zuständen:

Sei A ein DFA. Zwei Zustände sind äquivalent bezüglich A, falls beide dieselbe Sprache akzeptieren. D.h.:

$$p \equiv_A q \iff \left(\forall w \in \Sigma^*: \hat{\delta}(p,w) \in F \iff \hat{\delta}(q,w) \in F\right)$$

Daraus folgt:

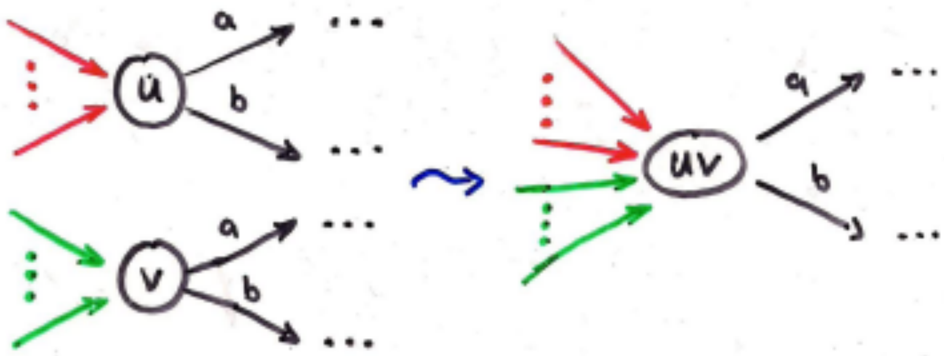
$$p \equiv_A q \iff \left(\forall a \in \Sigma: \delta(p,a) \equiv_A \delta(q,a)\right)$$

Und natürlich auch:

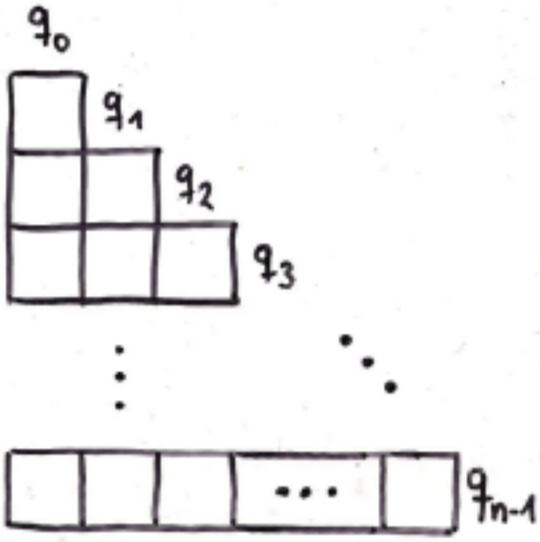
$$p \not\equiv_A q \iff \left(\exists a \in \Sigma: \delta(p,a) \not\equiv_A \delta(q,a)\right) \qquad (*)$$

Idee des Algorithmus:

1) Finde alle äquivalente Zustände. (Mit (*))
2) Sind u und v äquivalent, dann akzeptiert der DFA von u aus und von v aus genau dieselbe Sprache.
3) Kollabiere u und v



Algorithmus:



1. Markiere alle Zellen, in denen ein Zustand Endzustand ist und der andere nicht, mit dem leeren Wort ε.
2. Bis in einer Runde keine Zelle markiert wurde, wiederhole:
3. Für alle $q_i, q_j \in Q$ und $a \in \Sigma$: markiere $\{q_i, q_j\}$ mit $a\omega$, falls $\{\delta(q_i,a), \delta(q_j,a)\}$ mit ω markiert ist.
4. Ist $\{q_i, q_j\}$ unmarkiert so sind q_i und q_j äquivalent. Kollabiere nun alle äquivalente Zustände.

Beispiel:

A:

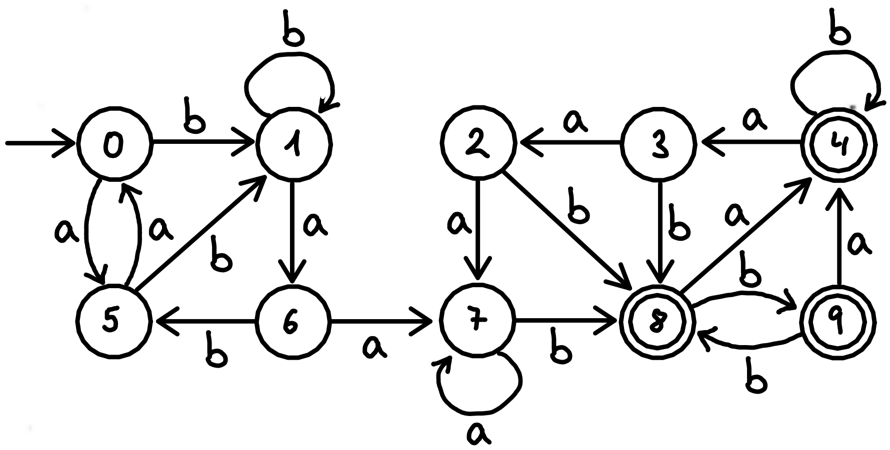


Tabelle:

	0	1	2	3	4	5	6	7	8
1	aab								
2	b	b							
3	b	b	$\equiv_A$						
4	ε	ε	ε	ε					
5	$\equiv_A$	aab	b	b	ε				
6	ab	ab	b	b	ε	ab			
7	b	b	$\equiv_A$	$\equiv_A$	ε	b	b		
8	ε	ε	ε	ε	a	ε	ε	ε	
9	ε	ε	ε	ε	a	ε	ε	ε	$\equiv_A$

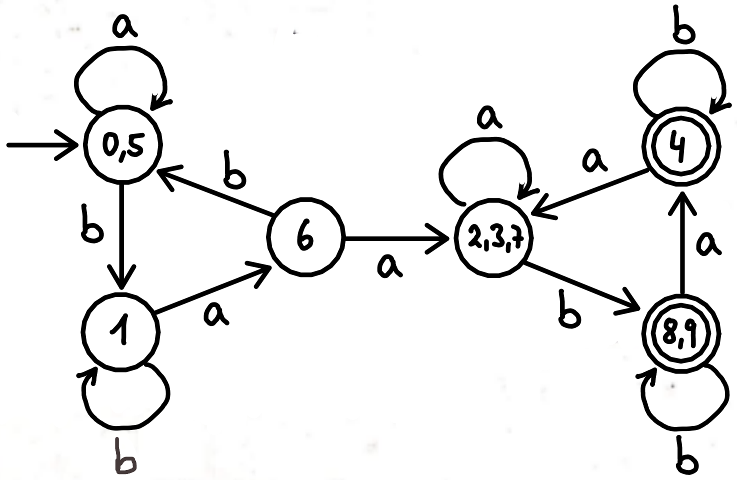
$2 \equiv_A 3$

$5 \not\equiv_A 6$, da $\hat{\delta}(5, ab) \notin F$, aber $\hat{\delta}(6, ab) \in F$.

$\rightsquigarrow$ Äquivalenzklassen sind: $\underbrace{\{0,5\}}_{[0]_\equiv}, \underbrace{\{1\}}_{[1]_\equiv}, \underbrace{\{2,3,7\}}_{[2]_\equiv}, \underbrace{\{6\}}_{[6]_\equiv}, \underbrace{\{4\}}_{[4]_\equiv}, \underbrace{\{8,9\}}_{[8]_\equiv}$

Kollabierter Automat:

$A/\equiv$:



„Quotienten-automat"

Formal:

$$A/\equiv \; = (Q/\equiv, \Sigma, \delta', [0]_\equiv, F/\equiv) \quad \text{mit:}$$

- $Q/\equiv \; = \{[0]_\equiv, [1]_\equiv, [2]_\equiv, [4]_\equiv, [6]_\equiv, [8]_\equiv\}$
- $\Sigma = \{a, b\}$
- δ' wie auf der Zeichnung
- $F/\equiv \; = \{[4]_\equiv, [8]_\equiv\}$

Wie kommt man aber auf den minimalen DFA für L wenn man nur L hat aber keinen Ausgangs-DFA?
Gruppiere nicht äquivalente Zustände, sondern äquivalente Wörter!

Äquivalenz von Wörtern:

Zwei Wörter sind äquivalent, falls man an beide genau dieselben Suffixe anhängen kann, sodass die entstehenden Wörter in der Sprache enthalten sind:

„Nerode-Relation"

$$u \equiv_L v \iff (\forall w \in \Sigma^* : uw \in L \iff vw \in L)$$

$\equiv_L$ bildet (analog zu $\equiv_A$) eine Äquivalenzrelation. Gruppiert man alle Wörter die zueinander äquivalent sind so entsteht eine Partition von Σ^* in disjunkte Teilmengen $[u]_{\equiv_L}$ („Äquivalenzklassen").

beliebiger Repräsentant aus der Klasse

$[u]_{\equiv_L}$ enthält alle Wörter an denen man genau dieselben Suffixe anhängen kann.

Idee:

Benutze diese Äquivalenzklassen als Zustände. Dabei wird jeder Zustand von genau den Wörtern in seiner Äquivalenzklasse erreicht und von ihm aus werden genau die Wörter akzeptiert die in der Äquivalenzklasse als Suffix erlaubt sind.

Beispiel:

$L = \{a^n b^m \mid n,m \text{ gerade}\}$, $\Sigma = \{a,b\}$

$\rightsquigarrow$
$$\left.\begin{array}{lcl}
[\varepsilon]_{\equiv_L} & = & \{a^n \mid n \text{ gerade}\} \\
[a]_{\equiv_L} & = & \{a^n \mid n \text{ ungerade}\} \\
[b]_{\equiv_L} & = & \{a^n b^m \mid n \text{ gerade}, m \text{ ungerade}\} \\
[bb]_{\equiv_L} & = & \{a^n b^m \mid n \text{ gerade}, m \text{ gerade}, m > 0\} \\
[ab]_{\equiv_L} & & \text{enthält die restlichen Wörter aus } \Sigma^*
\end{array}\right\} \text{Partition von } \Sigma^* \text{ in 5 Klassen}$$

Mögliche Suffixe sind:

für $[\varepsilon]_{\equiv_L} \rightsquigarrow$ L selber

für $[a]_{\equiv_L} \rightsquigarrow \{a^n b^m \mid n \text{ ungerade}, m \text{ gerade}\}$

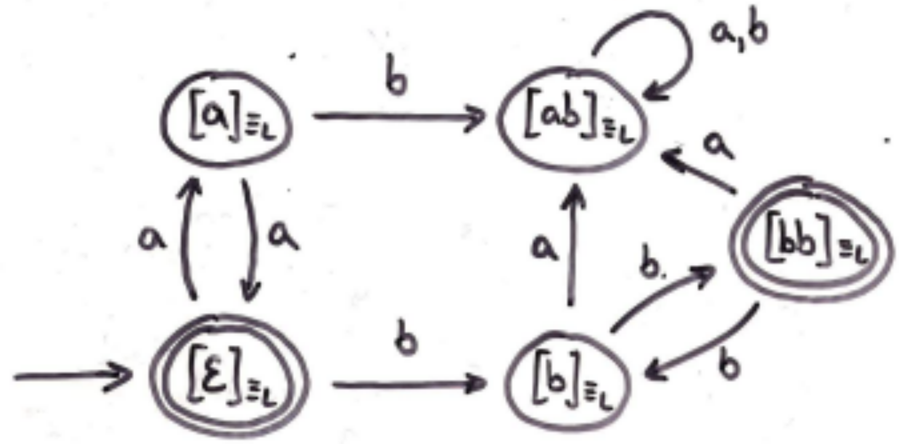
für $[b]_{\equiv_L} \rightsquigarrow \{b^m \mid m \text{ ungerade}\}$

für $[bb]_{\equiv_L} \rightsquigarrow \{b^m \mid m \text{ gerade}\}$

für $[ab]_{\equiv_L} \rightsquigarrow$ keine, also $\emptyset$

DFA: („Kanonischer Minimalautomat")

M_L :



Formal: $M_L = (\Sigma^*/\equiv_L, \Sigma, \delta_L, [\varepsilon]_{\equiv_L}, F_L)$ mit:

- $\Sigma^*/\equiv_L = \{[\varepsilon]_{\equiv_L}, [a]_{\equiv_L}, [b]_{\equiv_L}, [bb]_{\equiv_L}, [ab]_{\equiv_L}\}$
- $\Sigma = \{a,b\}$
- δ_L wie auf dem Bild
- $F_L = \{[\varepsilon]_{\equiv_L}, [bb]_{\equiv_L}\}$

Fazit:

Der Minimalautomat hat genau so viele Zustände wie es Äquivalenzklassen $[u]_{\equiv_L}$ gibt. Sind es aber unendlich viele dann ist der „kleinste" Automat unendlich groß
$\Rightarrow$ Die Sprache wäre nicht regulär! (Satz von Myhill-Nerode)

Beispiel:

$L = \{a^n b^n \mid n \in \mathbb{N}_0\}$, $\Sigma = \{a, b\}$

$\rightsquigarrow$

$[\varepsilon]_{\equiv_L} = \{\varepsilon\}$
$[a]_{\equiv_L} = \{a\}$
$[aa]_{\equiv_L} = \{aa\}$
$[aaa]_{\equiv_L} = \{aaa\}$
$\vdots$
$[ab]_{\equiv_L} = \{a^n b^n \mid n \in \mathbb{N}\}$
$[aab]_{\equiv_L} = \{a^{n+1} b^n \mid n \in \mathbb{N}\}$
$[aaab]_{\equiv_L} = \{a^{n+2} b^n \mid n \in \mathbb{N}\}$
$\vdots$
$[b]_{\equiv_L}$ enthält die restlichen Wörter aus Σ^*

Partition von Σ^* in unendlich viele Klassen.

$\Rightarrow$ L nicht regulär!

„Automat":