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:
  - weitere Abschlusseigenschaften endlicher Automaten:
    - Komplement
    - Durchschnitt

Themen

- Für die heutige Vorlesung geplant:
  - weitere Abschlusseigenschaften endlicher Automaten:
    - Komplement
    - Durchschnitt
  - minimale DFAs
    - Äquivalenz von Zuständen
    - Konstruktion eines minimalen DFA

Motivation (zur Erinnerung)

- Warum studieren wir endliche Automaten?
  - Verwendung für Algorithmen, z.B. Mustersuche in Texten (vDFA)
  - in vielen Publikationen zu finden, z.B. Robert Sedgewick: Algorithmen . . . aber auch in der c't

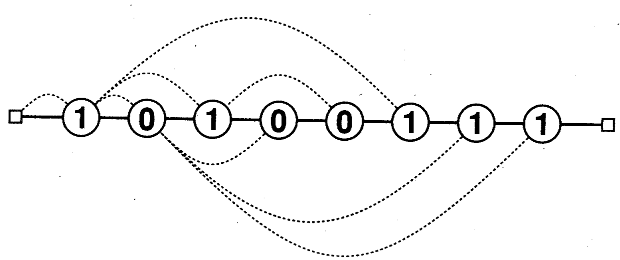


Abbildung 19.3 *Endlicher Automat für den Algorithmus von Knuth-Morris-Pratt.*

```
   i:=0;
0: i:=i+1;
1: if a[i]<>'1' then goto 0; i:=i+1;
2: if a[i]<>'0' then goto 1; i:=i+1;
3: if a[i]<>'1' then goto 1; i:=i+1;
4: if a[i]<>'0' then goto 2; i:=i+1;
5: if a[i]<>'0' then goto 3; i:=i+1;
6: if a[i]<>'1' then goto 1; i:=i+1;
7: if a[i]<>'1' then goto 2; i:=i+1;
8: if a[i]<>'1' then goto 2; i:=i+1;
   search:=i-8;
```

```
function kmpsearch: integer;
  var i,j: integer;
  begin
  i:=1; j:=1; initnext;
  repeat
    if (j=0) or (a[i]=p[j])
      then begin i:=i+1; j:=j+1 end
      else begin j:=next[j] end;
  until (j>M) or (i>N);
  if j>M then kmpsearch:=i-M else kmpsearch:=i;
  end;
```

Komplementbildung

Theorem:

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $\overline{L} := \Sigma^* \setminus L \in \mathcal{A}kz(\Sigma)$.

Komplementbildung

Theorem:

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $\overline{L} := \Sigma^* \setminus L \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein vollständiger DFA mit $L = L(A)$. Definiere einen vDFA $C_{\overline{A}}$ mit $L(C_{\overline{A}}) = \overline{L}$ durch:

Komplementbildung

Theorem:

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $\overline{L} := \Sigma^* \setminus L \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein vollständiger DFA mit $L = L(A)$. Definiere einen vDFA $C_{\overline{A}}$ mit $L(C_{\overline{A}}) = \overline{L}$ durch:

$$C_{\overline{A}} := (Z, \Sigma, \delta, z_0, Z \setminus Z_{\text{end}}).$$

Komplementbildung

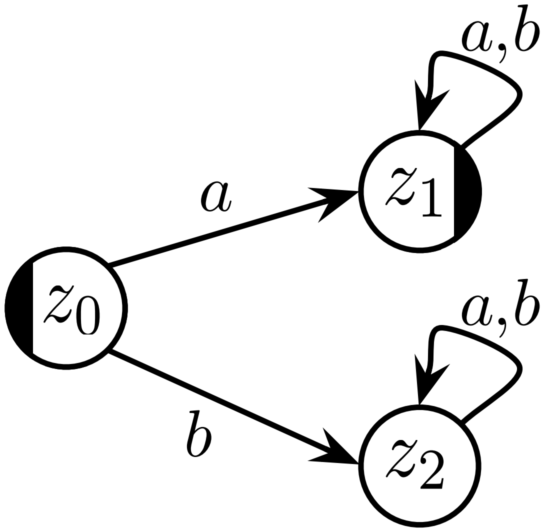
Theorem:

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $\overline{L} := \Sigma^* \setminus L \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** Sei $A := (Z, \Sigma, \delta, z_0, Z_{\textsf{end}})$ ein vollständiger DFA mit $L = L(A)$. Definiere einen vDFA $C_{\overline{A}}$ mit $L(C_{\overline{A}}) = \overline{L}$ durch:

  $$C_{\overline{A}} := (Z, \Sigma, \delta, z_0, Z \setminus Z_{\textsf{end}}).$$

  - Da A vollständig war, ist nun jede der nicht akzeptierenden Rechnungen von A eine Erfolgsrechnung in $C_{\overline{A}}$ und jede Erfolgsrechnung von A ist in $C_{\overline{A}}$ nicht mehr akzeptierend.

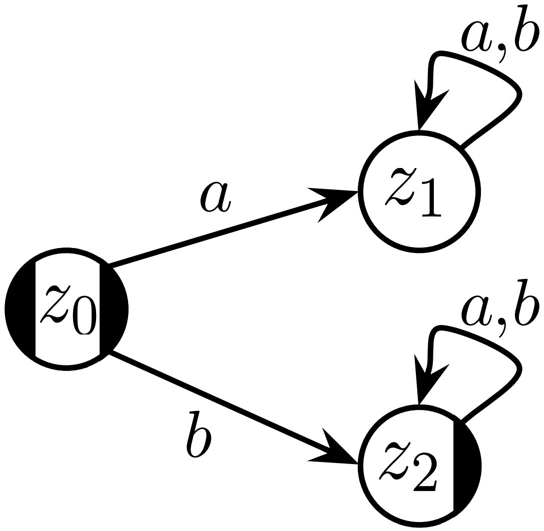
Beispiel: Komplementabschluss



...akzeptiert die Sprache

$$\{w \in \{a, b\}^* \mid w \text{ beginnt mit einem } a\}$$

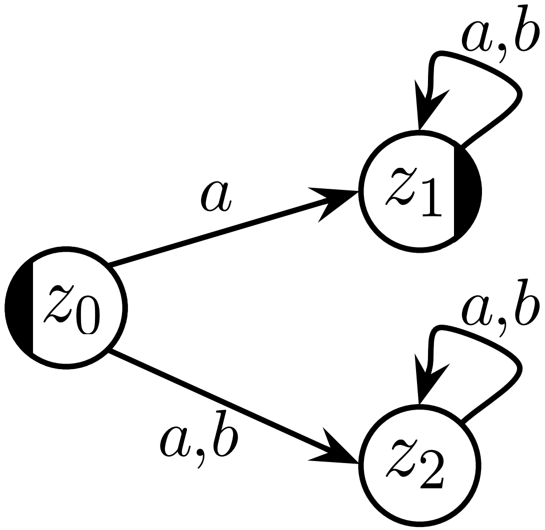
Beispiel: Komplementabschluss



. . . akzeptiert die Sprache

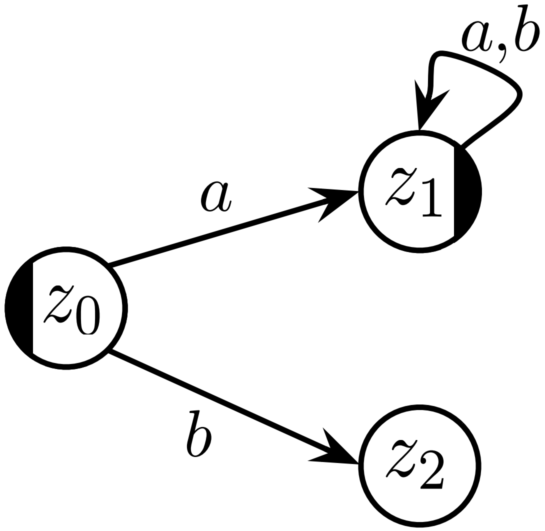
$$\begin{aligned} & \{a,b\}^* \setminus \{w \in \{a,b\}^* \mid w \text{ beginnt mit einem } a\} \\ = \;& \{\lambda, b, ba, bb, baa, bab, \ldots\} \\ = \;& \{bw \mid w \in \{a,b\}^*\} \end{aligned}$$

Beispiel: Komplementabschluss



. . . für diesen Automaten funktioniert die Konstruktion nicht! Wieso?

Beispiel: Komplementabschluss



…und für diesen Automaten funktioniert die Konstruktion auch nicht! Wieso?

Durchschnitt regulärer Mengen

- Gesetz von de Morgan: $A \cap B = \overline{\overline{A} \cup \overline{B}}$

Durchschnitt regulärer Mengen

- Gesetz von de Morgan: $A \cap B = \overline{\overline{A} \cup \overline{B}}$
- **Theorem:** Sei $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist auch $L_1 \cap L_2 \in \mathcal{A}kz(\Sigma_1 \cap \Sigma_2)$.

Durchschnitt regulärer Mengen

- Gesetz von de Morgan: $A \cap B = \overline{\overline{A} \cup \overline{B}}$
- **Theorem:** Sei $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist auch $L_1 \cap L_2 \in \mathcal{A}kz(\Sigma_1 \cap \Sigma_2)$.
- **Beweis (direkt):** (Konstruktion eines vDFA)
  Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und
  $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ vDFA's mit
  $L_1 = L(A)$ und $L_2 = L(B)$.

Durchschnitt regulärer Mengen

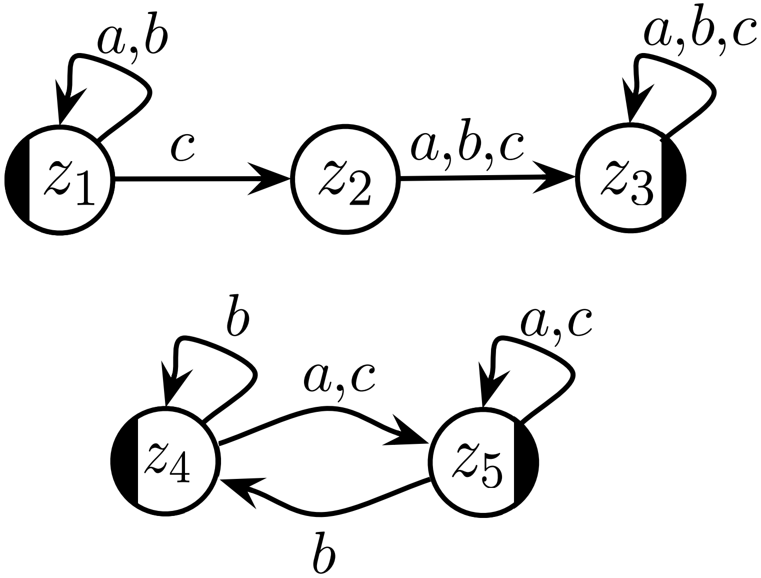
- Gesetz von de Morgan: $A \cap B = \overline{\overline{A} \cup \overline{B}}$
- **Theorem:** Sei $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist auch $L_1 \cap L_2 \in \mathcal{A}kz(\Sigma_1 \cap \Sigma_2)$.
- **Beweis (direkt):** (Konstruktion eines vDFA)
  Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ vDFA's mit $L_1 = L(A)$ und $L_2 = L(B)$.

  $C := (Z_1 \times Z_2, \Sigma_3, \delta_3, (z_{1,0}, z_{2,0}), Z_{1,\text{end}} \times Z_{2,\text{end}})$ mit $L(C) = L_1 \cap L_2$ heißt **Produktautomat**:

$$\begin{aligned} \Sigma_3 &:= \Sigma_1 \cap \Sigma_2 \\ \delta_3((z_1, z_2), x) &:= (\delta_1(z_1, x), \delta_2(z_2, x)) \\ &\ \text{für } (z_1, z_2) \in Z_1 \times Z_2 \end{aligned}$$

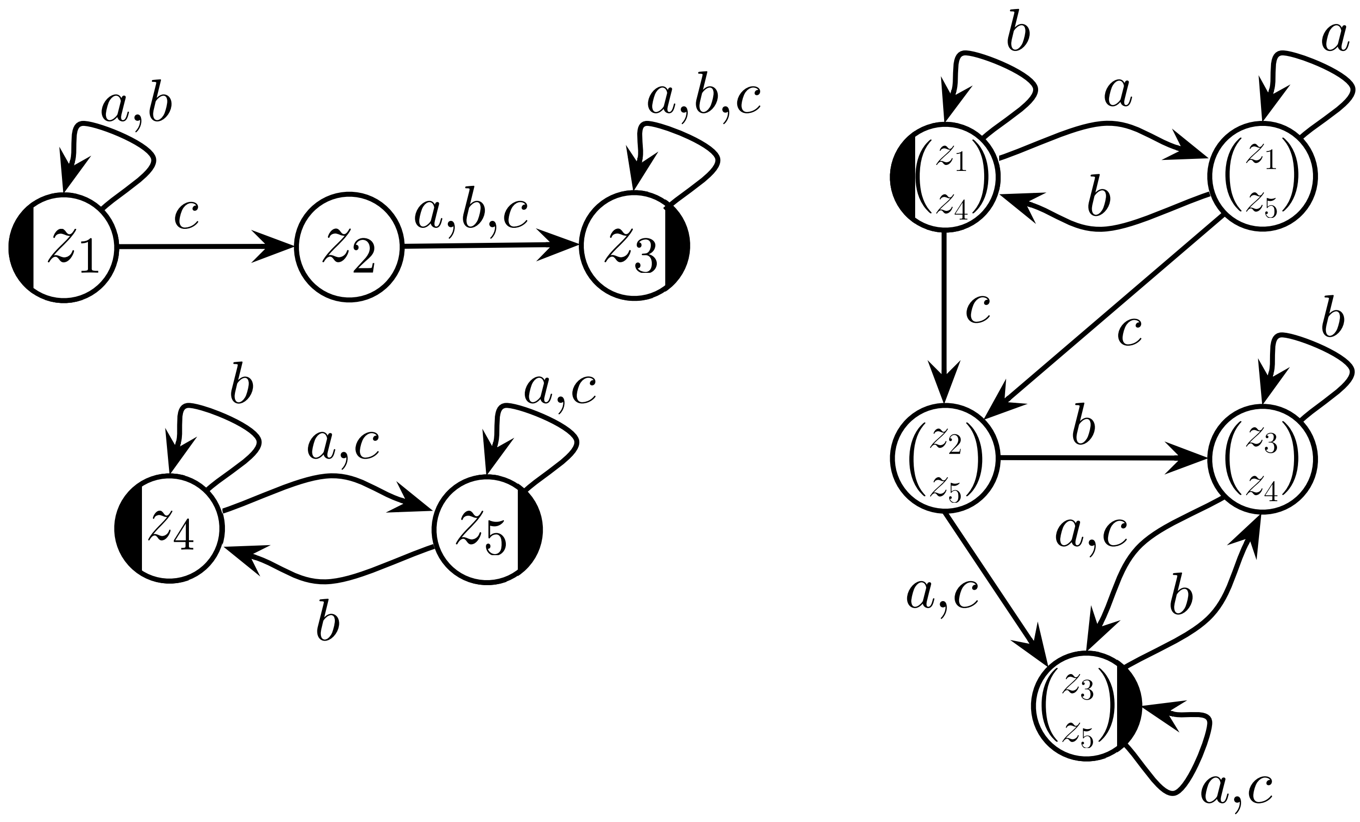
Beispiel: Produktautomat

„Synchronisieren" der beiden Automaten:



Beispiel: Produktautomat

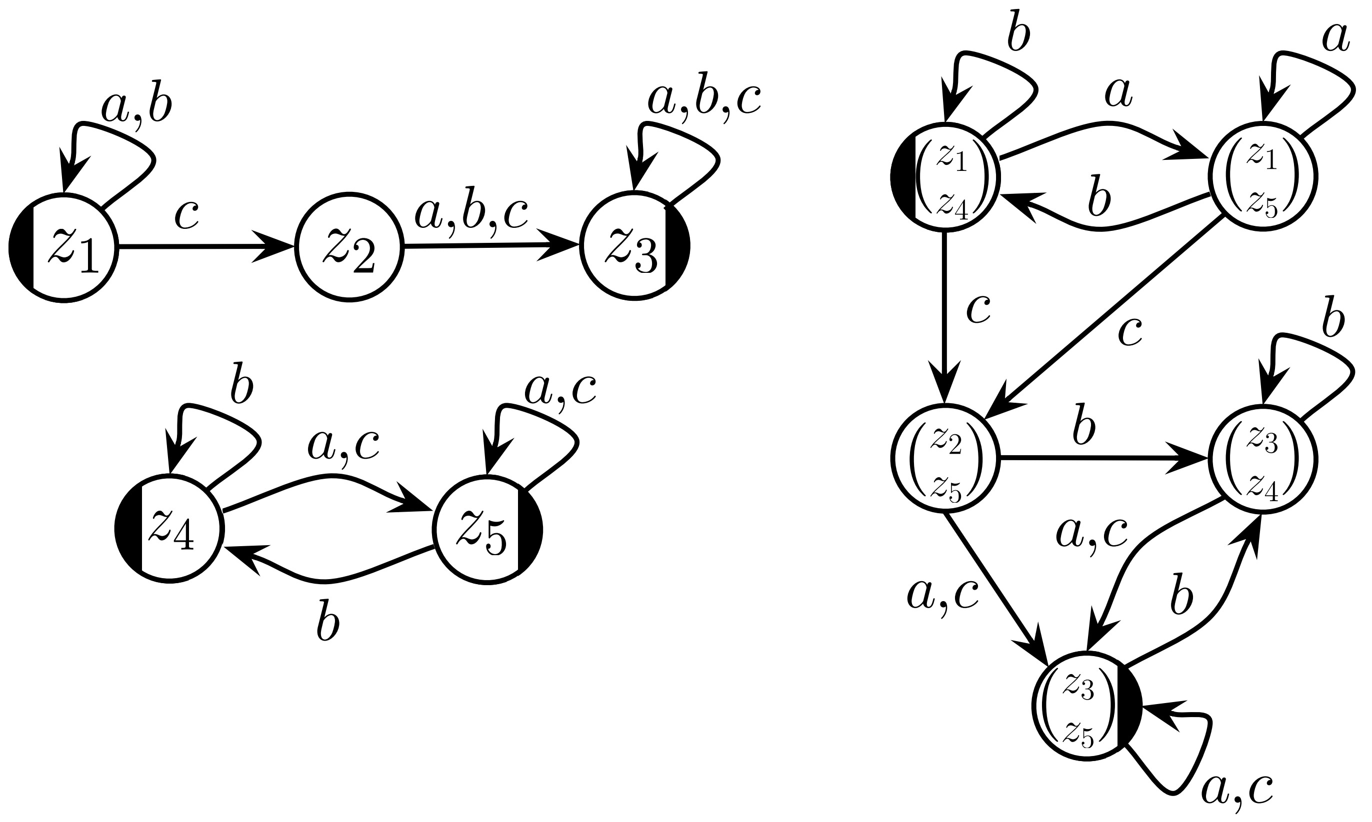
„Synchronisieren“ der beiden Automaten:



…der resultierende (vereinfachte) Produktautomat.

Beispiel: Produktautomat

„Synchronisieren“ der beiden Automaten:



...der resultierende (vereinfachte) Produktautomat.

Das Verfahren funktioniert entsprechend auch mit buchstabierenden NFAs.

Spiegelwortbildung (Reversal)

- Spiegelwortbildung:
  $w = a_1 \ldots a_n \Rightarrow w^{\mathrm{rev}} = a_n \ldots a_1$

Spiegelwortbildung (Reversal)

- Spiegelwortbildung:
  $w = a_1 \ldots a_n \Rightarrow w^{\text{rev}} = a_n \ldots a_1$
- **Theorem:** Für jede reguläre Menge $L \in \mathcal{R}eg$ gilt $L^{\text{rev}} \in \mathcal{R}eg$,

Spiegelwortbildung (Reversal)

- Spiegelwortbildung:
  $w = a_1 \dots a_n \Rightarrow w^{\mathrm{rev}} = a_n \dots a_1$
- **Theorem:** Für jede reguläre Menge $L \in \mathcal{R}eg$ gilt $L^{\mathrm{rev}} \in \mathcal{R}eg$,
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Sigma, \delta, z_0, Z_{\text{end}})$.

Spiegelwortbildung (Reversal)

- Spiegelwortbildung:
  $w = a_1 \dots a_n \Rightarrow w^{\mathrm{rev}} = a_n \dots a_1$
- **Theorem:** Für jede reguläre Menge $L \in \mathcal{R}eg$ gilt $L^{\mathrm{rev}} \in \mathcal{R}eg$,
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Sigma, \delta, z_0, Z_{\mathsf{end}})$.
  - Wir konstruieren den buchstabierenden NFA $A_{\mathrm{rev}} = (Z, \Sigma, K, Z_{\mathsf{end}}, Z_{\mathsf{start}})$ mit

Spiegelwortbildung (Reversal)

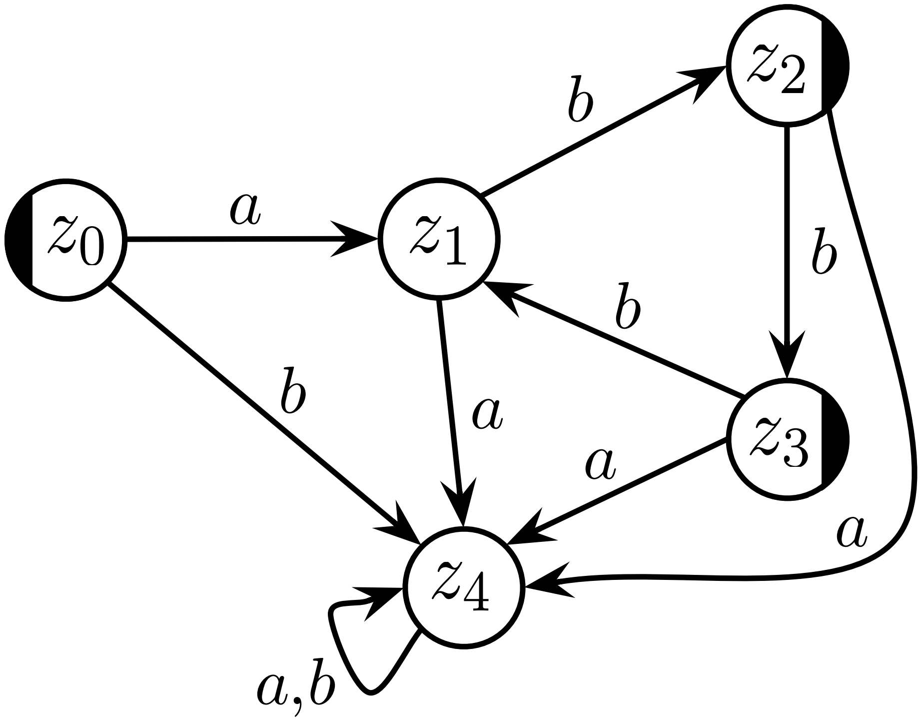
- Spiegelwortbildung:
  $w = a_1 \ldots a_n \Rightarrow w^{\text{rev}} = a_n \ldots a_1$
- **Theorem:** Für jede reguläre Menge $L \in \mathcal{R}eg$ gilt $L^{\text{rev}} \in \mathcal{R}eg$,
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Sigma, \delta, z_0, Z_{\mathsf{end}})$.
  - Wir konstruieren den buchstabierenden NFA $A_{\text{rev}} = (Z, \Sigma, K, Z_{\mathsf{end}}, Z_{\mathsf{start}})$ mit
  - $K := \{(q, x, p) \mid \delta(p, x) = q\}$.

Spiegelwortbildung (Reversal)

- Spiegelwortbildung:
  $w = a_1 \ldots a_n \Rightarrow w^{\text{rev}} = a_n \ldots a_1$
- **Theorem:** Für jede reguläre Menge $L \in \mathcal{R}eg$ gilt $L^{\text{rev}} \in \mathcal{R}eg$,
- **Beweis:** Sei $L = L(A)$ für einen vDFA $A = (Z, \Sigma, \delta, z_0, Z_{\textsf{end}})$.
  - Wir konstruieren den buchstabierenden NFA $A_{\text{rev}} = (Z, \Sigma, K, Z_{\textsf{end}}, Z_{\textsf{start}})$ mit
  - $K := \{(q, x, p) \mid \delta(p, x) = q\}$.
  - Offensichtlich entspricht jeder Erfolgspfad in A_{rev} einem rückwärts von einem End- zu einem Startzustand in A gegangenen Erfolgspfad.

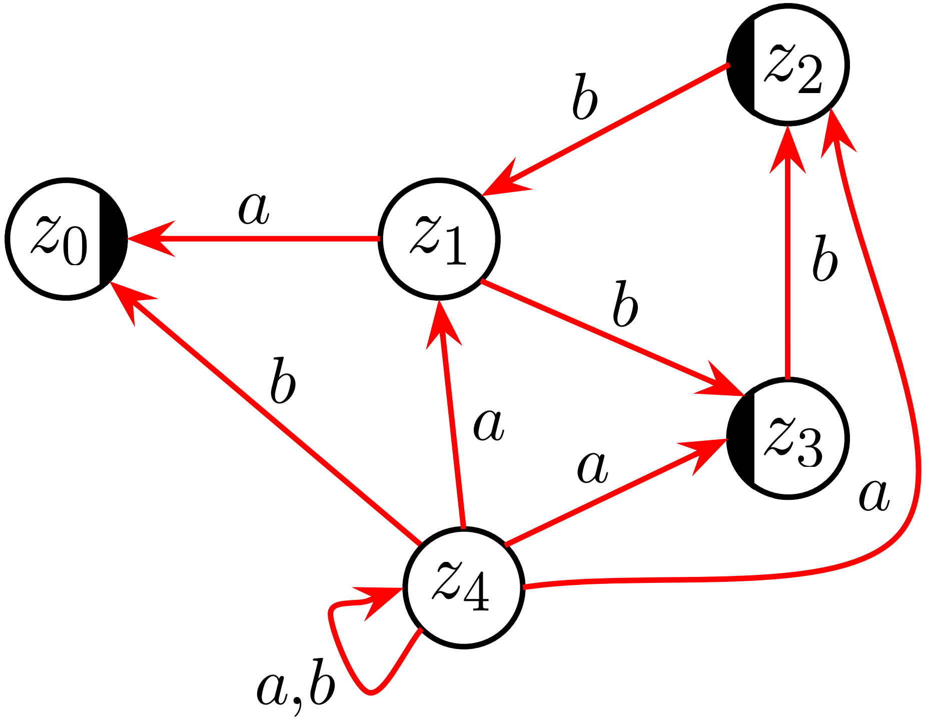
Beispiel: Reversal

„Umdrehen“ der Kanten eines vDFA ...



Beispiel: Reversal

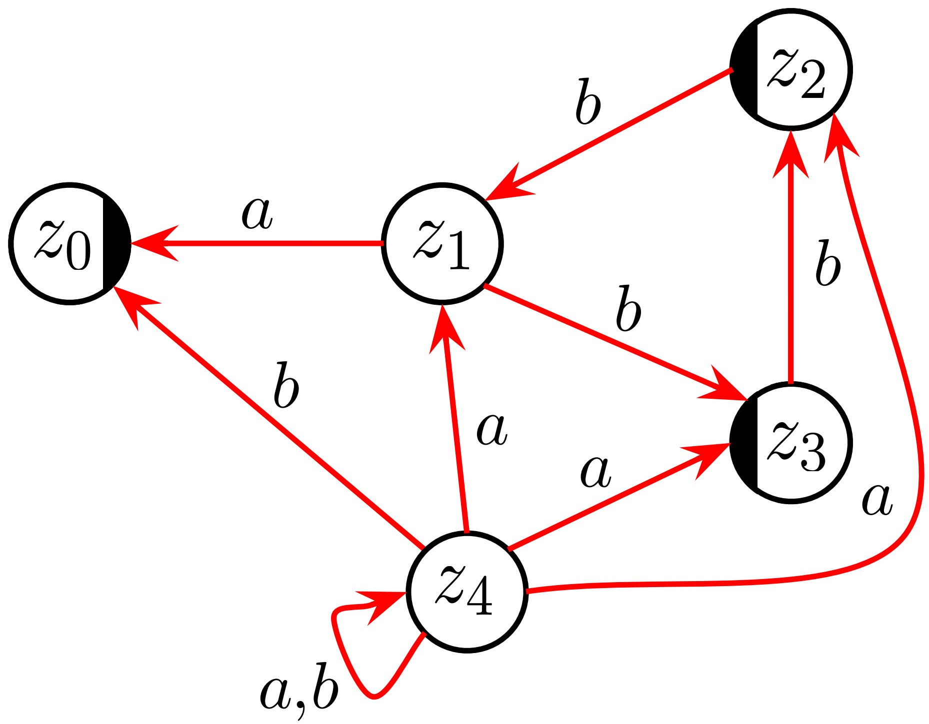
„Umdrehen" der Kanten eines vDFA ...



... führt zum Spiegelwortautomaten.

Beispiel: Reversal

„Umdrehen“ der Kanten eines vDFA . . .



. . . führt zum Spiegelwortautomaten.

Auch hier kann das Verfahren leicht für buchstabierende NFAs angepasst werden.

Definition: äquivalente Zustände

- Zu jedem endlichen Automaten (NFA) $A = (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ heiße die Menge

  $$L(z) := \{w \in \Sigma^* \mid z \xrightarrow[w]{*} z', z' \in Z_{\mathsf{end}}\}$$

  die **Leistung** des Zustandes z.

Definition: äquivalente Zustände

- Zu jedem endlichen Automaten (NFA) $A = (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ heiße die Menge

  $$L(z) := \{w \in \Sigma^* \mid z \xrightarrow[w]{*} z', z' \in Z_{\mathsf{end}}\}$$

  die **Leistung** des Zustandes z.
- Zwei Zustände von A heißen genau dann **äquivalent**, wenn $L(z) = L(z')$ ist. Wir notieren dies in der Form $z \equiv z'$. Nicht äquivalente Zustände, $z \not\equiv z'$, nennen wir **unterscheidbar**.

Definition: äquivalente Zustände

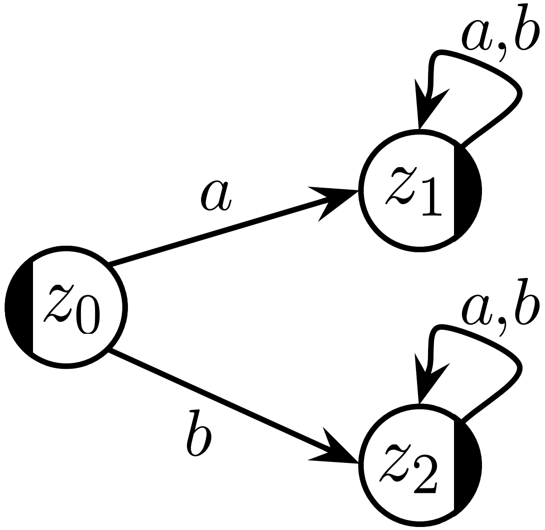
- Zu jedem endlichen Automaten (NFA) $A = (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ heiße die Menge

  $$L(z) := \{w \in \Sigma^* \mid z \xrightarrow[w]{*} z', z' \in Z_{\mathsf{end}}\}$$

  die **Leistung** des Zustandes z.
- Zwei Zustände von A heißen genau dann **äquivalent**, wenn $L(z) = L(z')$ ist. Wir notieren dies in der Form $z \equiv z'$. Nicht äquivalente Zustände, $z \not\equiv z'$, nennen wir **unterscheidbar**.
- A heißt genau dann **reduziert**, wenn keine zwei verschiedenen Zustände äquivalent sind.

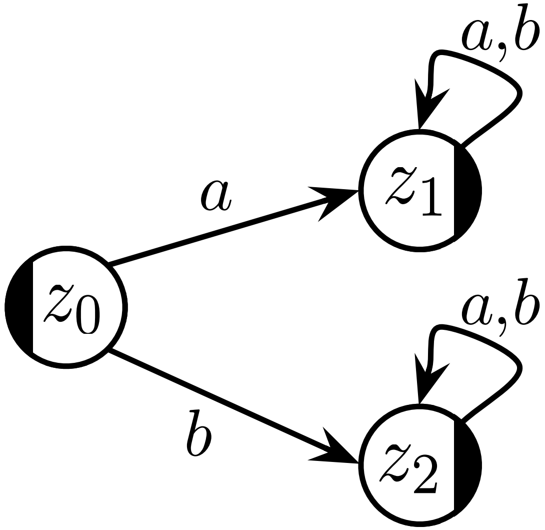
Beispiel: äquivalente FAs

Betrachten wir folgenden Automaten:



Beispiel: äquivalente FAs

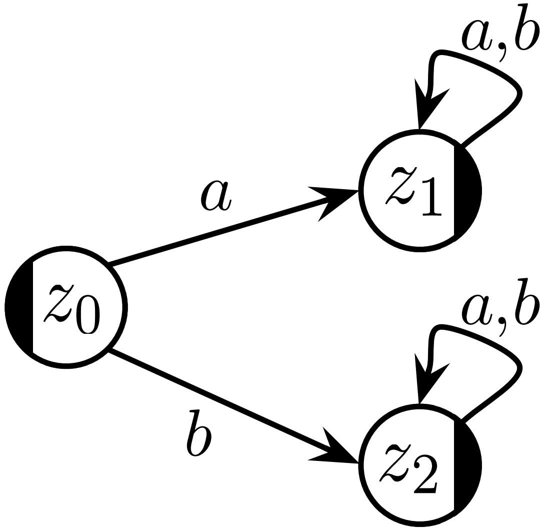
Betrachten wir folgenden Automaten:



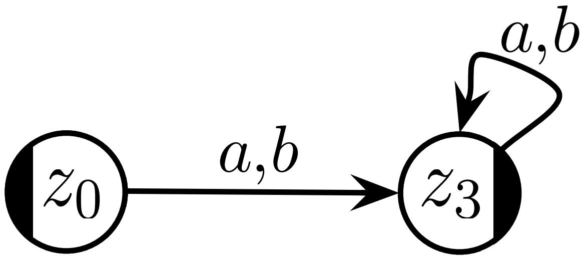
Die Zustände z_1 und z_2 sind äquivalent.

Beispiel: äquivalente FAs

Betrachten wir folgenden Automaten:



Die Zustände z_1 und z_2 sind äquivalent.



ist ein äquivalenter DFA mit weniger Zuständen!

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:
  1. vollständig

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:
  1. vollständig
  2. initial zusammenhängend

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:
  1. vollständig
  2. initial zusammenhängend
  3. deterministisch

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:
  1. vollständig
  2. initial zusammenhängend
  3. deterministisch
  4. reduziert

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:
  1. vollständig
  2. initial zusammenhängend
  3. deterministisch
  4. reduziert
- Warum heißt ein solcher Automat „minimal“?

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:
  1. vollständig
  2. initial zusammenhängend
  3. deterministisch
  4. reduziert
- Warum heißt ein solcher Automat „minimal“?
- **Theorem:** Ein vollständiger DFA $A = (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ist genau dann minimal, wenn kein äquivalenter, vollständiger DFA weniger Zustände besitzt.

Definition: minimaler DFA

- Ein endlicher Automat heißt genau dann **minimal**, wenn er folgende Eigenschaften hat:
  1. vollständig
  2. initial zusammenhängend
  3. deterministisch
  4. reduziert
- Warum heißt ein solcher Automat „minimal“?
- **Theorem:** Ein vollständiger DFA $A = (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ist genau dann minimal, wenn kein äquivalenter, vollständiger DFA weniger Zustände besitzt.

$\boxed{\text{Das Potenzautomatenverfahren liefert in der Regel keinen minimalen DFA!}}$

Äquivalenzautomat

- **Theorem:** Je zwei äquivalente, minimale DFA sind isomorph, d.h., sie besitzen bis auf die Bezeichnungen der Zustände das gleiche Zustandsdiagramm.

Äquivalenzautomat

- **Theorem:** Je zwei äquivalente, minimale DFA sind isomorph, d.h., sie besitzen bis auf die Bezeichnungen der Zustände das gleiche Zustandsdiagramm.
- **Definition:** Sei $A = (Z, \Sigma, \delta, z_0, Z_{\mathsf{end}})$ ein DFA, und für jeden Zustand $z \in Z$ seine Äquivalenzklasse $[z] := \{z' \mid z' \equiv z\}$ die Menge aller Zustände mit gleicher Leistung.

Äquivalenzautomat

- **Theorem:** Je zwei äquivalente, minimale DFA sind isomorph, d.h., sie besitzen bis auf die Bezeichnungen der Zustände das gleiche Zustandsdiagramm.
- **Definition:** Sei $A = (Z, \Sigma, \delta, z_0, Z_{\mathsf{end}})$ ein DFA, und für jeden Zustand $z \in Z$ seine Äquivalenzklasse $[z] := \{z' \mid z' \equiv z\}$ die Menge aller Zustände mit gleicher Leistung.
- Der **Äquivalenzautomat** zu A ist :
  $A' := (Z', X, \delta', [z_0], Z'_{\mathsf{end}})$, mit
  $Z' := \{[z] \mid z \in Z\}$, $Z'_{\mathsf{end}} := \{[z] \mid z \in Z_{\mathsf{end}}\}$ und
  $\delta'([z_1]_A, x) = [z_2]_A$ gdw. $\delta(p, x) = q$ für irgendwelche Zustände $p \in [z_1]$ und $q \in [z_2]$ gilt.

…der Weg zum minimalen DFA

- **Theorem:** Der Äquivalenzautomat A' zu einem initial zusammenhängenden vollständigen DFA A ist minimal und akzeptiert die gleiche Sprache.

. . . der Weg zum minimalen DFA

- **Theorem:** Der Äquivalenzautomat A' zu einem initial zusammenhängenden vollständigen DFA A ist minimal und akzeptiert die gleiche Sprache.
- Jetzt wissen wir wann ein Automat minimal ist und wie ein minimaler Automat zu einem gegebenen DFA aussieht.

…der Weg zum minimalen DFA

- **Theorem:** Der Äquivalenzautomat A' zu einem initial zusammenhängenden vollständigen DFA A ist minimal und akzeptiert die gleiche Sprache.
- Jetzt wissen wir wann ein Automat minimal ist und wie ein minimaler Automat zu einem gegebenen DFA aussieht.
- Aber wir wissen nicht, wie wir einen solchen Automaten konstruieren können!

…der Weg zum minimalen DFA

- **Theorem:** Der Äquivalenzautomat A' zu einem initial zusammenhängenden vollständigen DFA A ist minimal und akzeptiert die gleiche Sprache.
- Jetzt wissen wir wann ein Automat minimal ist und wie ein minimaler Automat zu einem gegebenen DFA aussieht.
- Aber wir wissen nicht, wie wir einen solchen Automaten konstruieren können!
- Dazu ist nötig:

. . . der Weg zum minimalen DFA

- **Theorem:** Der Äquivalenzautomat A' zu einem initial zusammenhängenden vollständigen DFA A ist minimal und akzeptiert die gleiche Sprache.
- Jetzt wissen wir wann ein Automat minimal ist und wie ein minimaler Automat zu einem gegebenen DFA aussieht.
- Aber wir wissen nicht, wie wir einen solchen Automaten konstruieren können!
- Dazu ist nötig:
  - äquivalente Zustände zu bestimmen,

. . . der Weg zum minimalen DFA

- **Theorem:** Der Äquivalenzautomat A' zu einem initial zusammenhängenden vollständigen DFA A ist minimal und akzeptiert die gleiche Sprache.
- Jetzt wissen wir wann ein Automat minimal ist und wie ein minimaler Automat zu einem gegebenen DFA aussieht.
- Aber wir wissen nicht, wie wir einen solchen Automaten konstruieren können!
- Dazu ist nötig:
  - äquivalente Zustände zu bestimmen,
  - um die Zustände (Äquivalenzklassen) und Zustandsübergänge des Äquivalenzautomaten zu erlangen.

Konstruktion des minimalen DFA

- **Eingabe:** Ein DFA $A = (Z, \Sigma, \delta, z_0, Z_{end})$.

Konstruktion des minimalen DFA

- **Eingabe:** Ein DFA $A = (Z, \Sigma, \delta, z_0, Z_{end})$.
- **Initialisierung:** Notiere alle zweielementigen Teilmengen von Z.

Konstruktion des minimalen DFA

- **Eingabe:** Ein DFA $A = (Z, \Sigma, \delta, z_0, Z_{end})$.
- **Initialisierung:** Notiere alle zweielementigen Teilmengen von Z.

  1. **Endzustand und Nicht-Endzustand nicht äquivalent!**
     Färbe alle Mengen $\{p, q\}$, wo $p \in Z_{end}$ und $q \in Z \setminus Z_{end}$ ***schwarz***.

Konstruktion des minimalen DFA

- **Eingabe:** Ein DFA $A = (Z, \Sigma, \delta, z_0, Z_{end})$.
- **Initialisierung:** Notiere alle zweielementigen Teilmengen von Z.

  1. **Endzustand und Nicht-Endzustand nicht äquivalent!**
     Färbe alle Mengen $\{p, q\}$, wo $p \in Z_{end}$ und $q \in Z \setminus Z_{end}$ ***schwarz***.

...und wie sieht es mit anderen Zustandspaaren aus?

Konstruktion des minimalen DFA

- **Eingabe:** Ein DFA $A = (Z, \Sigma, \delta, z_0, Z_{end})$.
- **Initialisierung:** Notiere alle zweielementigen Teilmengen von Z.
  1. **Endzustand und Nicht-Endzustand nicht äquivalent!**
     Färbe alle Mengen $\{p, q\}$, wo $p \in Z_{end}$ und $q \in Z \setminus Z_{end}$ ***schwarz***.

...und wie sieht es mit anderen Zustandspaaren aus?
Kurzscheibweise: $(p)^a = \delta(p, a)$

Konstruktion des minimalen DFA

2. **Zwei Zustände sind nicht äquivalent, falls aus ihnen für dieselbe Eingabe zwei nicht äquivalente Zustände erreicht werden!**

Konstruktion des minimalen DFA

2. **Zwei Zustände sind nicht äquivalent, falls aus ihnen für dieselbe Eingabe zwei nicht äquivalente Zustände erreicht werden!**
   1. Wähle beliebige ungefärbte Zustandsmenge $\{p, q\}$ und färbe sie *weiss*.

Konstruktion des minimalen DFA

2. **Zwei Zustände sind nicht äquivalent, falls aus ihnen für dieselbe Eingabe zwei nicht äquivalente Zustände erreicht werden!**
   1. Wähle beliebige ungefärbte Zustandsmenge $\{p, q\}$ und färbe sie *weiss*.
      (a) Falls für ein $a \in \Sigma$ die Menge $\{(p)^a, (q)^a\}$ *schwarz* gefärbt ist, so färbe $\{p, q\}$ sowie alle von dort aus erreichbaren Mengen *schwarz*.

Konstruktion des minimalen DFA

2. **Zwei Zustände sind nicht äquivalent, falls aus ihnen für dieselbe Eingabe zwei nicht äquivalente Zustände erreicht werden!**
   1. Wähle beliebige ungefärbte Zustandsmenge $\{p, q\}$ und färbe sie *weiss*.
      (a) Falls für ein $a \in \Sigma$ die Menge $\{(p)^a, (q)^a\}$ *schwarz* gefärbt ist, so färbe $\{p, q\}$ sowie alle von dort aus erreichbaren Mengen *schwarz*.
      (b) Ansonsten: Zeichne für alle $a \in \Sigma$ eine Kante von $\{(p)^a, (q)^a\}$ nach $\{p, q\}$, sofern $(p)^a \neq (q)^a$ und $\{p, q\} \neq \{(p)^a, (q)^a\}$.

Konstruktion des minimalen DFA

2. **Zwei Zustände sind nicht äquivalent, falls aus ihnen für dieselbe Eingabe zwei nicht äquivalente Zustände erreicht werden!**
   1. Wähle beliebige ungefärbte Zustandsmenge $\{p, q\}$ und färbe sie *weiss*.
      (a) Falls für ein $a \in \Sigma$ die Menge $\{(p)^a, (q)^a\}$ *schwarz* gefärbt ist, so färbe $\{p, q\}$ sowie alle von dort aus erreichbaren Mengen *schwarz*.
      (b) Ansonsten: Zeichne für alle $a \in \Sigma$ eine Kante von $\{(p)^a, (q)^a\}$ nach $\{p, q\}$, sofern $(p)^a \neq (q)^a$ und $\{p, q\} \neq \{(p)^a, (q)^a\}$.
   2. Sind alle zweielementigen Zustandsmengen gefärbt, so terminiere.

Konstruktion des minimalen DFA

Ausgabe:

- $weiss$ gefärbte Zustandspaare sind äquivalent,

Konstruktion des minimalen DFA

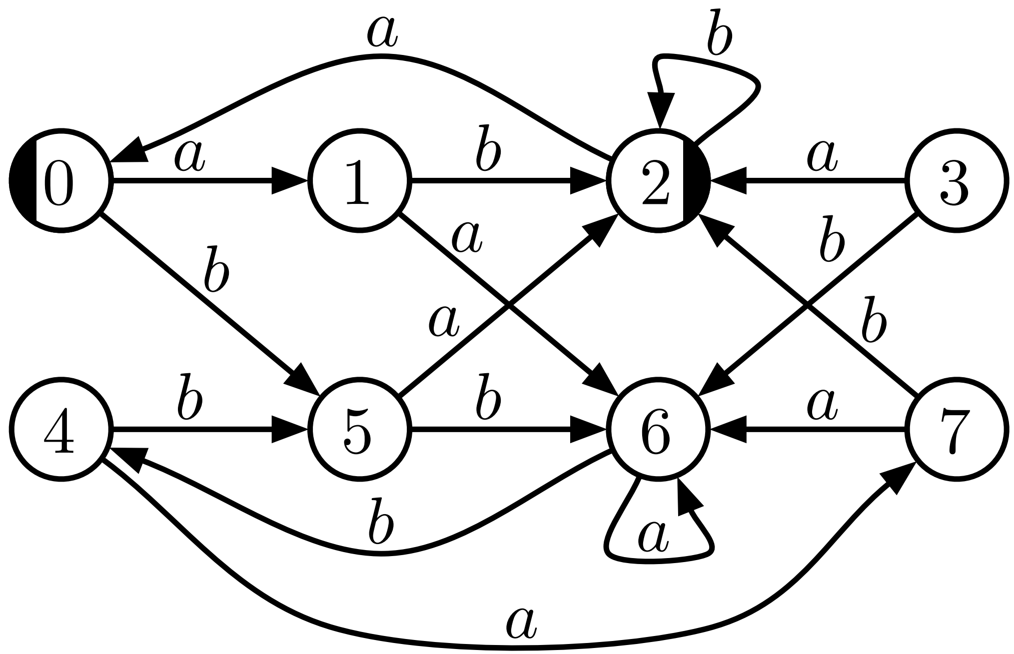
Ausgabe:

- *weiss* gefärbte Zustandspaare sind äquivalent,
- *schwarz* gefärbte Zustandspaare sind nicht äquivalent.

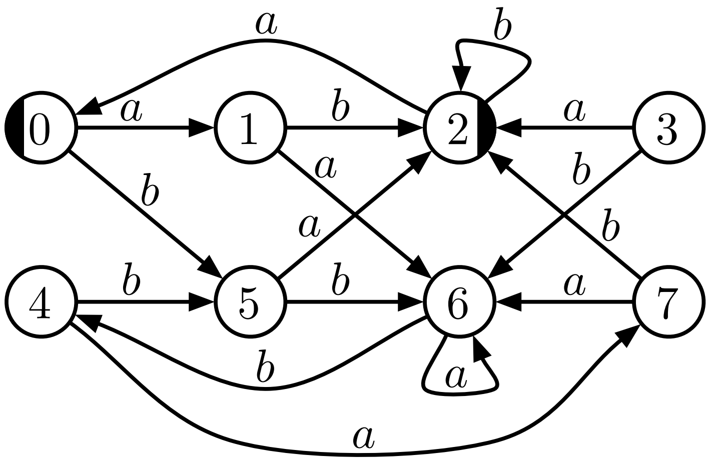
Konstruktion des minimalen DFA

Ausgabe:

- *weiss* gefärbte Zustandspaare sind äquivalent,
- *schwarz* gefärbte Zustandspaare sind nicht äquivalent.
- Ist dieser Automat minimal?

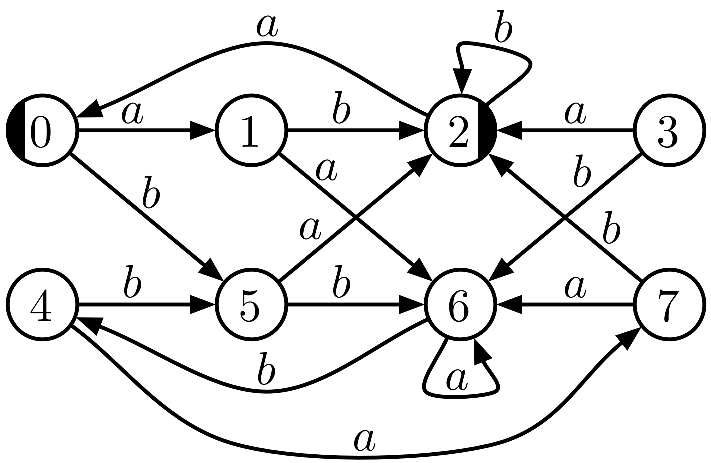


Beispiel: äquivalente Zustände



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

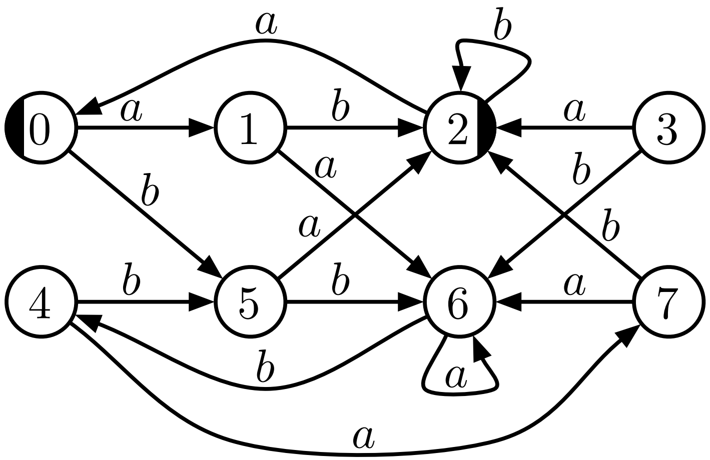
Beispiel: äquivalente Zustände



0,1	$\boxed{0,2}$	0,3	0,4	0,5	0,6	0,7
	$\boxed{1,2}$	1,3	1,4	1,5	1,6	1,7
		$\boxed{2,3}$	$\boxed{2,4}$	$\boxed{2,5}$	$\boxed{2,6}$	$\boxed{2,7}$
			3,4	3,5	3,6	3,7
				4,5	4,6	4,7
					5,6	5,7
						6,7

Situation nach der Initialisierung

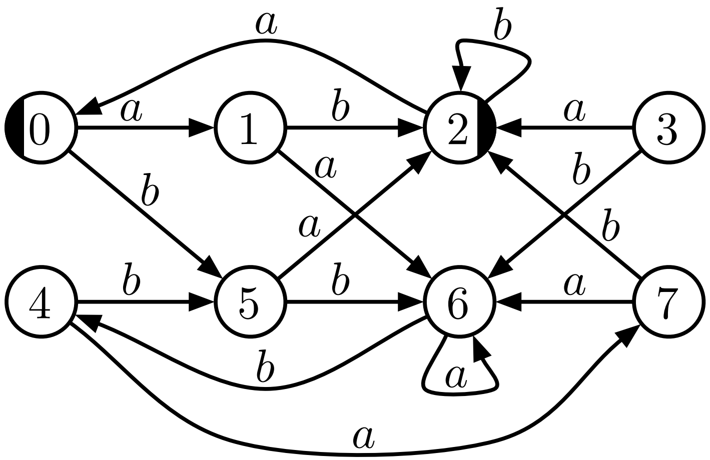
Beispiel: äquivalente Zustände

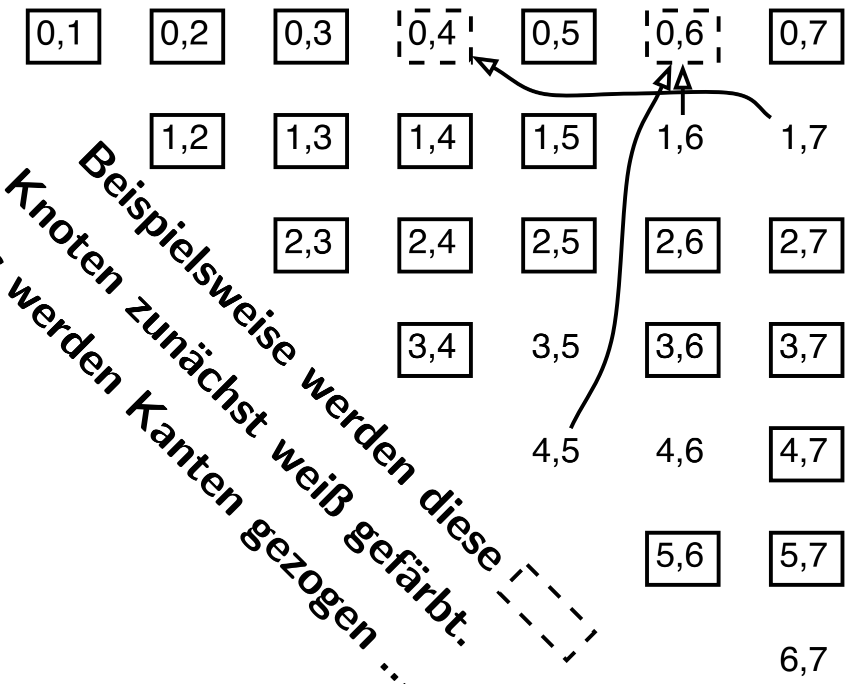


0,1	$\boxed{0,2}$	0,3	0,4	0,5	0,6	0,7
	$\boxed{1,2}$	1,3	1,4	1,5	1,6	1,7
		$\boxed{2,3}$	$\boxed{2,4}$	$\boxed{2,5}$	$\boxed{2,6}$	$\boxed{2,7}$
			3,4	3,5	3,6	3,7
				4,5	4,6	4,7
					5,6	5,7
						6,7

u.a. werden diese Mengen direkt schwarz gefärbt.

Beispiel: äquivalente Zustände



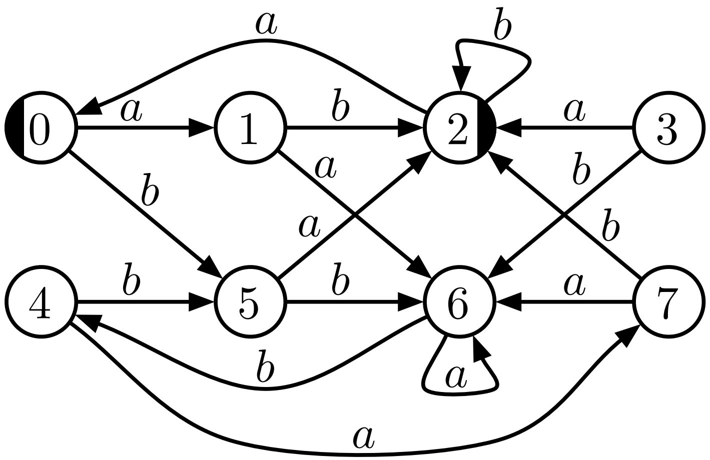


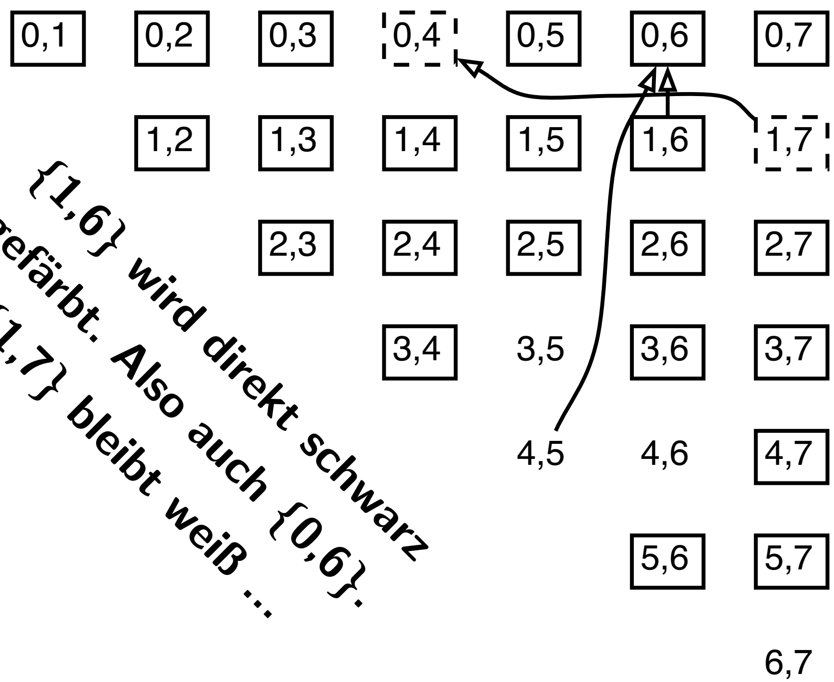
Beispielsweise werden diese

Knoten zunächst weiß gefärbt.

Es werden Kanten gezogen ...

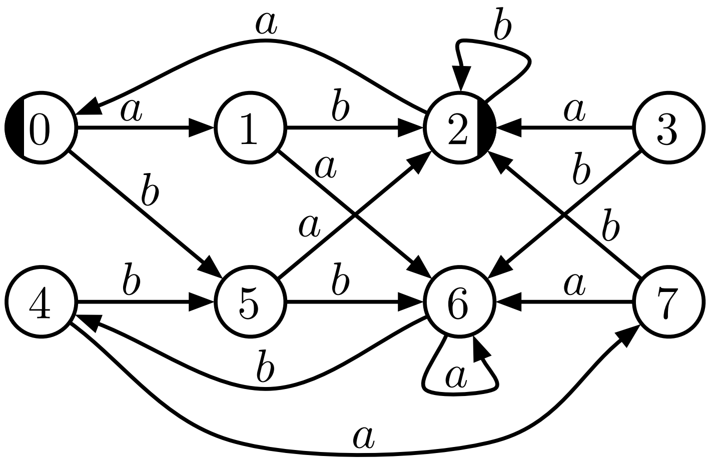
Beispiel: äquivalente Zustände





{1,6} wird direkt schwarz gefärbt. Also auch {0,6}. {1,7} bleibt weiß ...

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

Geschafft!!!

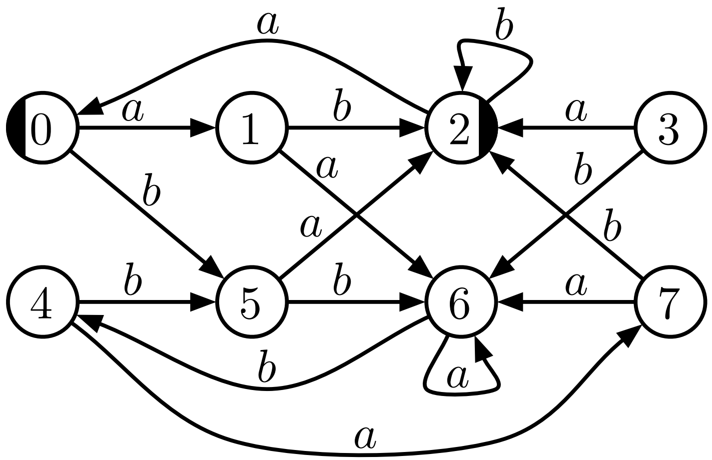
Ergebnis:

0 ist äquivalent zu 4.

1 ist äquivalent zu 7.

3 ist äquivalent zu 5.

Beispiel: äquivalente Zustände

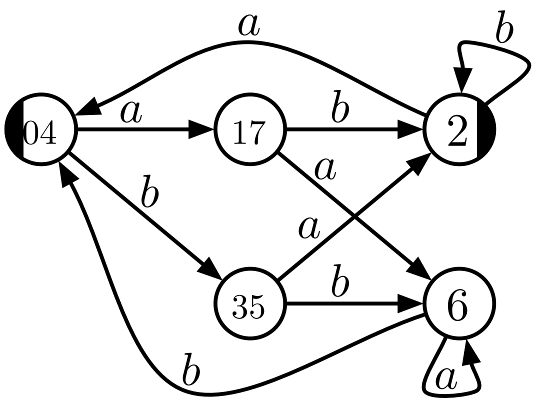


Ergebnis:

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*** $\quad\Rightarrow\quad p \not\equiv q$

Beweis

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

2.) **Korrektheit:**

A: $\{p, q\}$ ist $\boldsymbol{\mathit{schwarz}} \quad \Rightarrow \quad p \not\equiv q$

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

Beweis

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

2.) **Korrektheit:**

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

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

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

Ausblick

- Nächste Woche:
  - Beweis zum Färbealgorithmus
  - Homomorphismen
  - Substitutionen
  - Grenzen der regulären Sprachen
  - einfache Verfahren