TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik, Institut für Künstliche Intelligenz, Professur für Computational Logic

FORMALE SYSTEME

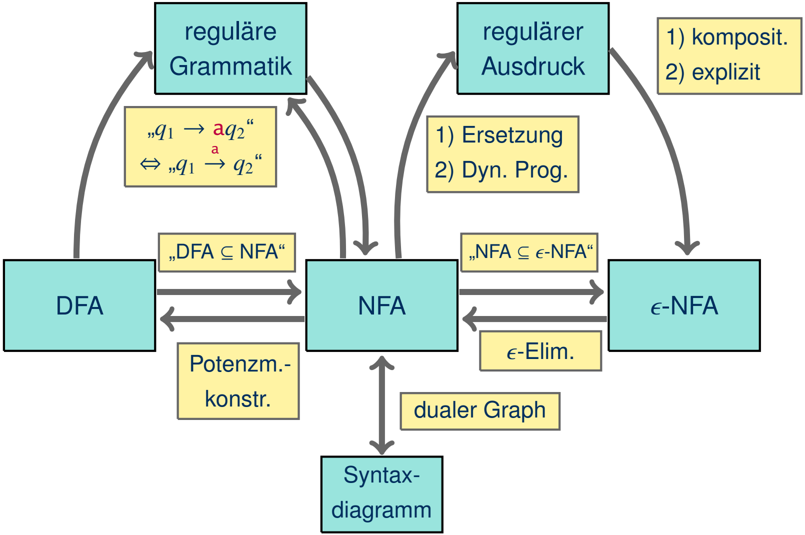
8. Vorlesung: Minimale Automaten

Hannes Straß

TU Dresden, 4. November 2021

Rückblick

Darstellungen von Typ-3-Sprachen



Aufzeichnung startet . . .

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:

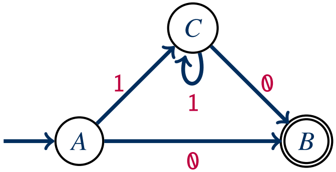
Beispiel: Sei $\mathcal{M}$ ein endlicher Automat, bei dem alle Zustände erreichbar sind und einen Endzustand erreichen können. Der Vereinigungsautomat[a] $\mathcal{M} \oplus \mathcal{M}$ akzeptiert die selbe Sprache, hat nur erreichbare Zustände, aber die doppelte Zustandszahl.

[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.

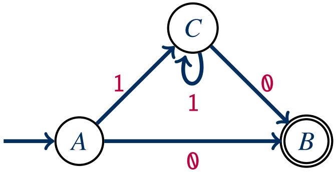
Interessanter sind nichtminimale DFAs:



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.

Ansatz zur Minimierung: 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 genau dann $\mathcal{M}$-äquivalent, in Symbolen $p \sim_{\mathcal{M}} q$, wenn gilt:

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

- Also gilt $p \sim_{\mathcal{M}} q$ genau dann, wenn für jedes $w \in \Sigma^*$ gilt: $\delta(p, w) \in F$ gdw. $\delta(q, w) \in F$.
- Wenn der Automat $\mathcal{M}$ klar ist, schreiben wir einfach $\sim$ statt $\sim_{\mathcal{M}}$.

Äquivalenz von Zuständen

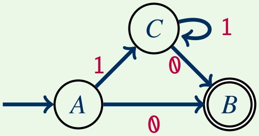
Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle$ und einen Zustand $q \in Q$ sei $\mathcal{M}_q = \langle Q, \Sigma, \delta, q, F \rangle$ der abgewandelte DFA mit Startzustand q.

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

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

- Also gilt $p \sim_{\mathcal{M}} q$ genau dann, wenn für jedes $w \in \Sigma^*$ gilt: $\delta(p, w) \in F$ gdw. $\delta(q, w) \in F$.
- 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): Es gilt $q \sim_{\mathcal{M}} p$ genau dann, wenn $\mathbf{L}(\mathcal{M}_p) = \mathbf{L}(\mathcal{M}_q)$.

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

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

Damit sehen wir leicht (jeweils für alle $q, q_1, q_2, q_3 \in Q$):

- $\sim$ ist reflexiv: Es gilt $q \sim q$.
- $\sim$ ist symmetrisch: Aus $q_1 \sim q_2$ folgt stets $q_2 \sim q_1$.
- $\sim$ ist transitiv: Wenn $q_1 \sim q_2$ und $q_2 \sim q_3$, dann auch $q_1 \sim q_3$.

Eigenschaft: $\sim$ ist eine Äquivalenzrelation.

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

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

Damit sehen wir leicht (jeweils für alle $q, q_1, q_2, q_3 \in Q$):

- $\sim$ ist reflexiv: Es gilt $q \sim q$.
- $\sim$ ist symmetrisch: Aus $q_1 \sim q_2$ folgt stets $q_2 \sim q_1$.
- $\sim$ ist transitiv: Wenn $q_1 \sim q_2$ und $q_2 \sim q_3$, dann auch $q_1 \sim q_3$.

Eigenschaft: $\sim$ ist eine Äquivalenzrelation.

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

- Wenn $q_1 \sim q_2$, dann gilt auch $\delta(q_1, \mathtt{a}) \sim \delta(q_2, \mathtt{a})$, 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 und Notationen:

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 (für alle $q_1, q_2 \in Q$):

- Aus $q_1 \sim q_2$ folgt stets $[q_1]_\sim = [q_2]_\sim$.
- Unterschiedliche Äquivalenzklassen sind disjunkt:

  $$[q_1]_\sim \neq [q_2]_\sim \text{ impliziert } [q_1]_\sim \cap [q_2]_\sim = \emptyset$$

- Die Äquivalenzklassen partitionieren Q:

$$Q = \bigcup_{q \in Q} [q]_\sim$$

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:

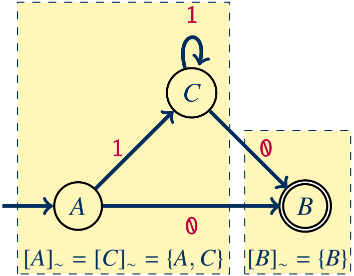
- $Q/_\sim = \{[q]_\sim \mid q \in Q\}$
- $\delta_\sim([q]_\sim, \mathtt{a}) = [\delta(q, \mathtt{a})]_\sim$
- $F/_\sim = \{[q]_\sim \mid q \in F\}$

Der Quotientenautomat $\mathcal{M}/_\sim$ ist wohldefiniert, da gilt:

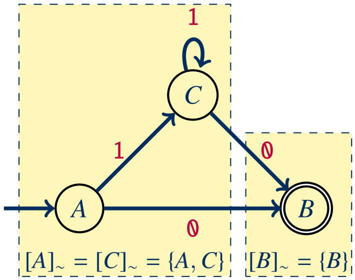
- Wenn $[q]_\sim = [p]_\sim$, dann ist auch $[\delta(q, \mathtt{a})]_{\sim_\mathcal{M}} = [\delta(p, \mathtt{a})]_{\sim_\mathcal{M}}$.
  (Verträglichkeit von $\sim$ und δ; benötigt totale Übergangsfunktion.)
- Aus $[q]_\sim = [p]_\sim$ folgt stets: $q \in F$ gdw. $p \in F$. (♠, Übung)

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

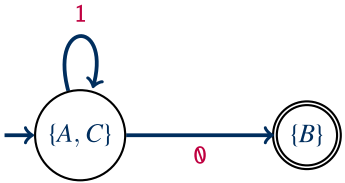
Beispiel



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:

$$w \in \mathbf{L}(\mathcal{M}) \quad \text{gdw.} \quad \delta(q_0, w) \in F \qquad \text{(Definition von } \mathbf{L}(\mathcal{M})\text{)}$$

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:

$$
\begin{array}{rlll}
w \in \mathbf{L}(\mathcal{M}) & \text{gdw.} & \delta(q_0, w) \in F & \text{(Definition von } \mathbf{L}(\mathcal{M})\text{)} \\
& \text{gdw.} & [\delta(q_0, w)]_\sim \in F/_\sim & (\spadesuit, \text{Übung})
\end{array}
$$

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:

$$
\begin{array}{rlll}
w \in \mathbf{L}(\mathcal{M}) & \text{gdw.} & \delta(q_0, w) \in F & \text{(Definition von } \mathbf{L}(\mathcal{M})\text{)} \\
& \text{gdw.} & [\delta(q_0, w)]_\sim \in F/_\sim & (\spadesuit, \text{Übung}) \\
& \text{gdw.} & \delta_\sim([q_0]_\sim, w) \in F/_\sim & (\text{Lemma } \heartsuit)
\end{array}
$$

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:

$$
\begin{array}{rlll}
w \in \mathbf{L}(\mathcal{M}) & \text{gdw.} & \delta(q_0, w) \in F & \text{(Definition von } \mathbf{L}(\mathcal{M})\text{)} \\
& \text{gdw.} & [\delta(q_0, w)]_\sim \in F/_\sim & (\spadesuit\text{, Übung}) \\
& \text{gdw.} & \delta_\sim([q_0]_\sim, w) \in F/_\sim & \text{(Lemma } \heartsuit\text{)} \\
& \text{gdw.} & w \in \mathbf{L}(\mathcal{M}/_\sim) & \text{(Definition von } \mathcal{M}/_\sim\text{)}
\end{array}
$$

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:

$$
\begin{array}{rlll}
w \in \mathbf{L}(\mathcal{M}) & \text{gdw.} & \delta(q_0, w) \in F & \text{(Definition von } \mathbf{L}(\mathcal{M})\text{)} \\
& \text{gdw.} & [\delta(q_0, w)]_\sim \in F/_\sim & (\spadesuit\text{, Übung}) \\
& \text{gdw.} & \delta_\sim([q_0]_\sim, w) \in F/_\sim & \text{(Lemma } \heartsuit\text{)} \\
& \text{gdw.} & w \in \mathbf{L}(\mathcal{M}/_\sim) & \text{(Definition von } \mathcal{M}/_\sim\text{)}
\end{array}
$$

Lemma $\heartsuit$: 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) $\square$

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

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

Zuvor bemerkten wir:

(1) Aus $q_1 \sim q_2$ folgt stets: $q_1 \in F$ gdw. $q_2 \in F$.

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

Umgekehrt gilt also:

$(1')$ Aus $q_1 \in F$ und $q_2 \notin F$ folgt immer $q_1 \nsim q_2$.

$(2')$ 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, man muss also $q \nsim q$ nicht betrachten;
- $\nsim$ ist symmetrisch, d.h. man muss jeweils nur entweder $q \nsim p$ oder $p \nsim q$ betrachten.

Darstellung von $\not\sim$ im Algorithmus

Die Anweisung „speichere $q \not\sim p$“ könnte umgesetzt werden als:

$$\not\sim \; := \; \not\sim \cup \{\langle q, p\rangle, \langle p, q\rangle\}$$

Es ist aber nicht nötig, alle Paare in $\not\sim$ einzeln zu speichern:

- $\not\sim$ ist irreflexiv, man muss also $q \not\sim q$ nicht betrachten;
- $\not\sim$ ist symmetrisch, d.h. man muss jeweils nur entweder $q \not\sim p$ oder $p \not\sim q$ betrachten.

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

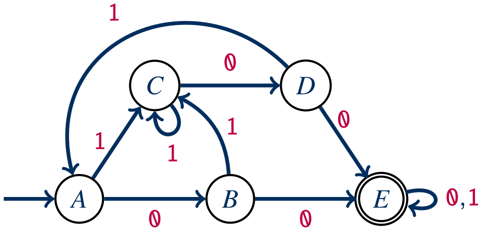
Beispiel: Für einen DFA mit 5 Zuständen $Q = \{A, B, C, D, E\}$ genügt eine Tabelle mit zehn Feldern.

(Statt $5^2 = 25$ also nur noch $(5^2 - 5)/2 = 10$).

(Dazu reihen wir Zustände vertikal in umgekehrter Reihenfolge.)

	A	B	C	D
E				
D				
C				
B				

Beispiel Quotientenautomat



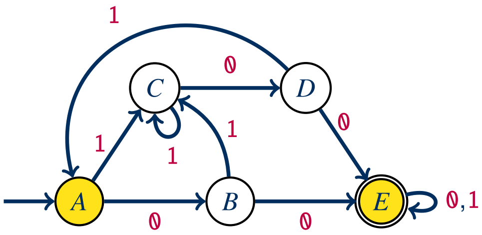
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



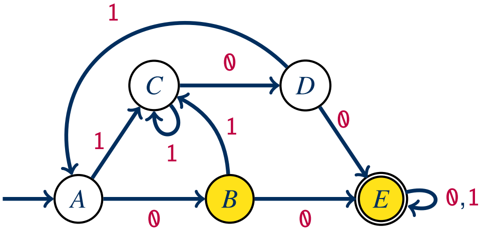
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



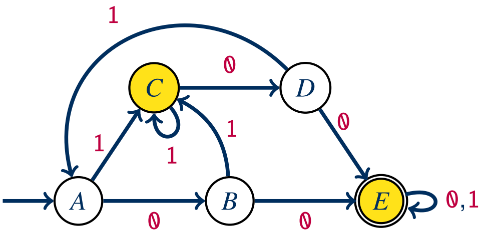
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



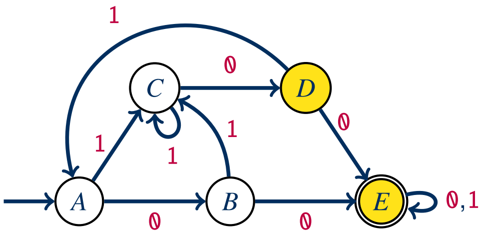
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



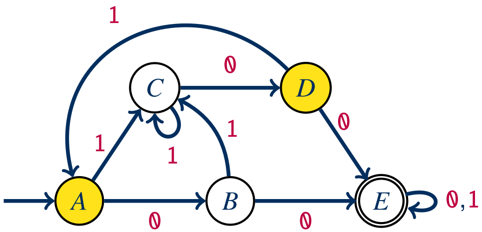
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



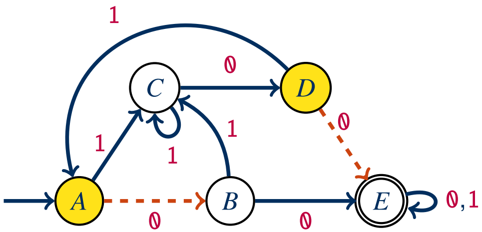
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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				
C				
B				

Beispiel Quotientenautomat



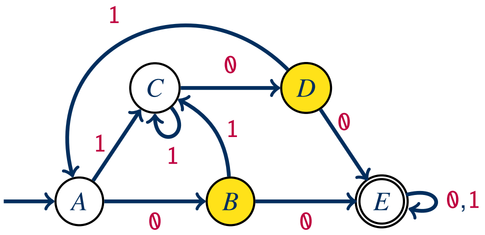
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



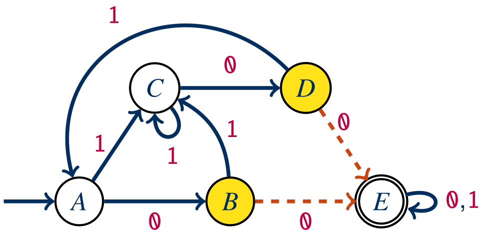
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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	$\mathtt{0}$			
C				
B				

Beispiel Quotientenautomat



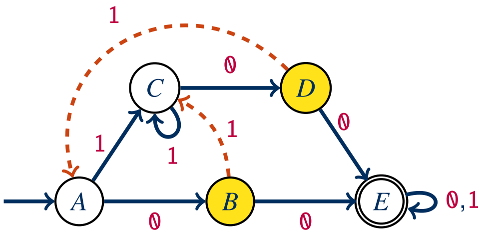
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



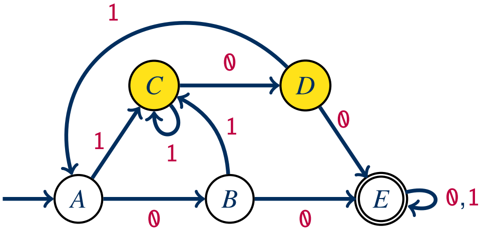
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



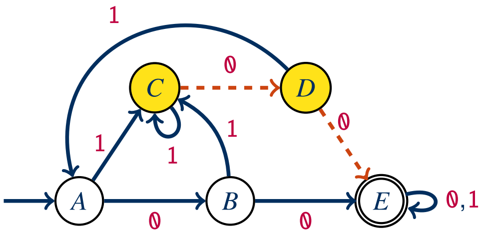
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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	$\mathtt{0}$			
C				
B				

Beispiel Quotientenautomat



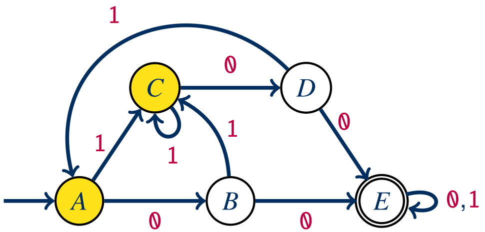
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



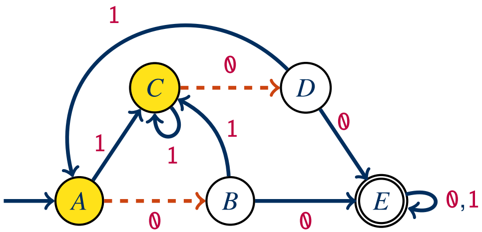
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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	$\mathtt{0}$		$\mathtt{0}$	
C				
B				

Beispiel Quotientenautomat



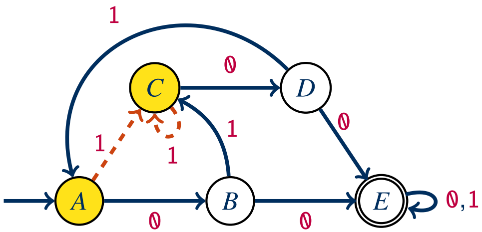
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



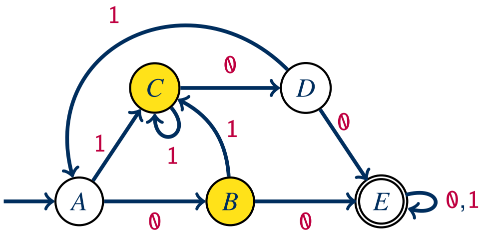
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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



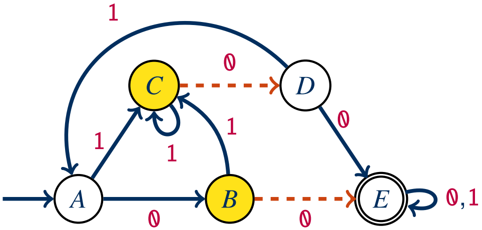
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

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



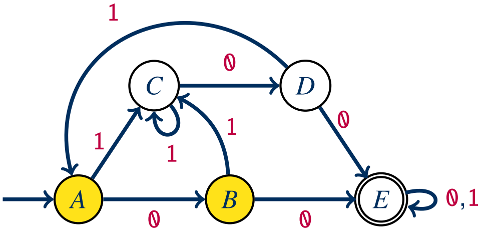
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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		0		
B				

Beispiel Quotientenautomat



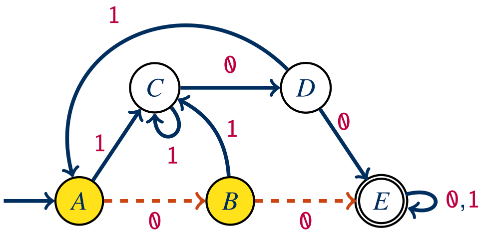
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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				

Beispiel Quotientenautomat



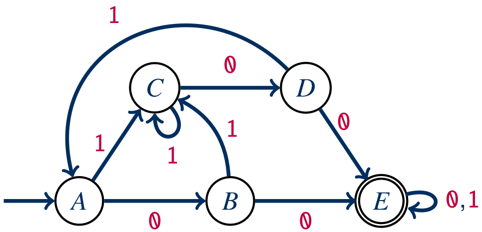
(1) $q \in F$ und $p \notin F$ impliziert $q \nsim p$.

(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:

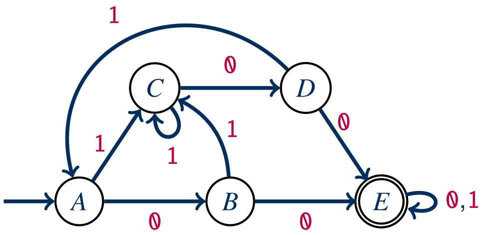
	A	B	C	D
E	ϵ	ϵ	ϵ	ϵ
D	0		0	
C		0		
B	0			

Weitere Abarbeitung von Regel (2) führt nicht mehr zu Änderungen. Es folgt:

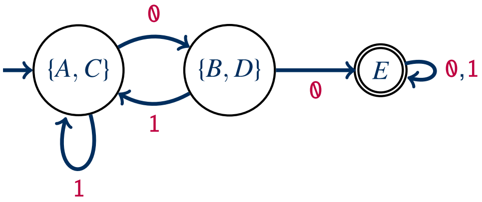
$$\sim\ = \{\langle B, D\rangle, \langle D, B\rangle, \langle A, C\rangle, \langle C, A\rangle\} \cup \{\langle q, q\rangle \mid q \in Q\},\ \text{also}$$

$$Q/_{\sim} = \{\{A, C\}, \{B, D\}, \{E\}\}.$$

Beispiel Quotientenautomat



Quotientenautomat:



Quiz: Quotientenautomat

Quiz: Sei $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F\rangle$ ein DFA mit totaler Übergangsfunktion.
...

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 $\mathbf{L}(\mathcal{M})$.
   - 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 $u \simeq_{\mathbf{L}} v$ genau dann, 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 $u \simeq_{\mathbf{L}} v$ genau dann, 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 $\simeq_{\mathbf{L}}$

Also: Für $u, v \in \Sigma^*$ gilt $u \simeq_{\mathbf{L}} v$ gdw. für alle $w \in \Sigma^*$ gilt:
(1) Ist $uw \in \mathbf{L}$, dann auch $vw \in \mathbf{L}$; und (2) Ist $uw \notin \mathbf{L}$, dann auch $vw \notin \mathbf{L}$.

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 $u \simeq_{\mathbf{L}} v$ genau dann, 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 $\simeq_{\mathbf{L}}$

Also: Für $u, v \in \Sigma^*$ gilt $u \simeq_{\mathbf{L}} v$ gdw. für alle $w \in \Sigma^*$ gilt:
(1) Ist $uw \in \mathbf{L}$, dann auch $vw \in \mathbf{L}$; und (2) Ist $uw \notin \mathbf{L}$, dann auch $vw \notin \mathbf{L}$.

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

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 $u \simeq_{\mathbf{L}} v$ genau dann, 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 $\simeq_{\mathbf{L}}$

Also: Für $u, v \in \Sigma^*$ gilt $u \simeq_{\mathbf{L}} v$ gdw. für alle $w \in \Sigma^*$ gilt:
(1) Ist $uw \in \mathbf{L}$, dann auch $vw \in \mathbf{L}$; und (2) Ist $uw \notin \mathbf{L}$, dann auch $vw \notin \mathbf{L}$.

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

Anders gesagt: Zwei Wörter u und v sind nicht kongruent, wenn es ein Wort $w \in \Sigma^*$ gibt, sodass $|\{uw, vw\} \cap \mathbf{L}| = 1$ gilt.

Dies kann mit der Idee der Zustandsäquivalenz verglichen werden:

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

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

Eigenschaften von $\simeq$

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

Eigenschaften von $\simeq$

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

Damit sehen wir leicht (jeweils für alle $u, v, w \in \Sigma^*$):

- $\simeq$ ist reflexiv: $u \simeq u$
- $\simeq$ ist symmetrisch: Aus $u \simeq v$ folgt stets $v \simeq u$.
- $\simeq$ ist transitiv: Wenn $u \simeq v$ und $v \simeq w$, dann auch $u \simeq w$.

Eigenschaft: $\simeq$ ist eine Äquivalenzrelation.

Eigenschaften von $\simeq$

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

Damit sehen wir leicht (jeweils für alle $u, v, w \in \Sigma^*$):

- $\simeq$ ist reflexiv: $u \simeq u$
- $\simeq$ ist symmetrisch: Aus $u \simeq v$ folgt stets $v \simeq u$.
- $\simeq$ ist transitiv: Wenn $u \simeq v$ und $v \simeq w$, dann auch $u \simeq w$.

Eigenschaft: $\simeq$ ist eine Äquivalenzrelation.

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

- $u \simeq v$ impliziert $uw \simeq vw$.

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

Dies rechtfertigt die Bezeichnung Rechtskongruenz.

Beispiele

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

Beispiele

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. $w \in \mathbf{L}$.

Beispiele

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. $w \in \{\mathtt{b}\}^*$.

Beispiele

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 $w \in \Sigma^*$.

Beispiele

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:

Beispiele

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. $w \in \mathbf{L}$.

Beispiele

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. $w \in \{\epsilon, \mathtt{b}\}$.

Beispiele

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:

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

Beispiele

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. $w = \epsilon$.

Beispiele

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

Beispiel (2)

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

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

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

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

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

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

Quiz: Nerode-Rechtskongruenz

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

Quiz: Wir betrachten die Nerode-Rechtskongruenz $\simeq_{\mathbf{L}}$ für die formale Sprache …

$\simeq$ 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?