F2 — Automaten und formale Sprachen

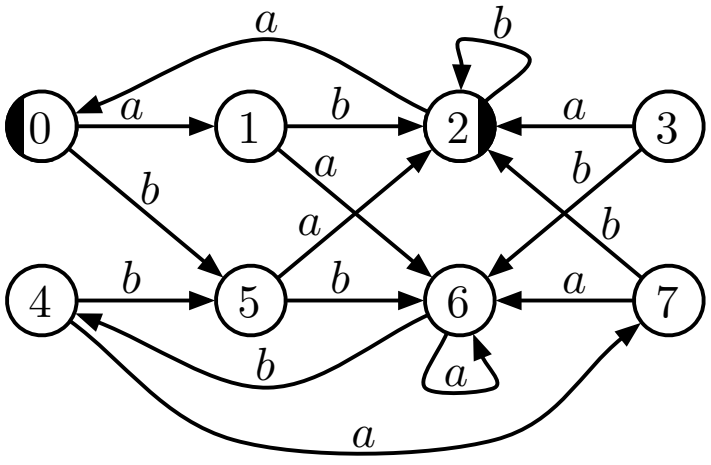
Berndt Farwer

Fachbereich Informatik
AB „Theoretische Grundlagen der Informatik“ (TGI)
Universität Hamburg
farwer@informatik.uni-hamburg.de

Themen

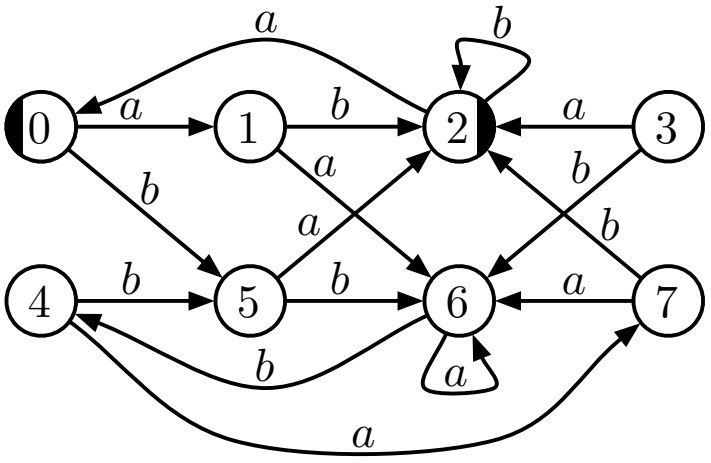
- Für die heutige Vorlesung geplant:
  - Beweis zum Färbealgorithmus
  - Homomorphismen
  - Substitutionen
  - initiale Zusammenhangskomponente
  - Grenzen der regulären Sprachen

Beispiel: äquivalente Zustände



0,1	0,2	0,3	0,4	0,5	0,6	0,7
	1,2	1,3	1,4	1,5	1,6	1,7
		2,3	2,4	2,5	2,6	2,7
			3,4	3,5	3,6	3,7
				4,5	4,6	4,7
					5,6	5,7
						6,7

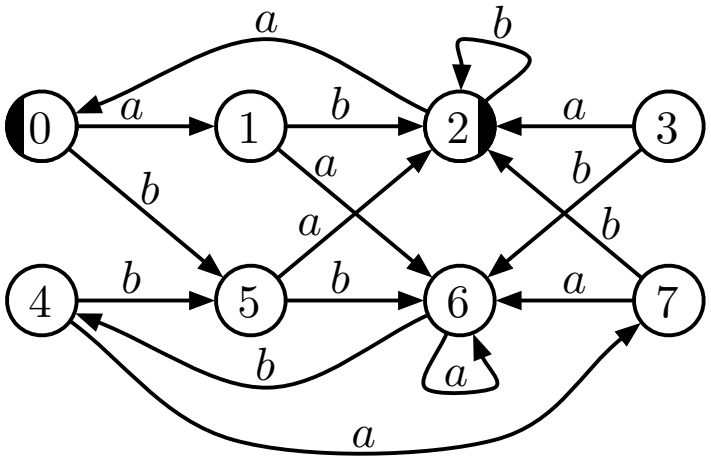
Beispiel: äquivalente Zustände



0,1	0,2	0,3	0,4	0,5	0,6	0,7
	1,2	1,3	1,4	1,5	1,6	1,7
		2,3	2,4	2,5	2,6	2,7
			3,4	3,5	3,6	3,7
				4,5	4,6	4,7
					5,6	5,7
						6,7

Situation nach der Initialisierung

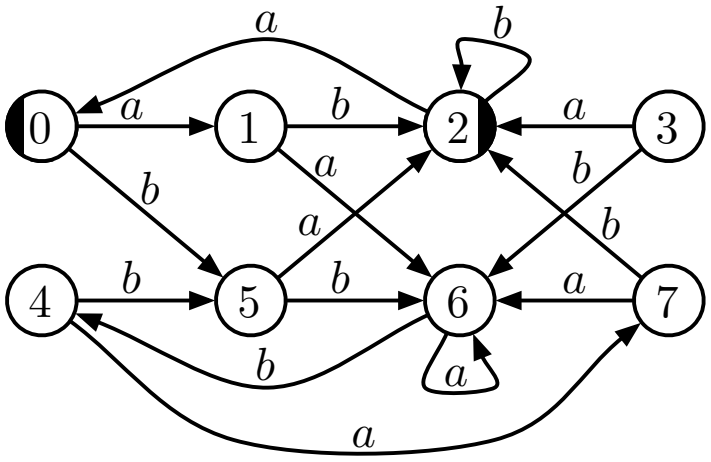
Beispiel: äquivalente Zustände

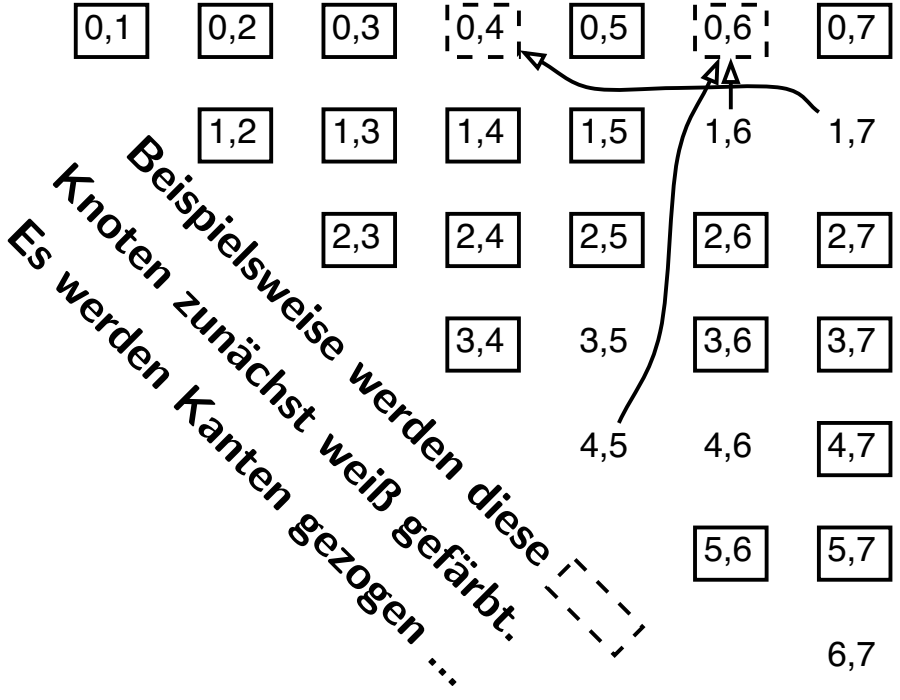


0,1	0,2	0,3	0,4	0,5	0,6	0,7
	1,2	1,3	1,4	1,5	1,6	1,7
		2,3	2,4	2,5	2,6	2,7
			3,4	3,5	3,6	3,7
				4,5	4,6	4,7
					5,6	5,7
						6,7

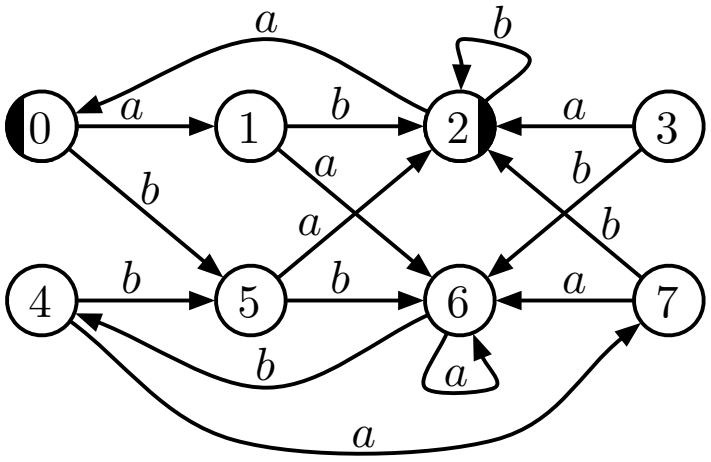
u.a. werden diese Mengen direkt schwarz gefärbt.

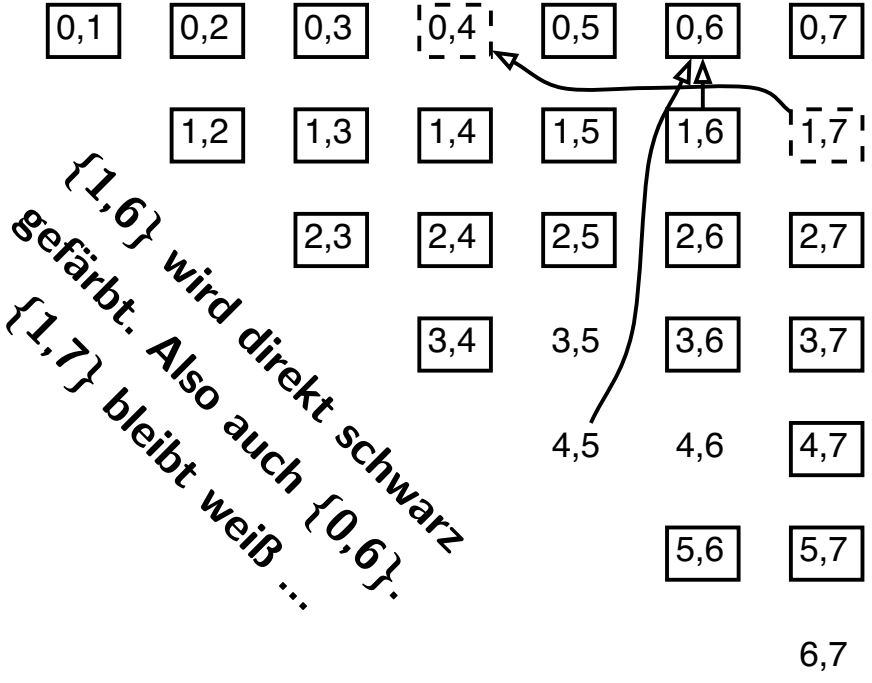
Beispiel: äquivalente Zustände



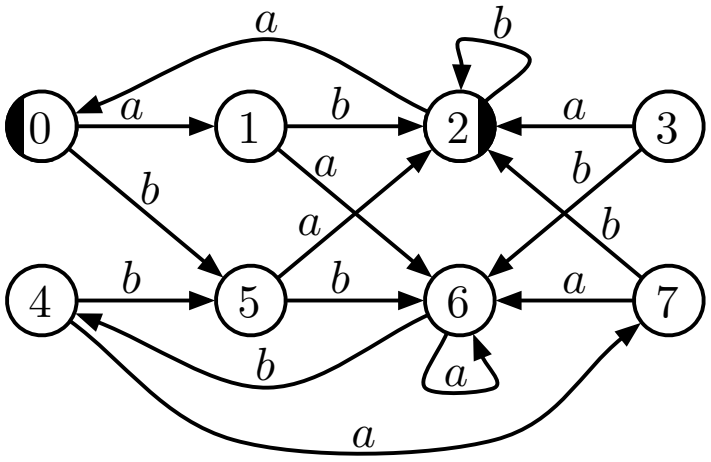


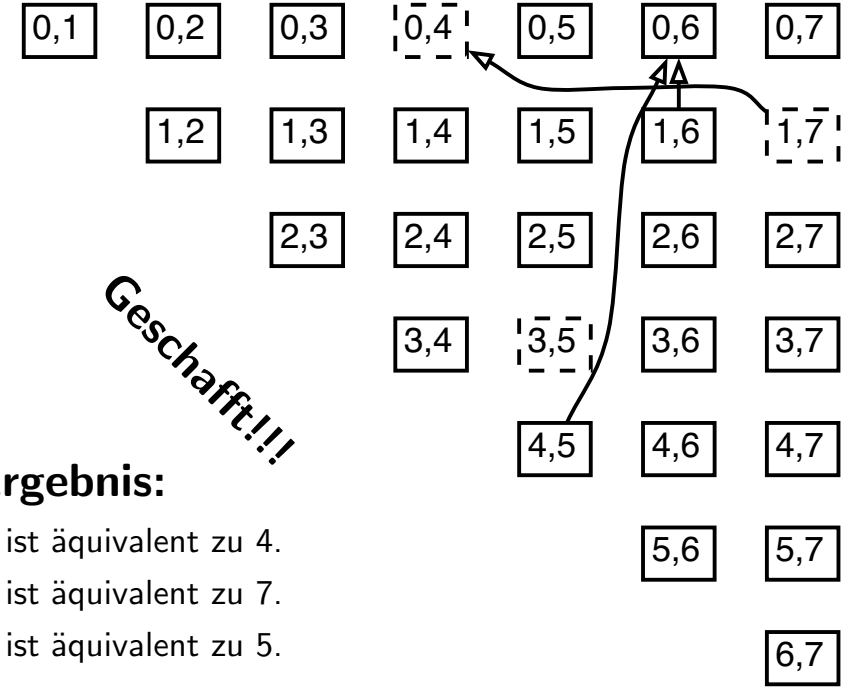
Beispiel: äquivalente Zustände





Beispiel: äquivalente Zustände





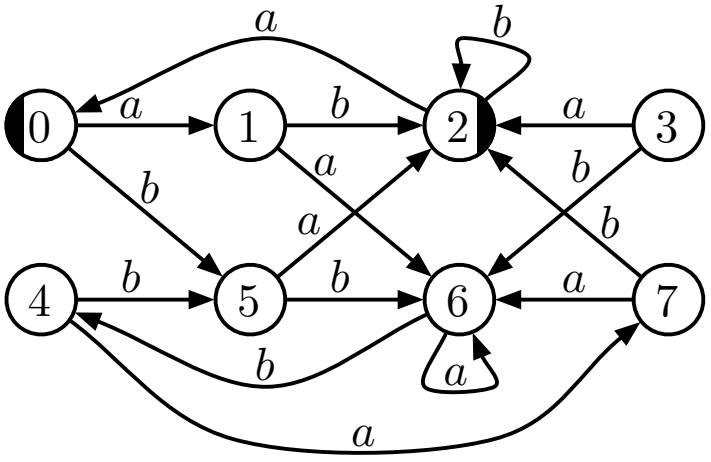
Ergebnis:

0 ist äquivalent zu 4.

1 ist äquivalent zu 7.

3 ist äquivalent zu 5.

Beispiel: äquivalente Zustände

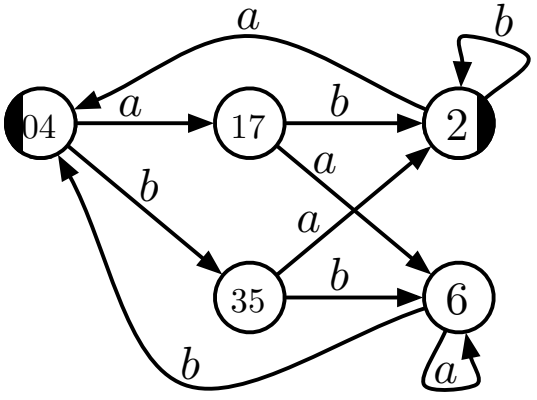


Ergebnis:

0 ist äquivalent zu 4.

1 ist äquivalent zu 7.

3 ist äquivalent zu 5.



Beweis

1.) **Termination:**
Die Schleife terminiert, denn jeder Knoten wird **höchstens einmal** *weiss* und höchstens einmal *schwarz* gefärbt.

Beweis

1.) **Termination:**
Die Schleife terminiert, denn jeder Knoten wird **höchstens einmal** *weiss* und höchstens einmal *schwarz* gefärbt.

2.) **Korrektheit:**

Beweis

1.) **Termination:**
Die Schleife terminiert, denn jeder Knoten wird **höchstens einmal** *weiss* und höchstens einmal *schwarz* gefärbt.

2.) **Korrektheit:**

A: $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

Beweis

1.) **Termination:**
Die Schleife terminiert, denn jeder Knoten wird **höchstens einmal** *weiss* und höchstens einmal *schwarz* gefärbt.

2.) **Korrektheit:**

A: $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

B: $p \not\equiv q$ $\Rightarrow$ $\{p, q\}$ ist am Ende *schwarz*

Beweis

1.) **Termination:**
Die Schleife terminiert, denn jeder Knoten wird **höchstens einmal** *weiss* und höchstens einmal *schwarz* gefärbt.

2.) **Korrektheit:**

A: $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

B: $p \not\equiv q$ $\Rightarrow$ $\{p, q\}$ ist am Ende *schwarz*

Inäquivalenz zweier Zustände $(p \not\equiv q) \Rightarrow$ Existenz eines p und q **unterscheidenden Wortes** (Zeugen, *witness*) $w \in \Sigma^*$.

Beweis A

- **zu A:** $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

Beweis A

- **zu A:** $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

  Induktion über die Zahl d der Schleifendurchläufe.

Beweis A

- **zu A:** $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

  Induktion über die Zahl d der Schleifendurchläufe.

  - **Verankerung:**
    $d = 0$, Initialisierung stellt sicher, dass $\{p, q\}$ geschwärzt ist, mit unterscheidendem Wort λ.

Beweis A

- **zu A:** $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

  Induktion über die Zahl d der Schleifendurchläufe.

  - **Verankerung:**
    $d = 0$, Initialisierung stellt sicher, dass $\{p, q\}$ geschwärzt ist, mit unterscheidendem Wort λ.
  - **Induktionsannahme:**
    Sei **A)** richtig für die in den ersten Schleifendurchläufen geschwärzten Knoten.

Beweis A (Forts.)

- $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$

Beweis A (Forts.)

- $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$
  - **Induktionsschritt:** $\{p, q\}$ schwärzen, wenn

Beweis A (Forts.)

- $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$
  - **Induktionsschritt:** $\{p, q\}$ schwärzen, wenn
    1. **direkte Färbung:** $\exists x \in \Sigma : \{(p)^x, (q)^x\}$ war schon schwarz, d.h. Zeuge $w \in \Sigma^*$ unterscheidet $(p)^x$ und $(q)^x$. Also: $p \not\equiv q$.

Beweis A (Forts.)

- $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$
  - **Induktionsschritt:** $\{p, q\}$ schwärzen, wenn
    2. **indirekte Färbung:** anderer Knoten $\{s, t\}$ ist geschwärzt und Knoten $\{p, q\}$ auf Pfad von $\{s, t\}$ erreichbar.
       - Einfügen von Kanten nur falls es ein Wort $v \in \Sigma^*$ gibt, mit $s = (p)^v$ und $t = (q)^v$. Da $\{s, t\}$ nur schwarz wurde, falls ein s und t unterscheidendes Wort w existiert:

Beweis A (Forts.)

- $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$
  - **Induktionsschritt:** $\{p, q\}$ schwärzen, wenn
    2. **indirekte Färbung:** anderer Knoten $\{s, t\}$ ist geschwärzt und Knoten $\{p, q\}$ auf Pfad von $\{s, t\}$ erreichbar.
       - Einfügen von Kanten nur falls es ein Wort $v \in \Sigma^*$ gibt, mit $s = (p)^v$ und $t = (q)^v$. Da $\{s, t\}$ nur schwarz wurde, falls ein s und t unterscheidendes Wort w existiert:
       - Entweder $(s)^w$ oder $(t)^w$ ist Endzustand. Also auch entweder $((p)^v)^w$ oder $((q)^v)^w$ Endzustand. Das p und q unterscheidende Wort ist vw.

Beweis A (Forts.)

- $\{p, q\}$ ist *schwarz* $\Rightarrow$ $p \not\equiv q$
  - **Induktionsschritt:** $\{p, q\}$ schwärzen, wenn
    2. **indirekte Färbung:** anderer Knoten $\{s, t\}$ ist geschwärzt und Knoten $\{p, q\}$ auf Pfad von $\{s, t\}$ erreichbar.
       - Einfügen von Kanten nur falls es ein Wort $v \in \Sigma^*$ gibt, mit $s = (p)^v$ und $t = (q)^v$. Da $\{s, t\}$ nur schwarz wurde, falls ein s und t unterscheidendes Wort w existiert:
       - Entweder $(s)^w$ oder $(t)^w$ ist Endzustand. Also auch entweder $((p)^v)^w$ oder $((q)^v)^w$ Endzustand. Das p und q unterscheidende Wort ist vw.

Damit ist **A** durch Induktion gezeigt.

Beweis B

- **zu B:** $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*

Beweis B

- **zu B:** $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*
- Für $p \not\equiv q$ gibt es kürzestes p und q unterscheidendes Wort w.

Beweis B

- **zu B:** $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*
- Für $p \not\equiv q$ gibt es kürzestes p und q unterscheidendes Wort w.

  Induktion über die Länge k der kürzesten unterscheidenden Wörter.

Beweis B

- **zu B:** $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*
- Für $p \not\equiv q$ gibt es kürzestes p und q unterscheidendes Wort w.

  Induktion über die Länge k der kürzesten unterscheidenden Wörter.

  - **Verankerung:** $|w| = 0 \quad \Rightarrow w = \lambda$.

Beweis B

- **zu B:** $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*
- Für $p \not\equiv q$ gibt es kürzestes p und q unterscheidendes Wort w.

  Induktion über die Länge k der kürzesten unterscheidenden Wörter.

  - **Verankerung:** $|w| = 0 \quad \Rightarrow w = \lambda$.
  - Automat ist DFA: **genau einer** der beiden Zustände ist Endzustand
    $\Rightarrow$ Knoten $\{p, q\}$ bei der Initialisierung geschwärzt.

Beweis B

- **zu B:** $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*
- Für $p \not\equiv q$ gibt es kürzestes p und q unterscheidendes Wort w.

  Induktion über die Länge k der kürzesten unterscheidenden Wörter.

  - **Verankerung:** $|w| = 0 \quad \Rightarrow w = \lambda$.
  - Automat ist DFA: **genau einer** der beiden Zustände ist Endzustand
    $\Rightarrow$ Knoten $\{p, q\}$ bei der Initialisierung geschwärzt.
  - Einmal geschwärzte Knoten werden nicht wieder anders gefärbt.

Beweis B (Forts.)

- $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*

Beweis B (Forts.)

- $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*
  - **Induktionsannahme:**
    Alle Knoten $\{s, t\}$, bei denen s und t durch Wörter mit Längen bis zu k unterscheidbar waren, wurden irgendwann geschwärzt.

Beweis B (Forts.)

- $p \not\equiv q \quad \Rightarrow \quad \{p, q\}$ am Ende *schwarz*
  - **Induktionsannahme:**
    Alle Knoten $\{s, t\}$, bei denen s und t durch Wörter mit Längen bis zu k unterscheidbar waren, wurden irgendwann geschwärzt.
  - **Induktionsschritt:**
    Sei $p \not\equiv q$ unterscheidbar mit dem kürzesten Wort $v = aw$ mit $|v| = k + 1$ und $a \in X$, $w \in \Sigma^*$.

Beweis B (Forts.)

- w unterscheidet die Zustände $(p)^a$ und $(q)^a$

Beweis B (Forts.)

- w unterscheidet die Zustände $(p)^a$ und $(q)^a$
- $\{(p)^a, (q)^a\}$ wird gem. Induktionsannahme irgendwann *schwarz*. Irgendwann wird $\{p, q\}$ als nicht gefärbter Knoten ausgewählt.

Beweis B (Forts.)

- w unterscheidet die Zustände $(p)^a$ und $(q)^a$
- $\{(p)^a, (q)^a\}$ wird gem. Induktionsannahme irgendwann *schwarz*. Irgendwann wird $\{p, q\}$ als nicht gefärbter Knoten ausgewählt.
- Wenn für irgend ein $x \in \Sigma$ ($x = a$ ist dabei möglich) der Knoten $\{(p)^x, (q)^x\}$ schon *schwarz* ist, so wird $\{p, q\}$ direkt *schwarz* gefärbt.

Beweis B (Forts.)

- w unterscheidet die Zustände $(p)^a$ und $(q)^a$
- $\{(p)^a, (q)^a\}$ wird gem. Induktionsannahme irgendwann *schwarz*. Irgendwann wird $\{p, q\}$ als nicht gefärbter Knoten ausgewählt.
- Wenn für irgend ein $x \in \Sigma$ ($x = a$ ist dabei möglich) der Knoten $\{(p)^x, (q)^x\}$ schon *schwarz* ist, so wird $\{p, q\}$ direkt *schwarz* gefärbt.
- Falls $\{(p)^x, (q)^x\}$ für kein $x \in \Sigma$ *schwarz* ist, werden Kanten $\{(p)^x, (q)^x\} \longrightarrow \{p, q\}$ für jedes $x \in \Sigma$ gezeichnet.

Komplexität des Algorithmus

- **Theorem:** Der Färbealgorithmus zum Auffinden äquivalenter Zustände arbeitet in $c \cdot |\Sigma| \cdot |Z|^2$ Schritten, wobei c eine Konstante ist.

Komplexität des Algorithmus

- **Theorem:** Der Färbealgorithmus zum Auffinden äquivalenter Zustände arbeitet in $c \cdot |\Sigma| \cdot |Z|^2$ Schritten, wobei c eine Konstante ist.
- **Beweisskizze:**

Komplexität des Algorithmus

- **Theorem:** Der Färbealgorithmus zum Auffinden äquivalenter Zustände arbeitet in $c \cdot |\Sigma| \cdot |Z|^2$ Schritten, wobei c eine Konstante ist.
- **Beweisskizze:**
  - Jeder Knoten wird in Schritt 1 maximal einmal betrachtet.

Komplexität des Algorithmus

- **Theorem:** Der Färbealgorithmus zum Auffinden äquivalenter Zustände arbeitet in $c \cdot |\Sigma| \cdot |Z|^2$ Schritten, wobei c eine Konstante ist.
- **Beweisskizze:**
  - Jeder Knoten wird in Schritt 1 maximal einmal betrachtet.
  - Jedesmal werden in Schritt 2 dazu $|\Sigma|$ Knoten $\{(p)^x, (q)^x\}$ untersucht.

Komplexität des Algorithmus

- **Theorem:** Der Färbealgorithmus zum Auffinden äquivalenter Zustände arbeitet in $c \cdot |\Sigma| \cdot |Z|^2$ Schritten, wobei c eine Konstante ist.
- **Beweisskizze:**
  - Jeder Knoten wird in Schritt 1 maximal einmal betrachtet.
  - Jedesmal werden in Schritt 2 dazu $|\Sigma|$ Knoten $\{(p)^x, (q)^x\}$ untersucht.
  - Für diese Knoten kann $\{p, q\}$ und alle von hier aus erreichbaren Knoten besucht werden, um diese zu schwärzen oder Kanten zu ziehen.

Komplexität des Algorithmus

- **Theorem:** Der Färbealgorithmus zum Auffinden äquivalenter Zustände arbeitet in $c \cdot |\Sigma| \cdot |Z|^2$ Schritten, wobei c eine Konstante ist.
- **Beweisskizze:**
  - Jeder Knoten wird in Schritt 1 maximal einmal betrachtet.
  - Jedesmal werden in Schritt 2 dazu $|\Sigma|$ Knoten $\{(p)^x, (q)^x\}$ untersucht.
  - Für diese Knoten kann $\{p, q\}$ und alle von hier aus erreichbaren Knoten besucht werden, um diese zu schwärzen oder Kanten zu ziehen.
  - Insgesamt ergibt sich die Zahl von höchstens $c \cdot |X| \cdot |Z|^2$ Besuchen von einzelnen Knoten.

Homomorphismen

- Das Wechseln des Alphabets und der Austausch einzelner Symbole macht aus einer regulären Menge wiederum eine reguläre Menge.

Homomorphismen

- Das Wechseln des Alphabets und der Austausch einzelner Symbole macht aus einer regulären Menge wiederum eine reguläre Menge.
- **Definition:** Eine Funktion h, für die $h(x \cdot y) = h(x) \circ h(y)$ gilt, wird strukturerhaltend oder **Homomorphismus** genannt.

Homomorphismen

- Das Wechseln des Alphabets und der Austausch einzelner Symbole macht aus einer regulären Menge wiederum eine reguläre Menge.
- **Definition:** Eine Funktion h, für die $h(x \cdot y) = h(x) \circ h(y)$ gilt, wird strukturerhaltend oder **Homomorphismus** genannt.
- **Beispiel:** Sei $\Sigma = \{a, b\}$, $\Gamma = \{b, c\}$ und $h : \Sigma^* \longrightarrow \Gamma^*$ mit

$$h(a) := c, \quad h(b) := bb.$$

Homomorphismen

- Das Wechseln des Alphabets und der Austausch einzelner Symbole macht aus einer regulären Menge wiederum eine reguläre Menge.
- **Definition:** Eine Funktion h, für die $h(x \cdot y) = h(x) \circ h(y)$ gilt, wird strukturerhaltend oder **Homomorphismus** genannt.
- **Beispiel:** Sei $\Sigma = \{a, b\}$, $\Gamma = \{b, c\}$ und $h : \Sigma^* \longrightarrow \Gamma^*$ mit

$$h(a) := c, \quad h(b) := bb.$$

  - Dann gilt: $h(abab) = h(aba)h(b) = h(aba)bb = h(a)h(b)h(a)bb = cbbcbb$.

Homomorphismen

- Das Wechseln des Alphabets und der Austausch einzelner Symbole macht aus einer regulären Menge wiederum eine reguläre Menge.
- **Definition:** Eine Funktion h, für die $h(x \cdot y) = h(x) \circ h(y)$ gilt, wird strukturerhaltend oder **Homomorphismus** genannt.
- **Beispiel:** Sei $\Sigma = \{a, b\}$, $\Gamma = \{b, c\}$ und $h : \Sigma^* \longrightarrow \Gamma^*$ mit

$$h(a) := c, \quad h(b) := bb.$$

  - Dann gilt: $h(abab) = h(aba)h(b) = h(aba)bb = h(a)h(b)h(a)bb = cbbcbb$.
  - h ist ein Homomorphismus, bei dem sowohl $\circ$ als auch $\cdot$ die Konkatenation von Wörtern ist. $(h(\lambda) = \lambda)$

Abschlussop.: Homomorphismus

- **Theorem:** Sei $L \in \mathcal{R}at(\Sigma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus. Dann ist $h(L) := \{h(v) \mid v \in L\} \in \mathcal{R}at(\Gamma)$

Abschlussop.: Homomorphismus

- **Theorem:** Sei $L \in \mathcal{R}at(\Sigma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus. Dann ist $h(L) := \{h(v) \mid v \in L\} \in \mathcal{R}at(\Gamma)$
- **Beweis:** Man ersetze die Symbole $a \in \Sigma$ in dem rationalen Ausdruck für L jeweils durch $h(a)$. Es resultiert ein rationaler Ausdruck für $h(L)$.

Abschlussop.: Homomorphismus

- **Theorem:** Sei $L \in \mathcal{R}at(\Sigma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus. Dann ist $h(L) := \{h(v) \mid v \in L\} \in \mathcal{R}at(\Gamma)$
- **Beweis:** Man ersetze die Symbole $a \in \Sigma$ in dem rationalen Ausdruck für L jeweils durch $h(a)$. Es resultiert ein rationaler Ausdruck für $h(L)$.
- Eine ähnliche Konstruktion ist auch mit NFAs möglich.

Abschlussop.: Homomorphismus

- **Theorem:** Sei $L \in \mathcal{R}at(\Sigma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus. Dann ist $h(L) := \{h(v) \mid v \in L\} \in \mathcal{R}at(\Gamma)$
- **Beweis:** Man ersetze die Symbole $a \in \Sigma$ in dem rationalen Ausdruck für L jeweils durch $h(a)$. Es resultiert ein rationaler Ausdruck für $h(L)$.
- Eine ähnliche Konstruktion ist auch mit NFAs möglich.
- Diese Idee wird nun verallgemeinert, indem für einzelne Symbole ganze Sprachen **substituiert** werden.

Substitution

- **Definition** Eine **Substitution** ist ein Homomorphismus $s : \Sigma^* \longrightarrow 2^{\Gamma^*}$, wobei Σ und Γ Alphabete sind und s folgende Eigenschaften besitzt:

Substitution

- **Definition** Eine **Substitution** ist ein Homomorphismus $s : \Sigma^* \longrightarrow 2^{\Gamma^*}$, wobei Σ und Γ Alphabete sind und s folgende Eigenschaften besitzt:

  1. Für jedes $a \in \Sigma$ ist $s(a) \subseteq \Gamma^*$ definiert.

Substitution

- **Definition** Eine **Substitution** ist ein Homomorphismus $s : \Sigma^* \longrightarrow 2^{\Gamma^*}$, wobei Σ und Γ Alphabete sind und s folgende Eigenschaften besitzt:

  1. Für jedes $a \in \Sigma$ ist $s(a) \subseteq \Gamma^*$ definiert.
  2. $s(\lambda) := \{\lambda\}$.

Substitution

- **Definition** Eine **Substitution** ist ein Homomorphismus $s : \Sigma^* \longrightarrow 2^{\Gamma^*}$, wobei Σ und Γ Alphabete sind und s folgende Eigenschaften besitzt:

  1. Für jedes $a \in \Sigma$ ist $s(a) \subseteq \Gamma^*$ definiert.
  2. $s(\lambda) := \{\lambda\}$.
  3. $\forall u, v \in \Sigma^* : s(u \cdot v) = s(u) \cdot s(v)$

Substitution

- **Definition** Eine **Substitution** ist ein Homomorphismus $s : \Sigma^* \longrightarrow 2^{\Gamma^*}$, wobei Σ und Γ Alphabete sind und s folgende Eigenschaften besitzt:

  1. Für jedes $a \in \Sigma$ ist $s(a) \subseteq \Gamma^*$ definiert.
  2. $s(\lambda) := \{\lambda\}$.
  3. $\forall u, v \in \Sigma^* : s(u \cdot v) = s(u) \cdot s(v)$

- Ist $s(a)$ regulär (bzw. endlich) für jedes $a \in \Sigma$, dann heißt s **reguläre** bzw. **endliche** Substitution.

Substitution

- **Definition** Eine **Substitution** ist ein Homomorphismus $s : \Sigma^* \longrightarrow 2^{\Gamma^*}$, wobei Σ und Γ Alphabete sind und s folgende Eigenschaften besitzt:

  1. Für jedes $a \in \Sigma$ ist $s(a) \subseteq \Gamma^*$ definiert.
  2. $s(\lambda) := \{\lambda\}$.
  3. $\forall u, v \in \Sigma^* : s(u \cdot v) = s(u) \cdot s(v)$

- Ist $s(a)$ regulär (bzw. endlich) für jedes $a \in \Sigma$, dann heißt s **reguläre** bzw. **endliche** Substitution.
- kanonische Erweiterung:

$$s(L) := \bigcup_{w \in L} s(w)$$

Beispiel: Substitution

- Seien

$$s : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto \{a\}, \quad 1 \mapsto \{b\}^*$$

$$s' : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto (ab)^+a, \quad 1 \mapsto \emptyset$$

Beispiel: Substitution

- Seien

$$s : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto \{a\}, \quad 1 \mapsto \{b\}^*$$

$$s' : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto (ab)^+a, \quad 1 \mapsto \emptyset$$

- $s(01) = s(0)s(1) = ab^* = \{a, ab, abb, abbb, \ldots\}$

Beispiel: Substitution

- Seien

$$s : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto \{a\}, \quad 1 \mapsto \{b\}^*$$

$$s' : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto (ab)^+a, \quad 1 \mapsto \emptyset$$

- $s(01) = s(0)s(1) = ab^* = \{a, ab, abb, abbb, \ldots\}$
- $s'(01) = s'(0)s'(1) = (ab)^+a \cdot \emptyset = \emptyset$

Beispiel: Substitution

- Seien

$$s : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto \{a\}, \quad 1 \mapsto \{b\}^*$$

$$s' : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto (ab)^+a, \quad 1 \mapsto \emptyset$$

- $s(01) = s(0)s(1) = ab^* = \{a, ab, abb, abbb, \ldots\}$
- $s'(01) = s'(0)s'(1) = (ab)^+a \cdot \emptyset = \emptyset$
- $s(0^*(0+1)+1^*) = a^*(a+b^*) + (b^*)^* = a^+ + a^*b^* + b^* = a^*b^*$

Beispiel: Substitution

- Seien

$$s : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto \{a\}, \quad 1 \mapsto \{b\}^*$$

$$s' : \{0,1\}^* \longrightarrow 2^{\{a,b\}^*} \text{ mit } 0 \mapsto (ab)^+a, \quad 1 \mapsto \emptyset$$

- $s(01) = s(0)s(1) = ab^* = \{a, ab, abb, abbb, \ldots\}$
- $s'(01) = s'(0)s'(1) = (ab)^+a \cdot \emptyset = \emptyset$
- $s(0^*(0+1)+1^*) = a^*(a+b^*) + (b^*)^* = a^+ + a^*b^* + b^* = a^*b^*$
- $s'(0^*(0+1)+1^*) = ((ab)^+a)^*((ab)^+a + \emptyset) + \emptyset^* = ((ab)^+a)^+ + \emptyset^* = ((ab)^+a)^*$

Abschlussop.: reg. Substitution

- **Theorem:** Die Familie der regulären Mengen ist gegenüber regulären Substitutionen abgeschlossen.

Abschlussop.: reg. Substitution

- **Theorem:** Die Familie der regulären Mengen ist gegenüber regulären Substitutionen abgeschlossen.
- **Beweis:** Jede reguläre Menge R wird durch einen rationalen Ausdruck dargestellt.

Abschlussop.: reg. Substitution

- **Theorem:** Die Familie der regulären Mengen ist gegenüber regulären Substitutionen abgeschlossen.
- **Beweis:** Jede reguläre Menge R wird durch einen rationalen Ausdruck dargestellt.
  - Ersetzt man nun jedes Symbol a in diesem Ausdruck durch den rationalen Ausdruck der $s(a)$ beschreibt, so ergibt sich ein rationaler Ausdruck für $s(R)$.

Abschlussop.: reg. Substitution

- **Theorem:** Die Familie der regulären Mengen ist gegenüber regulären Substitutionen abgeschlossen.
- **Beweis:** Jede reguläre Menge R wird durch einen rationalen Ausdruck dargestellt.
  - Ersetzt man nun jedes Symbol a in diesem Ausdruck durch den rationalen Ausdruck der $s(a)$ beschreibt, so ergibt sich ein rationaler Ausdruck für $s(R)$.
  - . . . aus dem vorigen **Beispiel:** $s(0^*(0+1)+1^*) = a^*(a+b^*)+(b^*)^*$ ist rationaler Ausdruck.

Verkürzen von Wörtern

- Substitutionen ersetzen einzelne Symbole durch Wortmengen, also in der Regel sogar durch unendlich viele Wörter, unter denen auch das leere Wort λ vorkommen darf.

Verkürzen von Wörtern

- Substitutionen ersetzen einzelne Symbole durch Wortmengen, also in der Regel sogar durch unendlich viele Wörter, unter denen auch das leere Wort λ vorkommen darf.
- Die Umkehrung, nämlich längere (Teil-)Wörter in einer Zeichenkette auf einzelne Symbole zu verkürzen, scheint zunächst nicht so leicht möglich.

Verkürzen von Wörtern

- Substitutionen ersetzen einzelne Symbole durch Wortmengen, also in der Regel sogar durch unendlich viele Wörter, unter denen auch das leere Wort λ vorkommen darf.
- Die Umkehrung, nämlich längere (Teil-)Wörter in einer Zeichenkette auf einzelne Symbole zu verkürzen, scheint zunächst nicht so leicht möglich.
- Mathematisch wird dies durch die Umkehrung eines Homomorphismus, also der Anwendung sogenannter **inverser Homomorphismen** geleistet.

inverse Homomorphismen

- **Definition:** Seien Σ und Γ endliche Alphabete sowie $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus. Die zu h inverse Funktion heißt **inverser Homomorphismus** und ist gegegen durch $h^{-1} : \Gamma^* \longrightarrow 2^{\Sigma^*}$.

inverse Homomorphismen

- **Definition:** Seien Σ und Γ endliche Alphabete sowie $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus. Die zu h inverse Funktion heißt **inverser Homomorphismus** und ist gegegen durch $h^{-1} : \Gamma^* \longrightarrow 2^{\Sigma^*}$.
- h^{-1} definiert für jedes $w \in \Gamma^*$ die Menge aller möglichen Urbilder von w durch:

$$h^{-1}(w) := \{v \in \Sigma^* \mid \exists w \in \Gamma^* : h(v) = w\}.$$

Diese Menge kann daher auch leer sein.

inverse Homomorphismen

- **Definition:** Seien Σ und Γ endliche Alphabete sowie $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus. Die zu h inverse Funktion heißt **inverser Homomorphismus** und ist gegegen durch $h^{-1} : \Gamma^* \longrightarrow 2^{\Sigma^*}$.
- h^{-1} definiert für jedes $w \in \Gamma^*$ die Menge aller möglichen Urbilder von w durch:

$$h^{-1}(w) := \{v \in \Sigma^* \mid \exists w \in \Gamma^* : h(v) = w\}.$$

  Diese Menge kann daher auch leer sein.
- Erweiterung auf Sprachen:

$$h^{-1}(L) := \bigcup_{w \in L} h^{-1}(w)$$

Beispiele: inv. Homomorphismen

- Wir definieren zwei Homomorphismen und betrachten die Anwendung auf
  (a) einzelne Wörter, (b) Sprachen.

Beispiele: inv. Homomorphismen

- Wir definieren zwei Homomorphismen und betrachten die Anwendung auf
  (a) einzelne Wörter, (b) Sprachen.
  - (a) Sei $h : \{a, b, c\}^* \longrightarrow \{x, y\}^*$ definiert durch

$$a \mapsto xyx, \quad b \mapsto xy, \quad c \mapsto yx.$$

    Dann ist $h^{-1}(xy) = \{b\}$, $h^{-1}(xx) = \emptyset$ und $h^{-1}(xyxyx) = \{ac, ba\}$.

Beispiele: inv. Homomorphismen

- Wir definieren zwei Homomorphismen und betrachten die Anwendung auf
  (a) einzelne Wörter, (b) Sprachen.
  - (a) Sei $h : \{a, b, c\}^* \longrightarrow \{x, y\}^*$ definiert durch

    $$a \mapsto xyx, \quad b \mapsto xy, \quad c \mapsto yx.$$

    Dann ist $h^{-1}(xy) = \{b\}$, $h^{-1}(xx) = \emptyset$ und $h^{-1}(xyxyx) = \{ac, ba\}$.
  - (b) Sei $f : \{a, b, c\}^* \longrightarrow \{x, y\}^*$ definiert durch

    $$a \mapsto x, \quad b \mapsto y, \quad c \mapsto \lambda.$$

    Dann ist $h^{-1}(\{\lambda\}) = \{c\}^*$, $h^{-1}(\{y\}) = \{c\}^*\{b\}\{c\}^*$ und $h^{-1}(\{x, y\}^*) = \{a, b, c\}^*$.

Abschlussoperator: inv. Hom.

- **Theorem:** Sei $L \in \mathcal{A}kz(\Gamma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus, dann ist $h^{-1}(L) \in \mathcal{A}kz(\Sigma)$.

Abschlussoperator: inv. Hom.

- **Theorem:** Sei $L \in \mathcal{A}kz(\Gamma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus, dann ist $h^{-1}(L) \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Gamma, \delta_1, z_0, Z_{\text{end}})$. Wir konstruieren einen DFA $B = (Z, \Sigma, \delta_2, z_0, Z_{\text{end}})$, der bei Eingabe von x aus Σ den vDFA A auf der Eingabe $h(x)$ simuliert.

Abschlussoperator: inv. Hom.

- **Theorem:** Sei $L \in \mathcal{A}kz(\Gamma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus, dann ist $h^{-1}(L) \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Gamma, \delta_1, z_0, Z_{\text{end}})$. Wir konstruieren einen DFA $B = (Z, \Sigma, \delta_2, z_0, Z_{\text{end}})$, der bei Eingabe von x aus Σ den vDFA A auf der Eingabe $h(x)$ simuliert.
- Es wird δ_2 definiert durch: $\forall (z, x) \in Z \times \Sigma : \delta_2(z, x) := (z)^{h(x)}$, womit in B der mit $h(x)$ in A erreichte Zustand $(z)^{h(x)}$ eingenommen wird.

Abschlussoperator: inv. Hom.

- **Theorem:** Sei $L \in \mathcal{A}kz(\Gamma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus, dann ist $h^{-1}(L) \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Gamma, \delta_1, z_0, Z_{\text{end}})$. Wir konstruieren einen DFA $B = (Z, \Sigma, \delta_2, z_0, Z_{\text{end}})$, der bei Eingabe von x aus Σ den vDFA A auf der Eingabe $h(x)$ simuliert.
- Es wird δ_2 definiert durch: $\forall (z, x) \in Z \times \Sigma : \delta_2(z, x) := (z)^{h(x)}$, womit in B der mit $h(x)$ in A erreichte Zustand $(z)^{h(x)}$ eingenommen wird.
- Ist in $L(A)$ ein Wort $v \in (h(\Sigma))^*$, so wird jedes $u \in \Sigma^*$ mit $h(u) \in (h(\Sigma))^*$ von B akzeptiert.

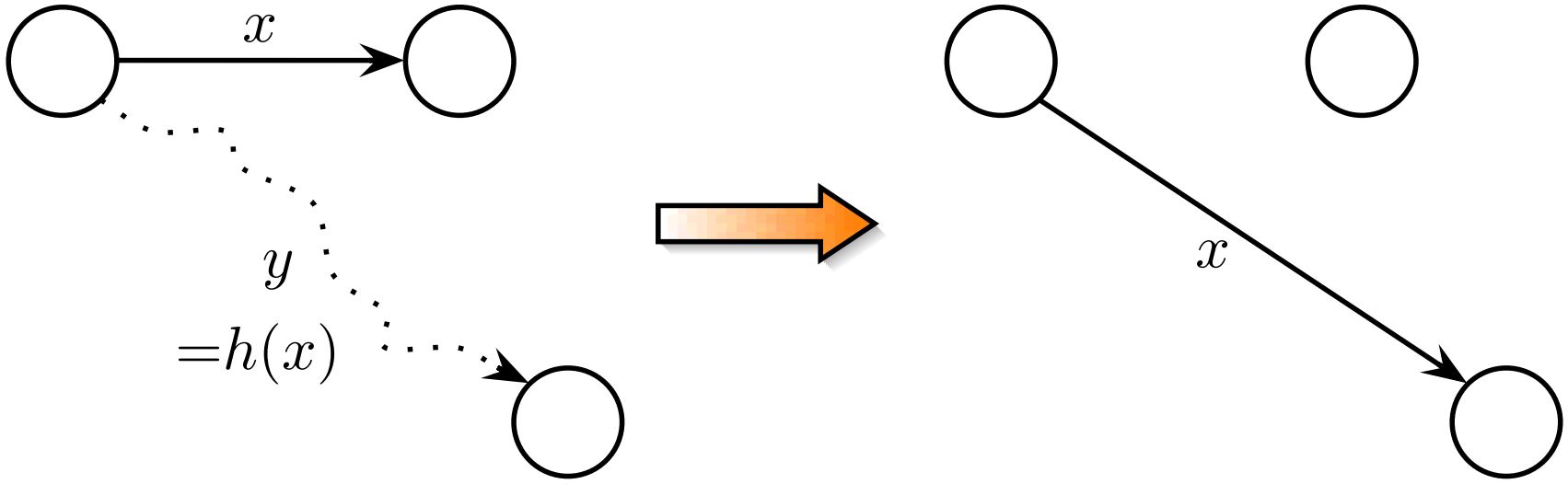
Abschlussoperator: inv. Hom.

- **Theorem:** Sei $L \in \mathcal{A}kz(\Gamma)$ und $h : \Sigma^* \longrightarrow \Gamma^*$ ein Homomorphismus, dann ist $h^{-1}(L) \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Gamma, \delta_1, z_0, Z_{\text{end}})$. Wir konstruieren einen DFA $B = (Z, \Sigma, \delta_2, z_0, Z_{\text{end}})$, der bei Eingabe von x aus Σ den vDFA A auf der Eingabe $h(x)$ simuliert.
- Es wird δ_2 definiert durch: $\forall (z, x) \in Z \times \Sigma : \delta_2(z, x) := (z)^{h(x)}$, womit in B der mit $h(x)$ in A erreichte Zustand $(z)^{h(x)}$ eingenommen wird.
- Ist in $L(A)$ ein Wort $v \in (h(\Sigma))^*$, so wird jedes $u \in \Sigma^*$ mit $h(u) \in (h(\Sigma))^*$ von B akzeptiert.
- Die Umkehrung ist ebenso einfach.

Illustration des Beweises

$$w \in L \quad \Leftrightarrow \quad h^{-1}(w) \in h^{-1}(L)$$

$$w \in (h(\Sigma)^*) \subseteq \Gamma^*$$



Die Konstruktion funktioniert, da ein Homomorphismus **strukturerhaltend** ist.

init. Zusammenhangskomponente

- Sei $A := (Z, \Sigma, \delta, z_0, Z_{\mathsf{end}})$ ein beliebiger DFA. Wir berechnen schrittweise die Mengen $M_i \subseteq Z$:

$$M_i := \begin{cases} M_{i-1} \cup \bigcup_{z \in M_{i-1}, x \in \Sigma} \delta(z, x) & \text{für } i > 0 \\ \{z_0\} & \text{für } i = 0 \end{cases}$$

init. Zusammenhangskomponente

- Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein beliebiger DFA. Wir berechnen schrittweise die Mengen $M_i \subseteq Z$:

$$M_i := \begin{cases} M_{i-1} \cup \bigcup_{z \in M_{i-1}, x \in \Sigma} \delta(z, x) & \text{für } i > 0 \\ \{z_0\} & \text{für } i = 0 \end{cases}$$

- Offensichtlich ist jeder Zustand $z \in M_i$, $i \in \mathbb{N}$ vom Startzustand aus erreichbar.

init. Zusammenhangskomponente

- Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein beliebiger DFA. Wir berechnen schrittweise die Mengen $M_i \subseteq Z$:

$$M_i := \begin{cases} M_{i-1} \cup \bigcup_{z \in M_{i-1}, x \in \Sigma} \delta(z, x) & \text{für } i > 0 \\ \{z_0\} & \text{für } i = 0 \end{cases}$$

- Offensichtlich ist jeder Zustand $z \in M_i$, $i \in \mathbb{N}$ vom Startzustand aus erreichbar.
- Wegen $M_{i-1} \subseteq M_i \subseteq Z$ und der Endlichkeit von Z existiert ein Index $k \leq |Z|$, mit $M_{k+1} = M_k$.

init. Zusammenhangskomponente

- Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein beliebiger DFA. Wir berechnen schrittweise die Mengen $M_i \subseteq Z$:

$$M_i := \begin{cases} M_{i-1} \cup \bigcup_{z \in M_{i-1}, x \in \Sigma} \delta(z, x) & \text{für } i > 0 \\ \{z_0\} & \text{für } i = 0 \end{cases}$$

- Offensichtlich ist jeder Zustand $z \in M_i$, $i \in \mathbb{N}$ vom Startzustand aus erreichbar.
- Wegen $M_{i-1} \subseteq M_i \subseteq Z$ und der Endlichkeit von Z existiert ein Index $k \leq |Z|$, mit $M_{k+1} = M_k$.
- Das Verfahren endet, wenn das erste mal $M_k = M_{k+1}$ ist, spätestens bei $k = |Z| - 1$.

init. Zusammenhangskomponente

- Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein beliebiger DFA. Wir berechnen schrittweise die Mengen $M_i \subseteq Z$:

$$M_i := \begin{cases} M_{i-1} \cup \bigcup_{z \in M_{i-1}, x \in \Sigma} \delta(z,x) & \text{für } i > 0 \\ \{z_0\} & \text{für } i = 0 \end{cases}$$

- Offensichtlich ist jeder Zustand $z \in M_i$, $i \in \mathbb{N}$ vom Startzustand aus erreichbar.
- Wegen $M_{i-1} \subseteq M_i \subseteq Z$ und der Endlichkeit von Z existiert ein Index $k \leq |Z|$, mit $M_{k+1} = M_k$.
- Das Verfahren endet, wenn das erste mal $M_k = M_{k+1}$ ist, spätestens bei $k = |Z| - 1$.
- Die Menge M_k enthält dann genau die von z_0 aus erreichbaren Zustände im DFA A.