

FORMALE SYSTEME

9. Vorlesung: Minimale Automaten (2)

Hannes Straß

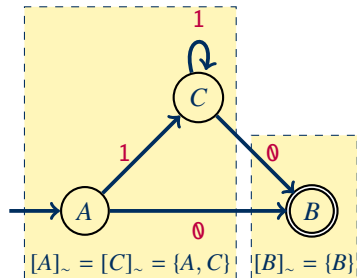
Folien: © Markus Krötzsch, <https://iccl.inf.tu-dresden.de/web/FS2020>, CC BY 3.0 DE

TU Dresden, 8. November 2021

Rückblick

Automaten verkleinern mit Quotientenbildung

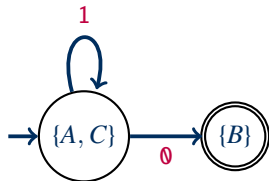
Wir betrachten DFAs mit totaler Übergangsfunktion.



Ermittlung der Zustandsäquivalenz \sim :

	A	B
C		ϵ
B	ϵ	

Quotientenautomat:



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

Verschiedene Äquivalenzrelationen

Für einen DFA \mathcal{M} und eine Sprache \mathbf{L} haben wir verschiedene Äquivalenzrelationen verwendet:

DFA-Zustandsäquivalenz $\sim_{\mathcal{M}} \subseteq Q \times Q$:

$q \sim_{\mathcal{M}} p$ genau dann, wenn für alle $w \in \Sigma^*$ gilt: $\delta(q, w) \in F$ gdw. $\delta(p, w) \in F$.

Anwendung: Vereinfachung von DFAs durch Quotientenbildung

Nerode-Rechtskongruenz $\simeq_{\mathbf{L}} \subseteq \Sigma^* \times \Sigma^*$:

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

Anwendung: automatenunabhängige Analyse von Sprachen; alternative Konstruktion eines Minimalautomaten

Aufzeichnung startet . . .

Der Satz von Myhill und Nerode

\simeq und reguläre Sprachen

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

(Die Anzahl der Äquivalenzklassen wird auch als **Index** bezeichnet.)

\simeq und reguläre Sprachen

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

(Die Anzahl der Äquivalenzklassen wird auch als **Index** bezeichnet.)

Beweis: Wir erhalten diese Eigenschaft aus der Darstellung von L mit einem DFA.

Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle$ definieren wir eine Relation $\simeq_{\mathcal{M}}$ wie folgt. Für Wörter $u, v \in \Sigma^*$ sei $u \simeq_{\mathcal{M}} v$ genau dann, wenn gilt:

$$\delta(q_0, u) = \delta(q_0, v).$$

Offensichtlich ist $\simeq_{\mathcal{M}}$ eine Äquivalenzrelation (Eigenschaften „geerbt“ von =) mit endlichem Index (es gibt nur endlich viele Zustände).

\simeq und reguläre Sprachen (2)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Behauptung: $\simeq_M \subseteq \simeq_L$, also:

Für alle $u, v \in \Sigma^*$ folgt aus $u \simeq_M v$ stets $u \simeq_L v$.

Wir betrachten einen DFA \mathcal{M} mit $L = L(\mathcal{M})$.

\simeq und reguläre Sprachen (2)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Behauptung: $\simeq_M \subseteq \simeq_L$, also:

Für alle $u, v \in \Sigma^*$ folgt aus $u \simeq_M v$ stets $u \simeq_L v$.

Wir betrachten einen DFA \mathcal{M} mit $L = L(\mathcal{M})$.

- Seien $u, v \in \Sigma^*$ Wörter mit $u \simeq_M v$.

\simeq und reguläre Sprachen (2)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Behauptung: $\simeq_M \subseteq \simeq_L$, also:

Für alle $u, v \in \Sigma^*$ folgt aus $u \simeq_M v$ stets $u \simeq_L v$.

Wir betrachten einen DFA \mathcal{M} mit $L = L(\mathcal{M})$.

- Seien $u, v \in \Sigma^*$ Wörter mit $u \simeq_M v$.
- Dann ist $\delta(q_0, u) = \delta(q_0, v)$.

\simeq und reguläre Sprachen (2)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Behauptung: $\simeq_M \subseteq \simeq_L$, also:

Für alle $u, v \in \Sigma^*$ folgt aus $u \simeq_M v$ stets $u \simeq_L v$.

Wir betrachten einen DFA \mathcal{M} mit $L = L(\mathcal{M})$.

- Seien $u, v \in \Sigma^*$ Wörter mit $u \simeq_M v$.
- Dann ist $\delta(q_0, u) = \delta(q_0, v)$.
- Dann gilt für alle $w \in \Sigma^*$: $\delta(q_0, uw) = \delta(q_0, vw)$.

\simeq und reguläre Sprachen (2)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Behauptung: $\simeq_M \subseteq \simeq_L$, also:

Für alle $u, v \in \Sigma^*$ folgt aus $u \simeq_M v$ stets $u \simeq_L v$.

Wir betrachten einen DFA \mathcal{M} mit $L = L(\mathcal{M})$.

- Seien $u, v \in \Sigma^*$ Wörter mit $u \simeq_M v$.
- Dann ist $\delta(q_0, u) = \delta(q_0, v)$.
- Dann gilt für alle $w \in \Sigma^*$: $\delta(q_0, uw) = \delta(q_0, vw)$.
- Dann gilt für alle $w \in \Sigma^*$: $\delta(q_0, uw) \in F$ gdw. $\delta(q_0, vw) \in F$.

\simeq und reguläre Sprachen (2)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Behauptung: $\simeq_M \subseteq \simeq_L$, also:

Für alle $u, v \in \Sigma^*$ folgt aus $u \simeq_M v$ stets $u \simeq_L v$.

Wir betrachten einen DFA \mathcal{M} mit $L = L(\mathcal{M})$.

- Seien $u, v \in \Sigma^*$ Wörter mit $u \simeq_M v$.
- Dann ist $\delta(q_0, u) = \delta(q_0, v)$.
- Dann gilt für alle $w \in \Sigma^*$: $\delta(q_0, uw) = \delta(q_0, vw)$.
- Dann gilt für alle $w \in \Sigma^*$: $\delta(q_0, uw) \in F$ gdw. $\delta(q_0, vw) \in F$.
- Dann gilt für alle $w \in \Sigma^*$: $uw \in L(\mathcal{M})$ gdw. $vw \in L(\mathcal{M})$.

\simeq und reguläre Sprachen (2)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Behauptung: $\simeq_M \subseteq \simeq_L$, also:

Für alle $u, v \in \Sigma^*$ folgt aus $u \simeq_M v$ stets $u \simeq_L v$.

Wir betrachten einen DFA \mathcal{M} mit $L = L(\mathcal{M})$.

- Seien $u, v \in \Sigma^*$ Wörter mit $u \simeq_M v$.
- Dann ist $\delta(q_0, u) = \delta(q_0, v)$.
- Dann gilt für alle $w \in \Sigma^*$: $\delta(q_0, uw) = \delta(q_0, vw)$.
- Dann gilt für alle $w \in \Sigma^*$: $\delta(q_0, uw) \in F$ gdw. $\delta(q_0, vw) \in F$.
- Dann gilt für alle $w \in \Sigma^*$: $uw \in L(\mathcal{M})$ gdw. $vw \in L(\mathcal{M})$.
- Dann ist $u \simeq_{L(\mathcal{M})} v$ und daher $u \simeq_L v$.

Damit ist die Behauptung gezeigt.

\simeq und reguläre Sprachen (3)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Wir haben gezeigt, dass $\simeq_M \subseteq \simeq_L$.

\simeq und reguläre Sprachen (3)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Wir haben gezeigt, dass $\simeq_M \subseteq \simeq_L$.

\rightsquigarrow Jede \simeq_L -Äquivalenzklasse besteht aus einer oder mehreren \simeq_M -Äquivalenzklassen.

\simeq und reguläre Sprachen (3)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Wir haben gezeigt, dass $\simeq_M \subseteq \simeq_L$.

\rightsquigarrow Jede \simeq_L -Äquivalenzklasse besteht aus einer oder mehreren \simeq_M -Äquivalenzklassen.

\rightsquigarrow Der Index von \simeq_L ist kleiner oder gleich dem Index von \simeq_M .

\simeq und reguläre Sprachen (3)

Satz: Wenn L regulär ist, dann hat \simeq_L endlich viele Äquivalenzklassen.

Beweis (Fortsetzung): Wir haben gezeigt, dass $\simeq_M \subseteq \simeq_L$.

- \rightsquigarrow Jede \simeq_L -Äquivalenzklasse besteht aus einer oder mehreren \simeq_M -Äquivalenzklassen.
- \rightsquigarrow Der Index von \simeq_L ist kleiner oder gleich dem Index von \simeq_M .
- \rightsquigarrow Der Index von \simeq_L ist endlich. □

Anmerkung: Der letzte Teil ist eine allgemeine Eigenschaft aller Äquivalenzrelationen. Für Äquivalenzrelationen $R, S \subseteq M \times M$ gilt: Wenn $R \subseteq S$, dann ist der Index von R größer oder gleich dem Index von S .

Die „kleinste“ Äquivalenzrelation, die Gleichheit $\{\langle m, m \rangle \mid m \in M\}$, hat den größten Index (die Anzahl der Elemente der Grundmenge M); die „größte“ Äquivalenzrelation, $M \times M$, hat nur eine Äquivalenzklasse (alle Elemente sind zueinander äquivalent). Formal gilt: $\{\langle m, m \rangle \mid m \in M\} \subseteq R \subseteq S \subseteq M \times M$.

Verschiedene Äquivalenzrelationen

Für einen DFA \mathcal{M} und eine Sprache \mathbf{L} haben wir verschiedene Äquivalenzrelationen verwendet:

DFA-Zustandsäquivalenz $\sim_{\mathcal{M}} \subseteq Q \times Q$:

$q \sim_{\mathcal{M}} p$ genau dann, wenn für alle $w \in \Sigma^*$ gilt: $\delta(q, w) \in F$ gdw. $\delta(p, w) \in F$.

Anwendung: Vereinfachung von DFAs durch Quotientenbildung

Nerode-Rechtskongruenz $\simeq_{\mathbf{L}} \subseteq \Sigma^* \times \Sigma^*$:

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

Anwendung: automatenunabhängige Analyse von Sprachen; alternative Konstruktion eines Minimalautomaten

DFA-Präfixäquivalenz $\simeq_{\mathcal{M}} \subseteq \Sigma^* \times \Sigma^*$:

$u \simeq_{\mathcal{M}} v$ genau dann, wenn gilt: $\delta(q_0, u) = \delta(q_0, v)$.

Anwendung: Beweis der Endlichkeit des $\simeq_{\mathbf{L}}$ -Indexes regulärer Sprachen

Der Satz von Myhill und Nerode

Das vorige Resultat kann noch verstärkt werden:

Satz (Myhill & Nerode):

Eine Sprache L ist genau dann regulär, wenn \simeq_L endlich viele Äquivalenzklassen hat.

Der Satz von Myhill und Nerode

Das vorige Resultat kann noch verstärkt werden:

Satz (Myhill & Nerode):

Eine Sprache L ist genau dann regulär, wenn \approx_L endlich viele Äquivalenzklassen hat.

Beweis: Die Richtung „ \Rightarrow “ haben wir soeben gezeigt.

Der Satz von Myhill und Nerode

Das vorige Resultat kann noch verstärkt werden:

Satz (Myhill & Nerode):

Eine Sprache L ist genau dann regulär, wenn \simeq_L endlich viele Äquivalenzklassen hat.

Beweis: Die Richtung „ \Rightarrow “ haben wir soeben gezeigt.

Für die Richtung „ \Leftarrow “ zeigen wir, wie man einen DFA \mathcal{M}_L für L erhalten kann, wenn \simeq_L endlich viele Äquivalenzklassen hat.

Der Satz von Myhill und Nerode

Das vorige Resultat kann noch verstärkt werden:

Satz (Myhill & Nerode):

Eine Sprache \mathbf{L} ist genau dann regulär, wenn $\simeq_{\mathbf{L}}$ endlich viele Äquivalenzklassen hat.

Beweis: Die Richtung „ \Rightarrow “ haben wir soeben gezeigt.

Für die Richtung „ \Leftarrow “ zeigen wir, wie man einen DFA $\mathcal{M}_{\mathbf{L}}$ für \mathbf{L} erhalten kann, wenn $\simeq_{\mathbf{L}}$ endlich viele Äquivalenzklassen hat.

Der DFA $\mathcal{M}_{\mathbf{L}} = \langle Q, \Sigma, \delta, q_0, F \rangle$ ist wie folgt definiert:

- $Q = \{[w]_{\simeq} \mid w \in \Sigma^*\}$ ist die (endliche) Menge der \simeq -Äquivalenzklassen
- $q_0 = [\epsilon]_{\simeq}$
- $F = \{[w]_{\simeq} \mid w \in \mathbf{L}\}$
- $\delta([w]_{\simeq}, \mathbf{a}) = [w\mathbf{a}]_{\simeq}$

Anmerkung: Diese Definition von F und δ ist zulässig, weil sie nicht vom gewählten Repräsentanten w abhängt. (\simeq ist eine Rechtskongruenz.)

Satz von Myhill und Nerode: Beweis

Beweis (Fortsetzung): Wir müssen noch zeigen, dass $L = L(\mathcal{M}_L)$.

$$w \in L(\mathcal{M}_L) \text{ gdw. } \delta(q_0, w) \in F$$

Satz von Myhill und Nerode: Beweis

Beweis (Fortsetzung): Wir müssen noch zeigen, dass $L = L(\mathcal{M}_L)$.

$$\begin{aligned}w \in L(\mathcal{M}_L) &\text{ gdw. } \delta(q_0, w) \in F \\ &\text{ gdw. } \delta([\epsilon]_{\approx}, w) \in F\end{aligned}$$

Satz von Myhill und Nerode: Beweis

Beweis (Fortsetzung): Wir müssen noch zeigen, dass $\mathbf{L} = \mathbf{L}(\mathcal{M}_{\mathbf{L}})$.

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

gdw. $\delta([\epsilon]_{\simeq}, w) \in F$

gdw. $[w]_{\simeq} \in F$

wegen $\delta([u], v) = [uv]$

(Induktion über $|v|$)

Satz von Myhill und Nerode: Beweis

Beweis (Fortsetzung): Wir müssen noch zeigen, dass $\mathbf{L} = \mathbf{L}(\mathcal{M}_{\mathbf{L}})$.

$$\begin{aligned}w \in \mathbf{L}(\mathcal{M}_{\mathbf{L}}) & \text{ gdw. } \delta(q_0, w) \in F \\ & \text{ gdw. } \delta([\epsilon]_{\approx}, w) \in F \\ & \text{ gdw. } [w]_{\approx} \in F \quad \text{wegen } \delta([u], v) = [uv] \\ & \quad \text{(Induktion über } |v|) \\ & \text{ gdw. } w \in \mathbf{L}\end{aligned}$$

□

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$L = \{w \mid w \text{ enthält kein Infix } ab\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

erlaubte Suffixe w

Klasse C

mit $C \cdot \{w\} \subseteq L$

C als Menge

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

erlaubte Suffixe w

Klasse C

mit $C \cdot \{w\} \subseteq \mathbf{L}$

C als Menge

$[\epsilon]_{\approx}$

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } \mathbf{a}\}$

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } \mathbf{a}\}$
$[\mathbf{a}]_{\approx}$		

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } \mathbf{a}\}$
$[\mathbf{a}]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } \mathbf{b}$	

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } \mathbf{a}\}$
$[\mathbf{a}]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } \mathbf{b}$	$\{v \in \mathbf{L} \mid v \text{ endet auf } \mathbf{a}\}$

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } \mathbf{a}\}$
$[\mathbf{a}]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } \mathbf{b}$	$\{v \in \mathbf{L} \mid v \text{ endet auf } \mathbf{a}\}$
$[\mathbf{ab}]_{\approx}$		

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } \mathbf{a}\}$
$[\mathbf{a}]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } \mathbf{b}$	$\{v \in \mathbf{L} \mid v \text{ endet auf } \mathbf{a}\}$
$[\mathbf{ab}]_{\approx}$	keine	

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache $\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } \mathbf{a}\}$
$[\mathbf{a}]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } \mathbf{b}$	$\{v \in \mathbf{L} \mid v \text{ endet auf } \mathbf{a}\}$
$[\mathbf{ab}]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$

Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } b$	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$



Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } b$	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$

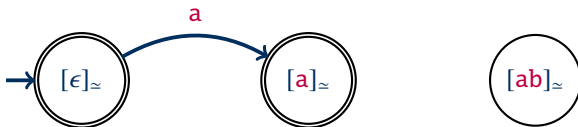


Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}, w \text{ beginnt nicht mit } b$	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$

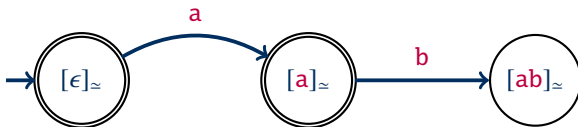


Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache

$\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

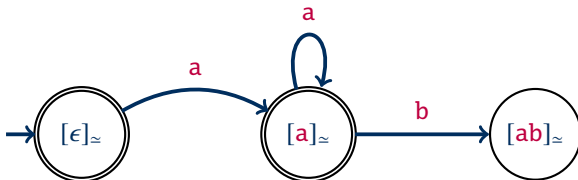
Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}, w$ beginnt nicht mit b	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$



Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache $\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

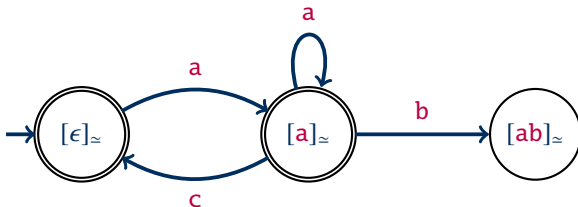
Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}$, w beginnt nicht mit b	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$



Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache $\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

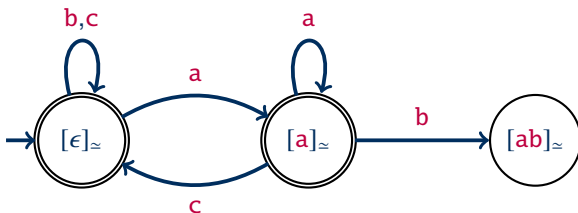
Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}, w$ beginnt nicht mit b	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$



Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache $\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

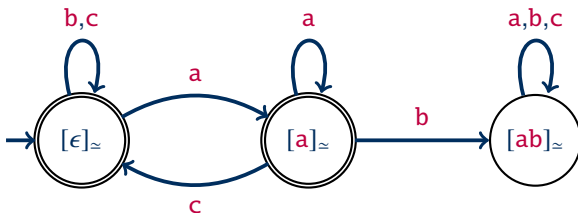
Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}, w$ beginnt nicht mit b	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$



Beispiel

Wir betrachten die Konstruktion von Myhill & Nerode für die Sprache $\mathbf{L} = \{w \mid w \text{ enthält kein Infix } \mathbf{ab}\}$ über dem Alphabet $\Sigma = \{a, b, c\}$:

Klasse C	erlaubte Suffixe w mit $C \cdot \{w\} \subseteq \mathbf{L}$	C als Menge
$[\epsilon]_{\approx}$	$w \in \mathbf{L}$	$\{v \in \mathbf{L} \mid v \text{ endet nicht auf } a\}$
$[a]_{\approx}$	$w \in \mathbf{L}, w$ beginnt nicht mit b	$\{v \in \mathbf{L} \mid v \text{ endet auf } a\}$
$[ab]_{\approx}$	keine	$\{v \mid v \notin \mathbf{L}\}$



Quiz: Konstruktion von Myhill und Nerode

$\simeq_{\mathbf{L}} \subseteq \Sigma^* \times \Sigma^*$: $u \simeq_{\mathbf{L}} v$ genau dann, wenn für alle $w \in \Sigma^*$ gilt: $uw \in \mathbf{L}$ gdw. $vw \in \mathbf{L}$.

Quiz: Wir betrachten das Alphabet $\Sigma = \{0, 1\}$ und darüber die formale Sprache ...

$\mathcal{M}_{\mathbf{L}}$ ist minimal

Für reguläre \mathbf{L} haben wir bereits gezeigt:

- $\mathcal{M}_{\mathbf{L}}$ ist ein DFA, da er insbesondere endlich viele Zustände hat (Index von \simeq).
- $\mathbf{L}(\mathcal{M}_{\mathbf{L}}) = \mathbf{L}$

Als nächstes interessiert uns der Bezug zum Minimalautomaten:

$\mathcal{M}_{\mathbf{L}}$ ist minimal

Für reguläre \mathbf{L} haben wir bereits gezeigt:

- $\mathcal{M}_{\mathbf{L}}$ ist ein DFA, da er insbesondere endlich viele Zustände hat (Index von \simeq).
- $\mathbf{L}(\mathcal{M}_{\mathbf{L}}) = \mathbf{L}$

Als nächstes interessiert uns der Bezug zum Minimalautomaten:

Satz: $\mathcal{M}_{\mathbf{L}}$ hat unter allen totalen DFAs, die \mathbf{L} erkennen, eine minimale Anzahl an Zuständen.

$\mathcal{M}_{\mathbf{L}}$ ist minimal

Für reguläre \mathbf{L} haben wir bereits gezeigt:

- $\mathcal{M}_{\mathbf{L}}$ ist ein DFA, da er insbesondere endlich viele Zustände hat (Index von \simeq).
- $\mathbf{L}(\mathcal{M}_{\mathbf{L}}) = \mathbf{L}$

Als nächstes interessiert uns der Bezug zum Minimalautomaten:

Satz: $\mathcal{M}_{\mathbf{L}}$ hat unter allen totalen DFAs, die \mathbf{L} erkennen, eine minimale Anzahl an Zuständen.

Beweis: Sei \mathcal{M} ein beliebiger totaler DFA mit $\mathbf{L}(\mathcal{M}) = \mathbf{L}$.

- Wir haben bereits gezeigt: $\simeq_{\mathcal{M}} \subseteq \simeq_{\mathbf{L}}$.
- Daraus folgerten wir:
„Zahl der $\simeq_{\mathbf{L}}$ -Äquivalenzklassen \leq Anzahl der erreichbaren Zustände von \mathcal{M} .“
- Die Zahl der $\simeq_{\mathbf{L}}$ -Äquivalenzklassen ist aber genau die Anzahl der Zustände von $\mathcal{M}_{\mathbf{L}}$.
- Also hat \mathcal{M} mindestens so viele erreichbare Zustände, wie $\mathcal{M}_{\mathbf{L}}$ Zustände hat. \square

Weitere Eigenschaften von $\mathcal{M}_{\mathbf{L}}$

Für reguläre \mathbf{L} haben wir bereits gezeigt:

- $\mathcal{M}_{\mathbf{L}}$ ist ein DFA, da er insbesondere endlich viele Zustände hat (Index von \simeq).
- $\mathbf{L}(\mathcal{M}_{\mathbf{L}}) = \mathbf{L}$
- Jeder Zustand $q \in Q$ von $\mathcal{M}_{\mathbf{L}}$ ist vom Startzustand aus erreichbar:

Für jedes $w \in \Sigma^* \neq \emptyset$ gilt $\delta(q_0, w) = \delta([\epsilon]_{\simeq}, w) = [w]_{\simeq} = q$.

- $\simeq_{\mathcal{M}_{\mathbf{L}}} = \simeq_{\mathbf{L}}$

Weitere Eigenschaften von $\mathcal{M}_{\mathbf{L}}$

Für reguläre \mathbf{L} haben wir bereits gezeigt:

- $\mathcal{M}_{\mathbf{L}}$ ist ein DFA, da er insbesondere endlich viele Zustände hat (Index von \simeq).
- $\mathbf{L}(\mathcal{M}_{\mathbf{L}}) = \mathbf{L}$
- Jeder Zustand $q \in Q$ von $\mathcal{M}_{\mathbf{L}}$ ist vom Startzustand aus erreichbar:
Für jedes $w \in \Sigma^* \neq \emptyset$ gilt $\delta(q_0, w) = \delta([\epsilon]_{\simeq}, w) = [w]_{\simeq} = q$.
- $\simeq_{\mathcal{M}_{\mathbf{L}}} = \simeq_{\mathbf{L}}$ (\clubsuit)

Beweis (\clubsuit): $\simeq_{\mathcal{M}_{\mathbf{L}}} \subseteq \simeq_{\mathbf{L}}$ haben wir bereits für beliebige DFA gezeigt

$\simeq_{\mathcal{M}_{\mathbf{L}}} \supseteq \simeq_{\mathbf{L}}$ kann wie folgt gezeigt werden:

- Seien $u, v \in \Sigma^*$ beliebige Wörter mit $u \simeq_{\mathbf{L}} v$.
- Dann gilt für $\mathcal{M}_{\mathbf{L}}$: $\delta(q_0, u) = \delta([\epsilon]_{\simeq}, u) = [u]_{\simeq} = [v]_{\simeq} = \delta([\epsilon]_{\simeq}, v) = \delta(q_0, v)$.
- Also ist $u \simeq_{\mathcal{M}_{\mathbf{L}}} v$. □

Isomorphismen von Automaten

„Isomorph“ bedeutet „gleich bis auf Umbenennung von Zuständen“:

Ein **Isomorphismus** zwischen zwei DFAs $\mathcal{M}_1 = \langle Q_1, \Sigma, \delta_1, q_1, F_1 \rangle$ und $\mathcal{M}_2 = \langle Q_2, \Sigma, \delta_2, q_2, F_2 \rangle$ ist eine bijektive Funktion $f : Q_1 \rightarrow Q_2$, so dass gilt:

- $f(q_1) = q_2$
- $f(\delta_1(q, \mathbf{a})) = \delta_2(f(q), \mathbf{a})$ für alle $\mathbf{a} \in \Sigma$
- $\{f(q) \mid q \in F_1\} = F_2$

Zwei Automaten sind **isomorph**, wenn es einen Isomorphismus zwischen ihnen gibt.

Man kann leicht zeigen, dass isomorphe Automaten die selbe Sprache akzeptieren.

Quiz: Isomorphismen von Automaten

Ein **Isomorphismus** zwischen zwei DFAs $\mathcal{M}_1 = \langle Q_1, \Sigma, \delta_1, q_1, F_1 \rangle$ und $\mathcal{M}_2 = \langle Q_2, \Sigma, \delta_2, q_2, F_2 \rangle$ ist eine bijektive Funktion $f : Q_1 \rightarrow Q_2$, so dass gilt:

$$f(q_1) = q_2, \quad f(\delta_1(q, \mathbf{a})) = \delta_2(f(q), \mathbf{a}) \text{ für alle } \mathbf{a} \in \Sigma, \quad \{f(q) \mid q \in F_1\} = F_2.$$

Zwei Automaten sind **isomorph**, wenn es einen Isomorphismus zwischen ihnen gibt.

Quiz: Gegeben seien die beiden Automaten \mathcal{M}_1 und \mathcal{M}_2 über $\Sigma = \{0, 1\}$:

...

Eindeutigkeit des Minimalautomaten

Satz: Ist \mathbf{L} eine reguläre Sprache, \mathcal{M} ein DFA mit totaler Übergangsfunktion und $\mathbf{L}(\mathcal{M}) = \mathbf{L}$, so sind der reduzierte Automat \mathcal{M}_r und der Myhill-Nerode-Minimalautomat $\mathcal{M}_{\mathbf{L}}$ isomorph.

Beweis: Sei $\mathcal{M}_{\mathbf{L}} = \langle Q, \Sigma, \delta, q_0, F \rangle$ und $\mathcal{M}_r = \langle Q_r, \Sigma, \delta_r, q_r, F_r \rangle$.

$\mathcal{M}_{\mathbf{L}} = \langle Q, \Sigma, \delta, q_0, F \rangle$ mit:

- $Q = \{[w]_{\sim} \mid w \in \Sigma^*\}$
- $q_0 = [\epsilon]_{\sim}$
- $F = \{[w]_{\sim} \mid w \in \mathbf{L}\}$
- $\delta([w]_{\sim}, \mathbf{a}) = [w\mathbf{a}]_{\sim}$

$\mathcal{M}_r = \langle Q_r, \Sigma, \delta_r, q_r, F_r \rangle$:

- (1) Entferne unerreichbare Zustände
- (2) Berechne den Quotienten-DFA bzgl. \sim

Eindeutigkeit des Minimalautomaten

Satz: Ist \mathbf{L} eine reguläre Sprache, \mathcal{M} ein DFA mit totaler Übergangsfunktion und $\mathbf{L}(\mathcal{M}) = \mathbf{L}$, so sind der reduzierte Automat \mathcal{M}_r und der Myhill-Nerode-Minimalautomat $\mathcal{M}_{\mathbf{L}}$ isomorph.

Beweis: Sei $\mathcal{M}_{\mathbf{L}} = \langle Q, \Sigma, \delta, q_0, F \rangle$ und $\mathcal{M}_r = \langle Q_r, \Sigma, \delta_r, q_r, F_r \rangle$.

$\mathcal{M}_{\mathbf{L}} = \langle Q, \Sigma, \delta, q_0, F \rangle$ mit:

- $Q = \{[w]_{\sim} \mid w \in \Sigma^*\}$
- $q_0 = [\epsilon]_{\sim}$
- $F = \{[w]_{\sim} \mid w \in \mathbf{L}\}$
- $\delta([w]_{\sim}, a) = [w\mathbf{a}]_{\sim}$

$\mathcal{M}_r = \langle Q_r, \Sigma, \delta_r, q_r, F_r \rangle$:

- (1) Entferne unerreichbare Zustände
- (2) Berechne den Quotienten-DFA bzgl. \sim

Wir definieren eine Funktion $f : Q_r \rightarrow Q$ wie folgt:

Für $q \in Q_r$ wähle ein Wort w mit $\delta_r(q_r, w) = q$ und setze $f(q) = [w]_{\sim}$.

Eindeutigkeit des Minimalautomaten (2)

Für $q \in Q_r$ wähle ein Wort w mit $\delta_r(q_r, w) = q$ und setze $f(q) = [w]_{\approx}$.

Ist f wohldefiniert?

Eindeutigkeit des Minimalautomaten (2)

Für $q \in Q_r$ wähle ein Wort w mit $\delta_r(q_r, w) = q$ und setze $f(q) = [w]_{\approx}$.

Ist f wohldefiniert?

- Für jeden Zustand $q \in Q_r$ gibt es ein geeignetes Wort w , da alle Zustände in \mathcal{M}_r vom Startzustand q_r aus erreichbar sind.

Eindeutigkeit des Minimalautomaten (2)

Für $q \in Q_r$ wähle ein Wort w mit $\delta_r(q_r, w) = q$ und setze $f(q) = [w]_{\simeq}$.

Ist f wohldefiniert?

- Für jeden Zustand $q \in Q_r$ gibt es ein geeignetes Wort w , da alle Zustände in \mathcal{M}_r vom Startzustand q_r aus erreichbar sind.
- Die konkrete Wahl von w ist dabei unerheblich:
 - Wenn $\delta_r(q_r, w) = q = \delta_r(q_r, w')$,
 - dann $w \simeq_{\mathcal{M}_r} w'$,
 - dann $w \simeq_{\mathbf{L}} w'$ (wie zuvor gezeigt),
 - dann $[w]_{\simeq} = [w']_{\simeq}$.

$\leadsto f$ ist wohldefiniert.

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “:

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\approx} = q_0$.

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\sim} = q_0$.

(2) „ $f(\delta_r(q, \mathbf{a})) = \delta(f(q), \mathbf{a})$ “:

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\approx} = q_0$.

(2) „ $f(\delta_r(q, \mathbf{a})) = \delta(f(q), \mathbf{a})$ “:

– Sei w ein Wort mit $\delta_r(q_r, w) = q$.

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\approx} = q_0$.

(2) „ $f(\delta_r(q, \mathbf{a})) = \delta(f(q), \mathbf{a})$ “:

– Sei w ein Wort mit $\delta_r(q_r, w) = q$.

– Dann ist $\delta_r(q_r, w\mathbf{a}) = \delta_r(q, \mathbf{a})$ und daher $f(\delta_r(q, \mathbf{a})) = [w\mathbf{a}]_{\approx}$.

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\simeq} = q_0$.

(2) „ $f(\delta_r(q, \mathbf{a})) = \delta(f(q), \mathbf{a})$ “:

- Sei w ein Wort mit $\delta_r(q_r, w) = q$.
- Dann ist $\delta_r(q_r, w\mathbf{a}) = \delta_r(q, \mathbf{a})$ und daher $f(\delta_r(q, \mathbf{a})) = [w\mathbf{a}]_{\simeq}$.
- Andererseits gilt $f(q) = [w]_{\simeq}$ und daher auch $\delta(f(q), \mathbf{a}) = [w\mathbf{a}]_{\simeq}$.

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\simeq} = q_0$.

(2) „ $f(\delta_r(q, \mathbf{a})) = \delta(f(q), \mathbf{a})$ “:

– Sei w ein Wort mit $\delta_r(q_r, w) = q$.

– Dann ist $\delta_r(q_r, w\mathbf{a}) = \delta_r(q, \mathbf{a})$ und daher $f(\delta_r(q, \mathbf{a})) = [w\mathbf{a}]_{\simeq}$.

– Andererseits gilt $f(q) = [w]_{\simeq}$ und daher auch $\delta(f(q), \mathbf{a}) = [w\mathbf{a}]_{\simeq}$.

(3) „ $\{f(q) \mid q \in F_r\} = F$ “:

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\simeq} = q_0$.

(2) „ $f(\delta_r(q, \mathbf{a})) = \delta(f(q), \mathbf{a})$ “:

– Sei w ein Wort mit $\delta_r(q_r, w) = q$.

– Dann ist $\delta_r(q_r, w\mathbf{a}) = \delta_r(q, \mathbf{a})$ und daher $f(\delta_r(q, \mathbf{a})) = [w\mathbf{a}]_{\simeq}$.

– Andererseits gilt $f(q) = [w]_{\simeq}$ und daher auch $\delta(f(q), \mathbf{a}) = [w\mathbf{a}]_{\simeq}$.

(3) „ $\{f(q) \mid q \in F_r\} = F$ “: Übung

Eindeutigkeit des Minimalautomaten (3)

Es bleibt zu zeigen: f ist ein Isomorphismus von \mathcal{M}_r zu \mathcal{M}_L .

(1) „ $f(q_r) = q_0$ “: Es gilt $\delta_r(q_r, \epsilon) = q_r$ und daher $f(q_r) = [\epsilon]_{\simeq} = q_0$.

(2) „ $f(\delta_r(q, \mathbf{a})) = \delta(f(q), \mathbf{a})$ “:

– Sei w ein Wort mit $\delta_r(q_r, w) = q$.

– Dann ist $\delta_r(q_r, w\mathbf{a}) = \delta_r(q, \mathbf{a})$ und daher $f(\delta_r(q, \mathbf{a})) = [w\mathbf{a}]_{\simeq}$.

– Andererseits gilt $f(q) = [w]_{\simeq}$ und daher auch $\delta(f(q), \mathbf{a}) = [w\mathbf{a}]_{\simeq}$.

(3) „ $\{f(q) \mid q \in F_r\} = F$ “: Übung

□

Damit ist der Satz bewiesen:

Satz: Ist L eine reguläre Sprache, \mathcal{M} ein DFA mit totaler Übergangsfunktion und $L(\mathcal{M}) = L$, so sind der reduzierte Automat \mathcal{M}_r und der Myhill-Nerode-Minimalautomat \mathcal{M}_L isomorph.

Zusammenfassung: Eigenschaften von $\mathcal{M}_{\mathbf{L}}$

Für reguläre \mathbf{L} haben wir bereits gezeigt:

- $\mathcal{M}_{\mathbf{L}}$ ist ein DFA, da er insbesondere endlich viele Zustände hat (Index von \simeq).
- $\mathbf{L}(\mathcal{M}_{\mathbf{L}}) = \mathbf{L}$
- Jeder Zustand von $\mathcal{M}_{\mathbf{L}}$ ist vom Startzustand aus erreichbar, da $\delta([\epsilon]_{\simeq}, w) = [w]_{\simeq}$.
- $\simeq_{\mathcal{M}_{\mathbf{L}}} = \simeq_{\mathbf{L}}$
- $\mathcal{M}_{\mathbf{L}}$ ist isomorph zu jedem reduzierten Automaten für \mathbf{L} .

Daraus folgt auch:

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} .

Automaten vergleichen

Wir wissen, dass reduzierte Automaten isomorph sind, wenn sie die selbe Sprache akzeptieren.

Damit können wir Äquivalenz zweier Automaten testen:

Eingabe: Zwei endliche Automaten \mathcal{M}_1 und \mathcal{M}_2

Ausgabe: Ist $\mathbf{L}(\mathcal{M}_1) = \mathbf{L}(\mathcal{M}_2)$?

- Transformiere \mathcal{M}_1 und \mathcal{M}_2 falls nötig in DFAs mit totaler Übergangsfunktion
- Bestimme die reduzierten Automaten
- Teste, ob die reduzierten Automaten isomorph sind (z.B. naiv durch systematisches Durchprobieren aller Bijektionen)

Auf ähnlichem Weg kann man auch reguläre Ausdrücke und reguläre Grammatiken vergleichen. Es gibt allerdings zum Teil effizientere Verfahren . . .

Minimieren ohne totale Übergänge?

Alle Ergebnisse zu Minimalautomaten gehen von DFAs mit **totaler** Übergangsfunktion aus.

Diskussion:

- Jeder DFA kann leicht in einen totalen umgewandelt werden (ein zusätzlicher Fangzustand).
- Der Fangzustand entspricht der Nerode-Kongruenzklasse $\{w \mid w \text{ ist kein Präfix eines Wortes in } \mathbf{L}\}$.
- In minimalen Automaten gibt es daher maximal einen entsprechenden Fangzustand, den man weglassen könnte, ohne die akzeptierte Sprache zu ändern.¹
- Man kann DFAs ohne totale Übergangsfunktion auch direkt reduzieren, wobei der Algorithmus zur Bestimmung von \sim ausgeführt wird, als ob man einen Fangzustand definiert hätte.

¹Ausnahme: Der DFA der leeren Sprache benötigt diese (einzige!) Klasse.

Minimale NFAs?

Alle Ergebnisse zu Minimalautomaten gehen von DFAs aus . . .

~> NFAs könnten viel kleiner sein als minimale DFAs!

Für NFAs gibt es keine vergleichbar elegante Art der Minimierung.

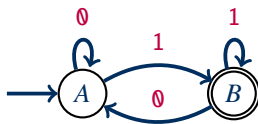
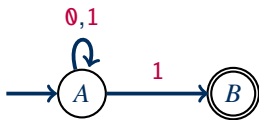
Minimale NFAs?

Alle Ergebnisse zu Minimalautomaten gehen von DFAs aus ...

~> NFAs könnten viel kleiner sein als minimale DFAs!

Für NFAs gibt es keine vergleichbar elegante Art der Minimierung.

Beispiel: Die folgenden beiden NFAs akzeptieren die selbe Sprache („Wörter die auf 1 enden“) und sind von minimaler Größe, aber sie sind nicht isomorph:



- NFA-Minimierung ist algorithmisch aufwändiger
- ... und liefert kein eindeutiges Ergebnis,
- ... ist aber durchaus praktisch machbar.

Sind NFAs wirklich kompakter?

Beispiel aus Vorlesung 4: $L_n = \{0, 1\}^* 1 \{0, 1\}^{n-1}$ („Wörter mit 1 an n -letzter Stelle“)

Wir haben gesehen, dass es dafür NFAs der Größe $n + 1$ gibt.

DFAs benötigen dagegen mindestens 2^n Zustände.

Sind NFAs wirklich kompakter?

Beispiel aus Vorlesung 4: $L_n = \{0, 1\}^* 1 \{0, 1\}^{n-1}$ („Wörter mit 1 an n -letzter Stelle“)

Wir haben gesehen, dass es dafür NFAs der Größe $n + 1$ gibt.

DFA's benötigen dagegen mindestens 2^n Zustände.

Beweis: Mit Hilfe von Myhill/Nerode:

- Die Kongruenzklassen haben die Form $[w]_{\approx}$ mit $w \in \{0, 1\}^n$.
- Jede dieser Klassen ist unterschiedlich, z.B. für $n = 3$:
wenn $v \in [011]$ dann $v \notin L_3$, $v0 \in L_3$ und $v00 \in L_3$.
wenn $v \in [101]$ dann $v \in L_3$, $v0 \notin L_3$ und $v00 \in L_3$.
...
- Es gibt also mindestens 2^n Kongruenzklassen, daher mindestens 2^n Zustände im Minimal-DFA.

Ausblick: Nichtreguläre Sprachen

Nichtreguläre Sprachen

Sind alle Sprachen regulär?

Nichtreguläre Sprachen

Sind alle Sprachen regulär?

Sicher nicht – dazu gibt es zu viele Sprachen.

Nichtreguläre Sprachen

Sind alle Sprachen regulär?

Sicher nicht – dazu gibt es zu viele Sprachen.

Sind alle Typ-0-Sprachen regulär?

Nichtreguläre Sprachen

Sind alle Sprachen regulär?

Sicher nicht – dazu gibt es zu viele Sprachen.

Sind alle Typ-0-Sprachen regulär?

Nein, auch das gilt nicht.

Wie aber zeigt man das?

- Behauptung „Sprache L ist regulär!“ \leadsto Es genügt, **einen** Automaten, regulären Ausdruck oder eine Typ-3-Grammatik für L anzugeben.
- Behauptung „Sprache L ist nicht regulär!“ \leadsto Man müsste zeigen, dass es **keinen** Automaten, **keinen** regulären Ausdruck bzw. **keine** Typ-3-Grammatik für L gibt.

Zusammenfassung und Ausblick

Der **Satz von Myhill und Nerode** charakterisiert reguläre Sprachen und liefert einen direkt konstruierten DFA für eine Sprache.

Der **reduzierte DFA** ist minimal und eindeutig.

NFAs können viel kleiner sein als minimale DFAs, aber ihre Minimierung ist viel komplizierter.

Offene Fragen:

- Wie findet man nichtreguläre Sprachen?
- Welche weiteren Berechnungsaufgaben gibt es im Zusammenhang mit regulären Sprachen?
- Und was ist mit kontextfreien Sprachen?