Technische Universität Dresden

Formale Systeme

8. Vorlesung: Minimale Automaten

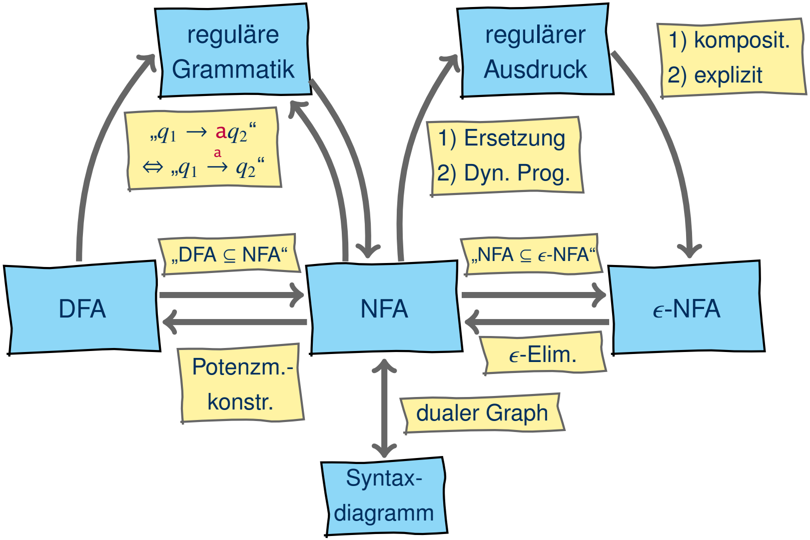
Markus Krötzsch

Professur für Wissensbasierte Systeme

TU Dresden, 10. November 2025

Rückblick

Darstellungen von Typ-3-Sprachen



Minimierung von Automaten

Automaten verkleinern

Wir haben bereits Methoden kennengelernt, um Automaten zu vereinfachen:

- Entfernen von Zuständen, die von keinem Anfangszustand aus erreichbar sind
- Entfernen von Zuständen, von denen aus kein Endzustand erreicht werden kann

Erhalten wir damit den kleinstmöglichen äquivalenten Automaten?

Automaten verkleinern

Wir haben bereits Methoden kennengelernt, um Automaten zu vereinfachen:

- Entfernen von Zuständen, die von keinem Anfangszustand aus erreichbar sind
- Entfernen von Zuständen, von denen aus kein Endzustand erreicht werden kann

Erhalten wir damit den kleinstmöglichen äquivalenten Automaten?

Nein – ein einfaches Gegenbeispiel:

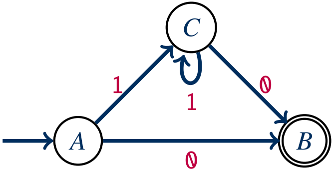
Beispiel: Sei $\mathcal{M}$ ein endlicher Automat, bei dem alle Zustände erreichbar sind und einen Endzustand erreichen können. Der Vereinigungsautomat[a] $\mathcal{M} \oplus \mathcal{M}$ akzeptiert die selbe Sprache, hat nur erreichbare Zustände, aber die doppelte Zustandszahl.

[a]Hierbei müssen die Zustände einer Kopie von $\mathcal{M}$ umbenannt werden.

Ein interessanteres Beispiel

Der Vereinigungsautomat ist immer ein NFA. Nichtdeterminismus macht es einfach, nichtminimale Automaten zu finden.

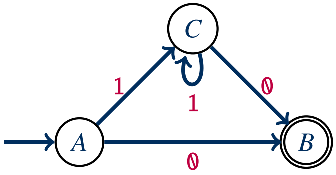
Interessanter sind nichtminimale DFAs:



Ein interessanteres Beispiel

Der Vereinigungsautomat ist immer ein NFA. Nichtdeterminismus macht es einfach, nichtminimale Automaten zu finden.

Interessanter sind nichtminimale DFAs:



Dieser DFA hat keine offensichtlich überflüssigen Zustände, aber der folgende kleinere DFA erkennt die selbe Sprache 1^*0:

Automaten minimieren?

Wie kann man Automaten weiter minimieren?

Beobachtungen:

- Zur Erkennung von Wörtern muss der Automat nur seinen aktuellen Zustand kennen
- Wichtig ist, wohin man vom aktuellen Zustand aus gelangt, wenn man das restliche Wort einliest
- Es ist nicht relevant, auf welchem Weg man zu diesem Zustand gelangt ist

Automaten minimieren?

Wie kann man Automaten weiter minimieren?

Beobachtungen:

- Zur Erkennung von Wörtern muss der Automat nur seinen aktuellen Zustand kennen
- Wichtig ist, wohin man vom aktuellen Zustand aus gelangt, wenn man das restliche Wort einliest
- Es ist nicht relevant, auf welchem Weg man zu diesem Zustand gelangt ist

Idee: Zwei Zustände sind gleichwertig, wenn man ausgehend von beiden Zuständen die selbe Sprache akzeptieren kann

$\leadsto$ Gleichwertige Zustände könnten verschmolzen werden ...

Äquivalenz von Zuständen

Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F\rangle$ und einen Zustand $q \in Q$ sei $\mathcal{M}_q = \langle Q, \Sigma, \delta, q, F\rangle$ der abgewandelte DFA mit Startzustand q.

Zwei Zustände $p, q \in Q$ sind $\mathcal{M}$-äquivalent, in Symbolen $p \sim_{\mathcal{M}} q$, wenn gilt:

$$\mathbf{L}(\mathcal{M}_p) = \mathbf{L}(\mathcal{M}_q)$$

das heißt wenn für jedes Wort $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \text{ genau dann wenn } \delta(q, w) \in F.$$

Wenn der Automat $\mathcal{M}$ klar ist, schreiben wir einfach $\sim$ statt $\sim_{\mathcal{M}}$.

Äquivalenz von Zuständen

Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle$ und einen Zustand $q \in Q$ sei $\mathcal{M}_q = \langle Q, \Sigma, \delta, q, F \rangle$ der abgewandelte DFA mit Startzustand q.

Zwei Zustände $p, q \in Q$ sind $\mathcal{M}$-äquivalent, in Symbolen $p \sim_{\mathcal{M}} q$, wenn gilt:

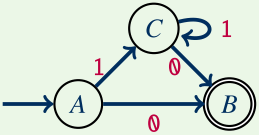
$$\mathbf{L}(\mathcal{M}_p) = \mathbf{L}(\mathcal{M}_q)$$

das heißt wenn für jedes Wort $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \text{ genau dann wenn } \delta(q, w) \in F.$$

Wenn der Automat $\mathcal{M}$ klar ist, schreiben wir einfach $\sim$ statt $\sim_{\mathcal{M}}$.

Beispiel:



$\mathbf{L}(\mathcal{M}_A) = \{1\}^*\{0\} = \mathbf{L}(\mathcal{M}_C)$

$\mathbf{L}(\mathcal{M}_B) = \{\epsilon\}$

$\rightsquigarrow A \sim C$

Eigenschaften von $\sim_{\mathcal{M}}$

Definition (kurz): $q \sim_{\mathcal{M}} p$ genau dann wenn $\mathbf{L}(\mathcal{M}_p) = \mathbf{L}(\mathcal{M}_q)$.

Eigenschaften von $\sim_{\mathcal{M}}$

Definition (kurz): $q \sim_{\mathcal{M}} p$ genau dann wenn $\mathbf{L}(\mathcal{M}_p) = \mathbf{L}(\mathcal{M}_q)$.

Damit sehen wir leicht:

- $\sim$ ist reflexiv: $q \sim q$
- $\sim$ ist symmetrisch: wenn $q_1 \sim q_2$ dann $q_2 \sim q_1$
- $\sim$ ist transitiv: wenn $q_1 \sim q_2$ und $q_2 \sim q_3$ dann $q_1 \sim q_3$

(jeweils für alle $q, q_1, q_2, q_3 \in Q$)

Eigenschaft: $\sim$ ist eine Äquivalenzrelation.

Eigenschaften von $\sim_{\mathcal{M}}$

Definition (kurz): $q \sim_{\mathcal{M}} p$ genau dann wenn $\mathbf{L}(\mathcal{M}_p) = \mathbf{L}(\mathcal{M}_q)$.

Damit sehen wir leicht:

- $\sim$ ist reflexiv: $q \sim q$
- $\sim$ ist symmetrisch: wenn $q_1 \sim q_2$ dann $q_2 \sim q_1$
- $\sim$ ist transitiv: wenn $q_1 \sim q_2$ und $q_2 \sim q_3$ dann $q_1 \sim q_3$

(jeweils für alle $q, q_1, q_2, q_3 \in Q$)

Eigenschaft: $\sim$ ist eine Äquivalenzrelation.

Außerdem gilt für alle $\mathtt{a} \in \Sigma$

- wenn $q_1 \sim q_2$ dann $\delta(q_1, \mathtt{a}) \sim \delta(q_2, \mathtt{a})$, falls diese Übergänge definiert sind
  (daher nehmen wir im Folgenden oft eine totale Übergangsfunktion an)

Eigenschaft: $\sim$ ist verträglich mit der Übergangsfunktion.

Notation für Äquivalenzrelationen

Wir verwenden die bei Äquivalenzen üblichen Begriffe:

Wir schreiben $[q]_\sim$ für die $\sim$-Äquivalenzklasse von q, d.h.

$$[q]_\sim = \{p \in Q \mid q \sim p\}.$$

Für eine Menge $P \subseteq Q$ schreiben wir $P/_\sim$ für den Quotienten von P und $\sim$:

$$P/_\sim = \{[p]_\sim \mid p \in P\}.$$

(Die Quotientenbildung heißt Faktorisierung; sie entspricht dem „Verschmelzen“ äquivalenter Zustände.)

Wie immer gilt:

- Wenn $q_1 \sim q_2$ dann $[q_1]_\sim = [q_2]_\sim$
- Unterschiedliche Äquivalenzklassen sind disjunkt, d.h. $[q_1]_\sim \neq [q_2]_\sim$ impliziert $[q_1]_\sim \cap [q_2]_\sim = \emptyset$
- Die Äquivalenzklassen partitionieren Q, d.h. Q ist die Vereinigung der (disjunkten) Klassen

Der Quotientenautomat

Wir vereinfachen Automaten, indem wir äquivalente Zustände verschmelzen:

Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F\rangle$ mit totaler Übergangsfunktion ist der Quotientenautomat $\mathcal{M}/_\sim$ gegeben durch $\mathcal{M}/_\sim = \langle Q/_\sim, \Sigma, \delta_\sim, [q_0]_{\sim_\mathcal{M}}, F/_\sim\rangle$ wobei gilt:

- $Q/_\sim = \{[q]_\sim \mid q \in Q\}$
- $\delta_\sim([q]_\sim, \mathtt{a}) = [\delta(q, \mathtt{a})]_\sim$
- $F/_\sim = \{[q]_\sim \mid q \in F\}$

Der Quotientenautomat

Wir vereinfachen Automaten, indem wir äquivalente Zustände verschmelzen:

Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F\rangle$ mit totaler Übergangsfunktion ist der Quotientenautomat $\mathcal{M}/_\sim$ gegeben durch $\mathcal{M}/_\sim = \langle Q/_\sim, \Sigma, \delta_\sim, [q_0]_{\sim_\mathcal{M}}, F/_\sim\rangle$ wobei gilt:

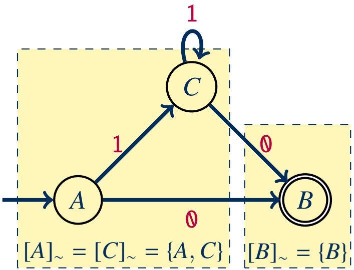
- $Q/_\sim = \{[q]_\sim \mid q \in Q\}$
- $\delta_\sim([q]_\sim, \mathtt{a}) = [\delta(q, \mathtt{a})]_\sim$
- $F/_\sim = \{[q]_\sim \mid q \in F\}$

Diese Definition ergibt Sinn, da gilt:

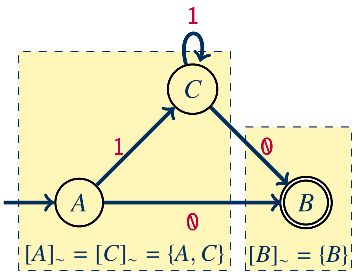
- wenn $[q]_\sim = [p]_\sim$ dann $[\delta(q, \mathtt{a})]_{\sim_\mathcal{M}} = [\delta(p, \mathtt{a})]_{\sim_\mathcal{M}}$ (Verträglichkeit von $\sim$ und δ; benötigt totale Übergangsfunktion)
- wenn $[q]_\sim = [p]_\sim$ dann $q \in F$ gdw. $p \in F$ (Übung)

$\rightsquigarrow$ Definition unabhängig vom gewählten Repräsentanten von $[q]_\sim$

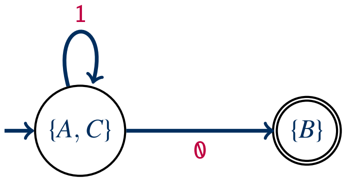
Beispiel



Beispiel



Es ergibt sich der folgende Quotientenautomat:



Korrektheit Quotientenautomat

Satz: Für jeden totalen DFA $\mathcal{M}$ gilt $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}/_\sim)$.

Beweis: Für alle $w \in \Sigma^*$ gilt:

$w \in \mathbf{L}(\mathcal{M})$ gdw. $\delta(q_0, w) \in F$ laut Definition

Korrektheit Quotientenautomat

Satz: Für jeden totalen DFA $\mathcal{M}$ gilt $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}/_{\sim})$.

Beweis: Für alle $w \in \Sigma^*$ gilt:

$w \in \mathbf{L}(\mathcal{M})$	gdw.	$\delta(q_0, w) \in F$	laut Definition
	gdw.	$[\delta(q_0, w)]_\sim \in F/_{\sim}$	wie zuvor bemerkt (Übung)

Korrektheit Quotientenautomat

Satz: Für jeden totalen DFA $\mathcal{M}$ gilt $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}/_\sim)$.

Beweis: Für alle $w \in \Sigma^*$ gilt:

$w \in \mathbf{L}(\mathcal{M})$	gdw.	$\delta(q_0, w) \in F$	laut Definition
	gdw.	$[\delta(q_0, w)]_\sim \in F/_\sim$	wie zuvor bemerkt (Übung)
	gdw.	$\delta_\sim([q_0]_\sim, w) \in F/_\sim$	Lemma ♥

Korrektheit Quotientenautomat

Satz: Für jeden totalen DFA $\mathcal{M}$ gilt $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}/_\sim)$.

Beweis: Für alle $w \in \Sigma^*$ gilt:

$w \in \mathbf{L}(\mathcal{M})$	gdw.	$\delta(q_0, w) \in F$	laut Definition
	gdw.	$[\delta(q_0, w)]_\sim \in F/_\sim$	wie zuvor bemerkt (Übung)
	gdw.	$\delta_\sim([q_0]_\sim, w) \in F/_\sim$	Lemma ♥
	gdw.	$w \in \mathbf{L}(\mathcal{M}/_\sim)$	laut Definition

Korrektheit Quotientenautomat

Satz: Für jeden totalen DFA $\mathcal{M}$ gilt $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}/_{\sim})$.

Beweis: Für alle $w \in \Sigma^*$ gilt:

$w \in \mathbf{L}(\mathcal{M})$	gdw.	$\delta(q_0, w) \in F$	laut Definition
	gdw.	$[\delta(q_0, w)]_{\sim} \in F/_{\sim}$	wie zuvor bemerkt (Übung)
	gdw.	$\delta_{\sim}([q_0]_{\sim}, w) \in F/_{\sim}$	Lemma ♥
	gdw.	$w \in \mathbf{L}(\mathcal{M}/_{\sim})$	laut Definition

Lemma ♥: Für beliebige $q \in Q$ und $w \in \Sigma^*$ gilt:

$$[\delta(q, w)]_{\sim} = \delta_{\sim}([q]_{\sim}, w).$$

Beweis durch Induktion über $|w|$ (Übung) □

Berechnung von $\sim_{\mathcal{M}}$

Wie kann man $\sim_{\mathcal{M}}$ praktisch ermitteln?

Zuvor bemerkten wir:

(1) Wenn $q_1 \sim q_2$ dann $q_1 \in F$ gdw. $q_2 \in F$

(2) Wenn $q_1 \sim q_2$ dann $\delta(q_1, \mathtt{a}) \sim \delta(q_2, \mathtt{a})$

Umgekehrt gilt also:

(1') Wenn $q_1 \in F$ und $q_2 \notin F$ dann $q_1 \nsim q_2$

(2') Wenn $\delta(q_1, \mathtt{a}) \nsim \delta(q_2, \mathtt{a})$ dann $q_1 \nsim q_2$

Tatsächlich ist $\nsim$ die kleinste Relation, die (1') und (2') erfüllt.

$\leadsto$ Wir können $\nsim$ (und damit auch $\sim$) durch rekursive Anwendung der Regeln (1') und (2') berechnen

Algorithmus zur Berechnung von $\sim_{\mathcal{M}}$

Eingabe: DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F\rangle$
Ausgabe: $\sim_{\mathcal{M}}$

- Initialisiere $\nsim := \emptyset$
- (Regel 1) Für jedes Paar von Zuständen $\langle q, p\rangle \in Q \times Q$:
  falls $q \in F$ und $p \notin F$, dann „speichere $q \nsim p$“
- (Regel 2) Für jedes Paar $\langle q, p\rangle \in Q \times Q \setminus \nsim$ und jedes $\mathtt{a} \in \Sigma$:
  falls $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$ dann „speichere $q \nsim p$“
- Wiederhole die Anwendung von Regel 2 solange, bis es keine Änderungen mehr gibt
- Das Ergebnis ist $(Q \times Q) \setminus \nsim$

Darstellung von $\nsim$ im Algorithmus

Die Anweisung „speichere $q \nsim p$“ könnte umgesetzt werden als:

$$\nsim \; := \; \nsim \cup \{\langle q, p\rangle, \langle p, q\rangle\}$$

Es ist aber nicht nötig, alle Paare in $\nsim$ einzeln zu speichern:

- $\nsim$ ist irreflexiv, d.h. man muss $q \nsim q$ nicht betrachten
- $\nsim$ ist symmetrisch, d.h. man muss jeweils nur entweder $q \nsim p$ oder $p \nsim q$ betrachten

Darstellung von $\nsim$ im Algorithmus

Die Anweisung „speichere $q \nsim p$“ könnte umgesetzt werden als:

$$\nsim \; := \; \nsim \cup \{\langle q, p\rangle, \langle p, q\rangle\}$$

Es ist aber nicht nötig, alle Paare in $\nsim$ einzeln zu speichern:

- $\nsim$ ist irreflexiv, d.h. man muss $q \nsim q$ nicht betrachten
- $\nsim$ ist symmetrisch, d.h. man muss jeweils nur entweder $q \nsim p$ oder $p \nsim q$ betrachten

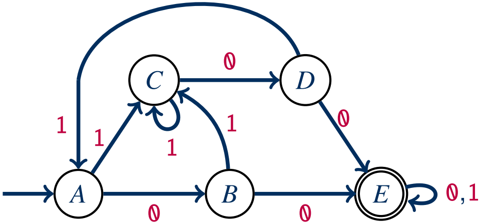
$\rightsquigarrow$ Halb-Tabelle genügt zum Eintragen der möglichen Paare

Beispiel: Für einen DFA mit Zuständen $Q = \{A, B, C, D, E\}$ genügt eine Tabelle mit zehn Feldern (statt $5^2 = 25$).

(dazu reihen wir Zustände vertikal in umgekehrter Reihenfolge)

	A	B	C	D
E				
D				
C				
B				

Beispiel Quotientenautomat



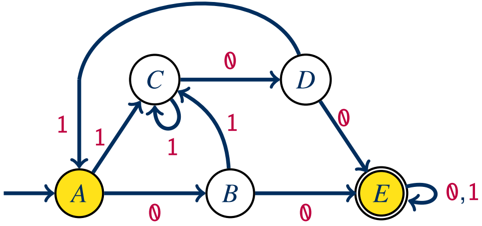
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$ impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E				
D				
C				
B				

Beispiel Quotientenautomat



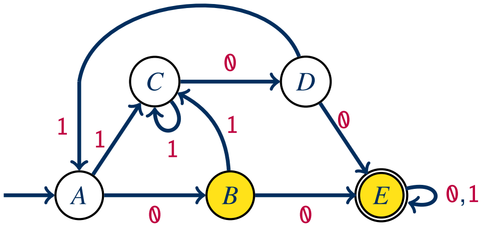
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ			
D				
C				
B				

Beispiel Quotientenautomat



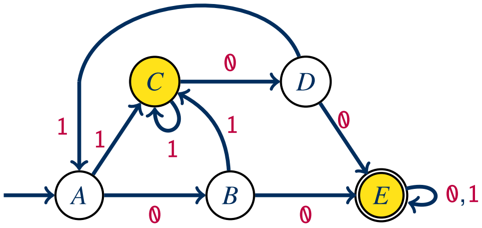
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ		
D				
C				
B				

Beispiel Quotientenautomat



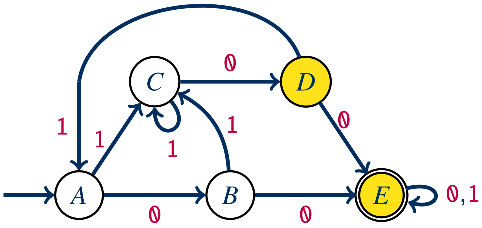
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$ impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	
D				
C				
B				

Beispiel Quotientenautomat



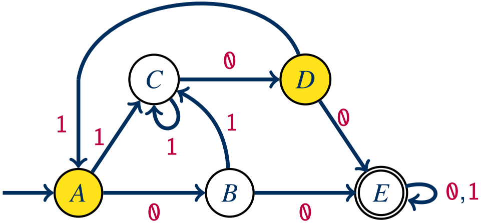
(1) $q \in F$ und $p \notin F$
impliziert $q \not\sim p$

(2) $\delta(q, \mathtt{a}) \not\sim \delta(p, \mathtt{a})$
impliziert $q \not\sim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \not\sim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D				
C				
B				

Beispiel Quotientenautomat



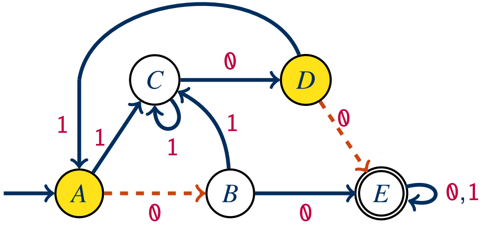
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D				
C				
B				

Beispiel Quotientenautomat



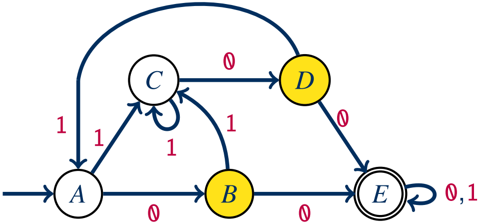
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$ impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0			
C				
B				

Beispiel Quotientenautomat



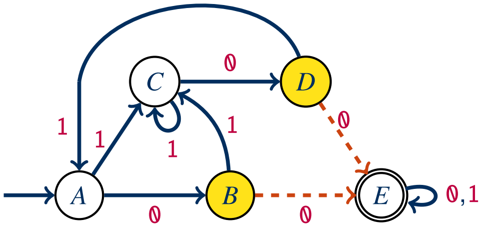
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0			
C				
B				

Beispiel Quotientenautomat



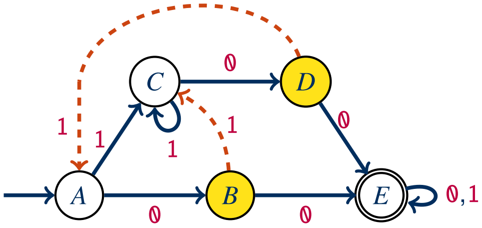
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0			
C				
B				

Beispiel Quotientenautomat



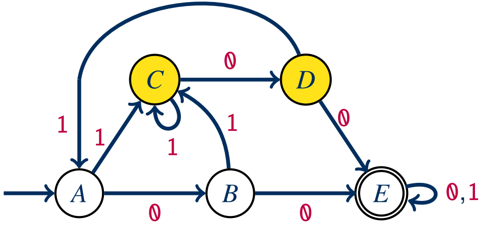
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0			
C				
B				

Beispiel Quotientenautomat



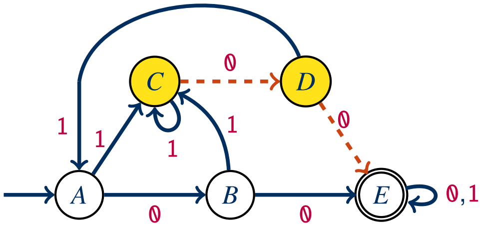
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0			
C				
B				

Beispiel Quotientenautomat



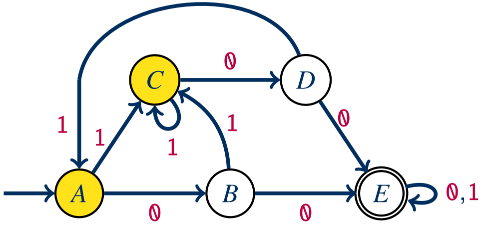
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \texttt{a}) \nsim \delta(p, \texttt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C				
B				

Beispiel Quotientenautomat



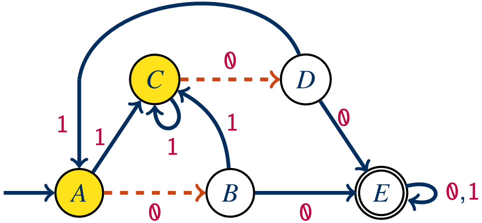
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$ impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C				
B				

Beispiel Quotientenautomat



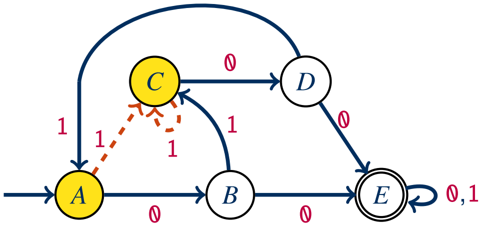
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$ impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C				
B				

Beispiel Quotientenautomat



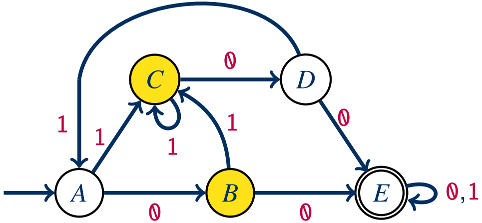
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \texttt{a}) \nsim \delta(p, \texttt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C				
B				

Beispiel Quotientenautomat



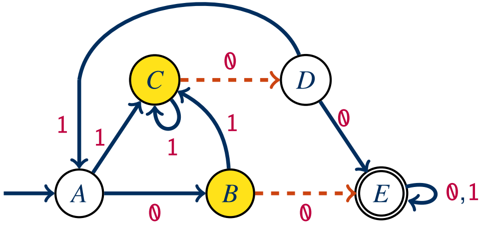
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \texttt{a}) \nsim \delta(p, \texttt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C				
B				

Beispiel Quotientenautomat



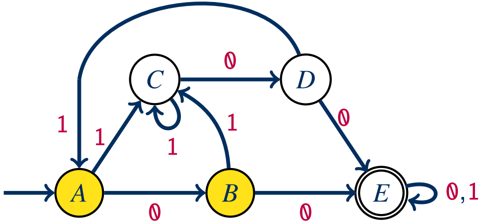
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \texttt{a}) \nsim \delta(p, \texttt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C		0		
B				

Beispiel Quotientenautomat



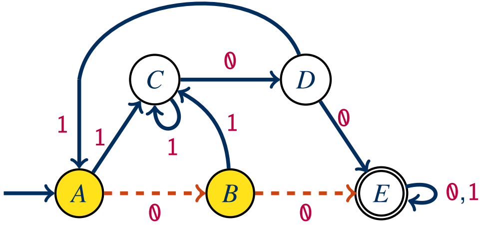
(1) $q \in F$ und $p \notin F$
impliziert $q \not\sim p$

(2) $\delta(q, \mathtt{a}) \not\sim \delta(p, \mathtt{a})$
impliziert $q \not\sim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \not\sim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C		0		
B				

Beispiel Quotientenautomat



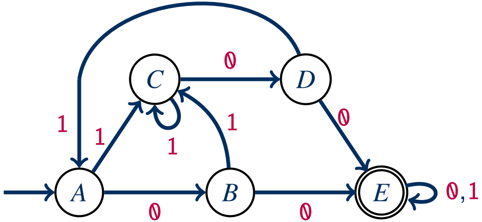
(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C		0		
B	0			

Beispiel Quotientenautomat



(1) $q \in F$ und $p \notin F$
impliziert $q \nsim p$

(2) $\delta(q, \mathtt{a}) \nsim \delta(p, \mathtt{a})$
impliziert $q \nsim p$

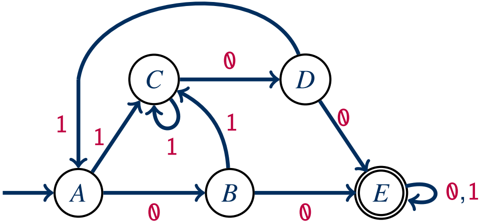
Wir tragen in der Tabelle jeweils die Wörter ein, die $q \nsim p$ zeigen:

	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C		0		
B	0			

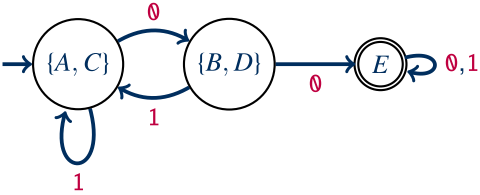
Weitere Abarbeitung von Regel (2) führt nicht mehr zu Änderungen

$$\sim \; = \{\langle B, D\rangle, \langle D, B\rangle, \langle A, C\rangle, \langle C, A\rangle\} \cup \{\langle q, q\rangle \mid q \in Q\}$$

Beispiel Quotientenautomat



Quotientenautomat:



Reduktion von Automaten

Wir können das bisher gezeigte zusammenfassen:

Sei $\mathcal{M}$ ein DFA mit totaler Übergangsfunktion. Der reduzierte Automat $\mathcal{M}_r$ ergibt sich durch folgende Schritte:

(1) Entferne alle unerreichbaren Zustände aus $\mathcal{M}$

(2) Berechne den Quotientenautomaten

Reduktion von Automaten

Wir können das bisher gezeigte zusammenfassen:

Sei $\mathcal{M}$ ein DFA mit totaler Übergangsfunktion. Der reduzierte Automat $\mathcal{M}_r$ ergibt sich durch folgende Schritte:

(1) Entferne alle unerreichbaren Zustände aus $\mathcal{M}$

(2) Berechne den Quotientenautomaten

Dieses Verfahren erzeugt den gewünschten minimalen DFA:

Satz: $\mathcal{M}_r$ ist bezüglich der Zustandsmenge der minimale DFA mit totaler Übergangsfunktion, der die Sprache $\mathbf{L}(\mathcal{M})$ erkennt.

Reduktion von Automaten

Wir können das bisher gezeigte zusammenfassen:

Sei $\mathcal{M}$ ein DFA mit totaler Übergangsfunktion. Der reduzierte Automat $\mathcal{M}_r$ ergibt sich durch folgende Schritte:

(1) Entferne alle unerreichbaren Zustände aus $\mathcal{M}$

(2) Berechne den Quotientenautomaten

Dieses Verfahren erzeugt den gewünschten minimalen DFA:

Satz: $\mathcal{M}_r$ ist bezüglich der Zustandsmenge der minimale DFA mit totaler Übergangsfunktion, der die Sprache $\mathbf{L}(\mathcal{M})$ erkennt.

Zudem stellt sich heraus, dass dieser minimale DFA eindeutig ist:

Satz: Alle minimalen DFA mit totaler Übergangsfunktion, die $\mathbf{L}(\mathcal{M})$ erkennen, sind bis auf Umbenennung von Zuständen gleich (sie sind isomorph). Daher hängt $\mathcal{M}_r$ nur von $\mathbf{L}(\mathcal{M})$ ab, nicht von $\mathcal{M}$.

Korrektheit Minimalautomat

Satz: $\mathcal{M}_r$ ist bezüglich der Zustandsmenge der minimale DFA mit totaler Übergangsfunktion, der die Sprache $\mathbf{L}(\mathcal{M})$ erkennt.

Beweisplan:

1. $\mathcal{M}_r$ erkennt $\mathbf{L}(\mathcal{M})$: Dies folgt aus der Korrektheit der Quotientenbildung bei Automaten
2. $\mathcal{M}_r$ ist minimal für diese Eigenschaft: Wir werden dies in mehreren Schritten zeigen:
   - Wir konstruieren einen weiteren minimalen Automaten $\mathcal{M}_{\mathbf{L}}$ direkt aus $\mathbf{L}$
   - Wir zeigen, dass $\mathcal{M}_{\mathbf{L}}$ und $\mathcal{M}_r$ bis auf Umbenennung von Zuständen gleich sind

   Damit ist auch die behauptete Eindeutigkeit gezeigt.

Die Nerode-Rechtskongruenz

Für eine Sprache $\mathbf{L} \subseteq \Sigma^*$ ist die Nerode-Rechtskongruenz $\simeq_{\mathbf{L}}$ wie folgt definiert. Für Wörter $u, v \in \Sigma^*$ sei $u \simeq_{\mathbf{L}} v$ wenn gilt:

für alle $w \in \Sigma^*$ gilt $uw \in \mathbf{L}$ genau dann wenn $vw \in \mathbf{L}$.

Wenn $\mathbf{L}$ klar ist, dann schreiben wir einfach $\simeq$ statt $\simeq_{\mathbf{L}}$

Die Nerode-Rechtskongruenz

Für eine Sprache $\mathbf{L} \subseteq \Sigma^*$ ist die Nerode-Rechtskongruenz $\simeq_{\mathbf{L}}$ wie folgt definiert. Für Wörter $u, v \in \Sigma^*$ sei $u \simeq_{\mathbf{L}} v$ wenn gilt:

$$\text{für alle } w \in \Sigma^* \text{ gilt } uw \in \mathbf{L} \text{ genau dann wenn } vw \in \mathbf{L}.$$

Wenn $\mathbf{L}$ klar ist, dann schreiben wir einfach $\simeq$ statt $\simeq_{\mathbf{L}}$

Anders gesagt: zwei Wörter v und u sind kongruent, wenn man in einem Wort das Präfix v gegen u vertauschen kann, ohne dass dies den Status des Worts bezüglich $\mathbf{L}$ verändert

Dies kann mit der Idee der Zustandsäquivalenz verglichen werden:

(Rückblick) Für Zustände $p, q \in Q$ sei $p \sim q$ wenn gilt:

$$\text{für alle } w \in \Sigma^* \text{ gilt } \delta(p, w) \in F \text{ genau dann wenn } \delta(q, w) \in F.$$

Eigenschaften von $\simeq$

Definition (kurz): $u \simeq_{\mathbf{L}} v$ wenn für alle $w \in \Sigma^*$ gilt: $uw \in \mathbf{L}$ genau dann wenn $vw \in \mathbf{L}$.

Eigenschaften von $\simeq$

Definition (kurz): $u \simeq_{\mathbf{L}} v$ wenn für alle $w \in \Sigma^*$ gilt: $uw \in \mathbf{L}$ genau dann wenn $vw \in \mathbf{L}$.

Damit sehen wir leicht:

- $\simeq$ ist reflexiv: $u \simeq u$
- $\simeq$ ist symmetrisch: wenn $u \simeq v$ dann $v \simeq u$
- $\simeq$ ist transitiv: wenn $u \simeq v$ und $v \simeq w$ dann $u \simeq w$

(jeweils für alle $u, v, w \in \Sigma^*$)

Eigenschaft: $\simeq$ ist eine Äquivalenzrelation.

Eigenschaften von $\simeq$

Definition (kurz): $u \simeq_{\mathbf{L}} v$ wenn für alle $w \in \Sigma^*$ gilt: $uw \in \mathbf{L}$ genau dann wenn $vw \in \mathbf{L}$.

Damit sehen wir leicht:

- $\simeq$ ist reflexiv: $u \simeq u$
- $\simeq$ ist symmetrisch: wenn $u \simeq v$ dann $v \simeq u$
- $\simeq$ ist transitiv: wenn $u \simeq v$ und $v \simeq w$ dann $u \simeq w$

(jeweils für alle $u, v, w \in \Sigma^*$)

Eigenschaft: $\simeq$ ist eine Äquivalenzrelation.

Außerdem gilt für alle $w \in \Sigma^*$

- wenn $u \simeq v$ dann $uw \simeq vw$

Eigenschaft: $\simeq$ ist verträglich mit der Konkatenation von rechts.

Dies rechtfertigt die Bezeichnung Rechtskongruenz.

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_{\simeq} = \{\mathtt{a}\}^*$: für jedes $v \in [\epsilon]_{\simeq}$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\mathtt{a}\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{b}]_\simeq = \{\mathtt{a}\}^*\{\mathtt{b}\}^+$: für jedes $v \in [\mathtt{b}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{\mathtt{b}\}^*$

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\mathtt{a}\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{b}]_\simeq = \{\mathtt{a}\}^*\{\mathtt{b}\}^+$: für jedes $v \in [\mathtt{b}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{\mathtt{b}\}^*$
- $[\mathtt{ba}]_\simeq = \Sigma^* \setminus \mathbf{L}$: für jedes $v \in [\mathtt{ba}]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Beispiel

Die Sprache $\mathbf{L} = \{a\}^*\{b\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{a\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[b]_\simeq = \{a\}^*\{b\}^+$: für jedes $v \in [b]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{b\}^*$
- $[ba]_\simeq = \Sigma^* \setminus \mathbf{L}$: für jedes $v \in [ba]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Die endliche Sprache $\mathbf{L} = \{a, ab, ba\}$ hat die folgenden Nerode-Äquivalenzklassen:

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\mathtt{a}\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{b}]_\simeq = \{\mathtt{a}\}^*\{\mathtt{b}\}^+$: für jedes $v \in [\mathtt{b}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{\mathtt{b}\}^*$
- $[\mathtt{ba}]_\simeq = \Sigma^* \setminus \mathbf{L}$: für jedes $v \in [\mathtt{ba}]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Die endliche Sprache $\mathbf{L} = \{\mathtt{a}, \mathtt{ab}, \mathtt{ba}\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\mathtt{a}\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{b}]_\simeq = \{\mathtt{a}\}^*\{\mathtt{b}\}^+$: für jedes $v \in [\mathtt{b}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{\mathtt{b}\}^*$
- $[\mathtt{ba}]_\simeq = \Sigma^* \setminus \mathbf{L}$: für jedes $v \in [\mathtt{ba}]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Die endliche Sprache $\mathbf{L} = \{\mathtt{a}, \mathtt{ab}, \mathtt{ba}\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_\simeq = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\epsilon, \mathtt{b}\}$

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\mathtt{a}\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{b}]_\simeq = \{\mathtt{a}\}^*\{\mathtt{b}\}^+$: für jedes $v \in [\mathtt{b}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{\mathtt{b}\}^*$
- $[\mathtt{ba}]_\simeq = \Sigma^* \setminus \mathbf{L}$: für jedes $v \in [\mathtt{ba}]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Die endliche Sprache $\mathbf{L} = \{\mathtt{a}, \mathtt{ab}, \mathtt{ba}\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_\simeq = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\epsilon, \mathtt{b}\}$
- $[\mathtt{b}]_\simeq = \{\mathtt{b}\}$: $\mathtt{b}w \in \mathbf{L}$ gdw. $w = \mathtt{a}$

Beispiel

Die Sprache $\mathbf{L} = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\mathtt{a}\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{b}]_\simeq = \{\mathtt{a}\}^*\{\mathtt{b}\}^+$: für jedes $v \in [\mathtt{b}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{\mathtt{b}\}^*$
- $[\mathtt{ba}]_\simeq = \Sigma^* \setminus \mathbf{L}$: für jedes $v \in [\mathtt{ba}]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Die endliche Sprache $\mathbf{L} = \{\mathtt{a}, \mathtt{ab}, \mathtt{ba}\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_\simeq = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\epsilon, \mathtt{b}\}$
- $[\mathtt{b}]_\simeq = \{\mathtt{b}\}$: $\mathtt{b}w \in \mathbf{L}$ gdw. $w = \mathtt{a}$
- $[\mathtt{ab}]_\simeq = \{\mathtt{ab}, \mathtt{ba}\}$: für jedes $v \in [\mathtt{ab}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w = \epsilon$

Beispiel

Die Sprache $\mathbf{L} = \{\mathrm{a}\}^*\{\mathrm{b}\}^*$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\mathrm{a}\}^*$: für jedes $v \in [\epsilon]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathrm{b}]_\simeq = \{\mathrm{a}\}^*\{\mathrm{b}\}^+$: für jedes $v \in [\mathrm{b}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w \in \{\mathrm{b}\}^*$
- $[\mathrm{ba}]_\simeq = \Sigma^* \setminus \mathbf{L}$: für jedes $v \in [\mathrm{ba}]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Die endliche Sprache $\mathbf{L} = \{\mathrm{a}, \mathrm{ab}, \mathrm{ba}\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathrm{a}]_\simeq = \{\mathrm{a}\}$: $\mathrm{a}w \in \mathbf{L}$ gdw. $w \in \{\epsilon, \mathrm{b}\}$
- $[\mathrm{b}]_\simeq = \{\mathrm{b}\}$: $\mathrm{b}w \in \mathbf{L}$ gdw. $w = \mathrm{a}$
- $[\mathrm{ab}]_\simeq = \{\mathrm{ab}, \mathrm{ba}\}$: für jedes $v \in [\mathrm{ab}]_\simeq$ ist $vw \in \mathbf{L}$ gdw. $w = \epsilon$
- $[\mathrm{bb}]_\simeq = \Sigma^* \setminus \{\epsilon, \mathrm{a}, \mathrm{b}, \mathrm{ab}, \mathrm{ba}\}$: für jedes $v \in [\mathrm{bb}]_\simeq$ ist $vw \notin \mathbf{L}$ für alle $w \in \Sigma^*$

Beispiel (2)

Die Sprache **L** $= \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

Beispiel (2)

Die Sprache $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_{\simeq} = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$

Beispiel (2)

Die Sprache $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_{\simeq} = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_{\simeq} = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+1} \mid n \geq 0\}$

Beispiel (2)

Die Sprache $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_{\simeq} = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_{\simeq} = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+1} \mid n \geq 0\}$
- $[\mathtt{aa}]_{\simeq} = \{\mathtt{aa}\}$: $\mathtt{aa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+2} \mid n \geq 0\}$

Beispiel (2)

Die Sprache $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_\simeq = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+1} \mid n \geq 0\}$
- $[\mathtt{aa}]_\simeq = \{\mathtt{aa}\}$: $\mathtt{aa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+2} \mid n \geq 0\}$
- $[\mathtt{aaa}]_\simeq = \{\mathtt{aaa}\}$: $\mathtt{aaa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+3} \mid n \geq 0\}$

Beispiel (2)

Die Sprache $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_\simeq = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+1} \mid n \geq 0\}$
- $[\mathtt{aa}]_\simeq = \{\mathtt{aa}\}$: $\mathtt{aa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+2} \mid n \geq 0\}$
- $[\mathtt{aaa}]_\simeq = \{\mathtt{aaa}\}$: $\mathtt{aaa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+3} \mid n \geq 0\}$
- …unendlich viele Äquivalenzklassen $[\mathtt{a}^n]_\simeq = \{\mathtt{a}^n\}$

Beispiel (2)

Die Sprache $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_\simeq = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+1} \mid n \geq 0\}$
- $[\mathtt{aa}]_\simeq = \{\mathtt{aa}\}$: $\mathtt{aa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+2} \mid n \geq 0\}$
- $[\mathtt{aaa}]_\simeq = \{\mathtt{aaa}\}$: $\mathtt{aaa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+3} \mid n \geq 0\}$
- ...unendlich viele Äquivalenzklassen $[\mathtt{a}^n]_\simeq = \{\mathtt{a}^n\}$

Es gibt weitere Formen von Äquivalenzklassen, z.B. $[\mathtt{aab}] = \{\mathtt{a}^{n+1}\mathtt{b}^n \mid n \geq 0\}$.

Beispiel (2)

Die Sprache $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat die folgenden Nerode-Äquivalenzklassen:

- $[\epsilon]_\simeq = \{\epsilon\}$: $\epsilon w \in \mathbf{L}$ gdw. $w \in \mathbf{L}$
- $[\mathtt{a}]_\simeq = \{\mathtt{a}\}$: $\mathtt{a}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+1} \mid n \geq 0\}$
- $[\mathtt{aa}]_\simeq = \{\mathtt{aa}\}$: $\mathtt{aa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+2} \mid n \geq 0\}$
- $[\mathtt{aaa}]_\simeq = \{\mathtt{aaa}\}$: $\mathtt{aaa}w \in \mathbf{L}$ gdw. $w \in \{\mathtt{a}^n\mathtt{b}^{n+3} \mid n \geq 0\}$
- …unendlich viele Äquivalenzklassen $[\mathtt{a}^n]_\simeq = \{\mathtt{a}^n\}$

Es gibt weitere Formen von Äquivalenzklassen, z.B. $[\mathtt{aab}] = \{\mathtt{a}^{n+1}\mathtt{b}^n \mid n \geq 0\}$.

$\leadsto$ $\mathbf{L} = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ hat unendlich viele Nerode-Äquivalenzklassen

$\simeq$ und reguläre Sprachen

Wir werden zeigen, dass jede reguläre Sprache endlich viele $\simeq$-Äquivalenzklassen hat.

$\simeq$ und reguläre Sprachen

Wir werden zeigen, dass jede reguläre Sprache endlich viele $\simeq$-Äquivalenzklassen hat.

Es gilt sogar noch etwas stärkeres:

Satz (Myhill & Nerode): Eine Sprache $\mathbf{L}$ ist genau dann regulär, wenn $\simeq_{\mathbf{L}}$ endlich viele Äquivalenzklassen hat.

Beweis: Siehe nächste Vorlesung.

Zusammenfassung und Ausblick

Im Quotientenautomaten werden äquivalente Zustände verschmolzen

Äquivalente Zustände in einem (totalen) DFA können rekursiv ermittelt werden

Der Satz von Myhill und Nerode charakterisiert reguläre Sprachen

Offene Fragen:

- Wie geht es weiter mit dem Beweis der Eindeutigkeit des Minimalautomaten?
- Wie aufwändig sind die verschiedenen Konstruktionen auf regulären Sprachen?
- Welche Sprachen sind nicht regulär?