LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

**Formale Sprachen und Komplexität**
**Theoretische Informatik für Studierende der Medieninformatik**
Sommersemester 2024

# 2c
# Minimierung von deterministischen endlichen Automaten

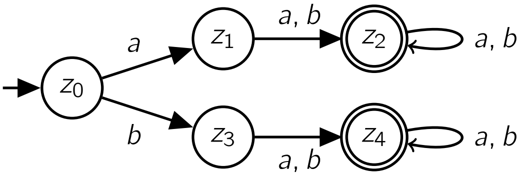

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen


Stand: 10. Mai 2024
Basiert auf Folien von PD Dr. David Sabel

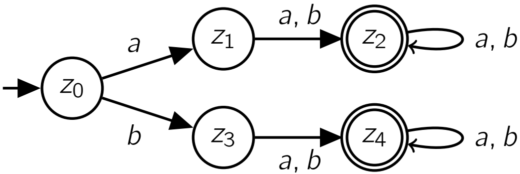
# Beispiel für Minimalität

Nicht minimaler DFA:

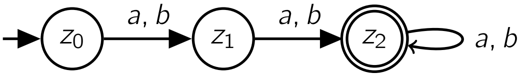


# Beispiel für Minimalität

Nicht minimaler DFA:

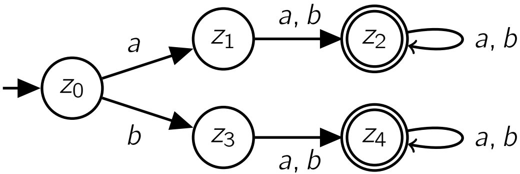



Minimaler DFA:

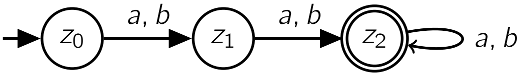


# Beispiel für Minimalität

Nicht minimaler DFA:




Minimaler DFA:




Beide DFAs akzeptieren die gleiche Sprache,
aber der zweite hat weniger Zustände.

# Vorteile der Minimierung von DFAs

- Minimale DFAs sind effizienter zu implementieren.
- Minimale DFAs sind oft einfacher zu verstehen oder zu analysieren.
- Da minimale DFAs (bis auf Umbenennung der Zustände) eindeutig sind, können sie Sprachen exakt darstellen.

# Grundgedanke der Minimierung von DFAs

Schritte:

1. Entferne alle nicht erreichbaren Zustände.
2. Bilde eine Partition (d.h. disjunkte Zerlegung), indem jeweils alle Endzustände und alle Nicht-Endzustände verschmolzen werden.
3. Verfeinere diese Partition schrittweise, bis sie sich nicht mehr verändert.

# Grundgedanke der Minimierung von DFAs

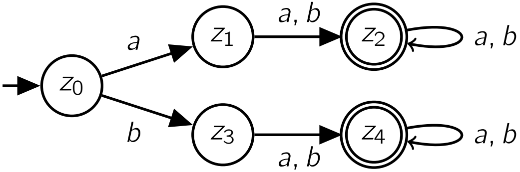
Schritte:

1. Entferne alle nicht erreichbaren Zustände.
2. Bilde eine Partition (d.h. disjunkte Zerlegung), indem jeweils alle Endzustände und alle Nicht-Endzustände verschmolzen werden.
3. Verfeinere diese Partition schrittweise, bis sie sich nicht mehr verändert.

Zwei Ansätze: graphisch und tabellarisch.

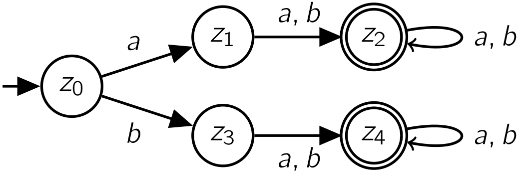
# Beispiel für den graphischen Ansatz

DFA:

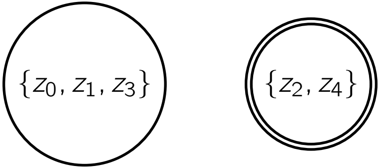


# Beispiel für den graphischen Ansatz

DFA:

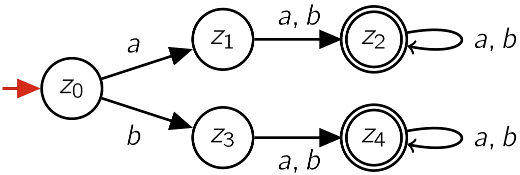



0.

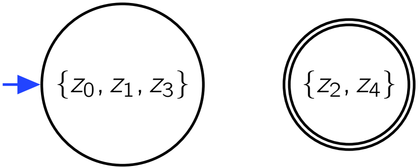


# Beispiel für den graphischen Ansatz

DFA:

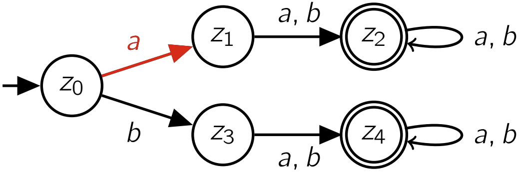



0.

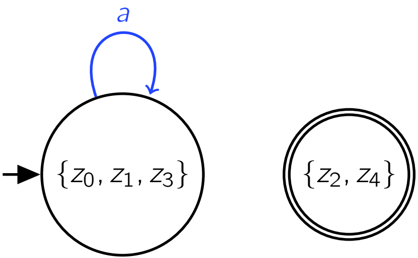


# Beispiel für den graphischen Ansatz

DFA:

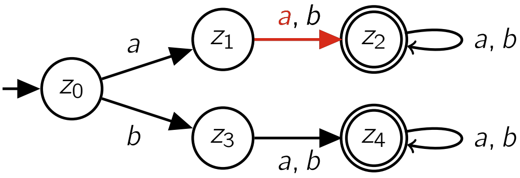



0.

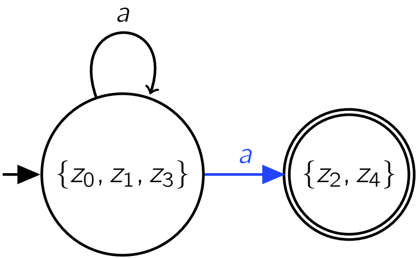


# Beispiel für den graphischen Ansatz

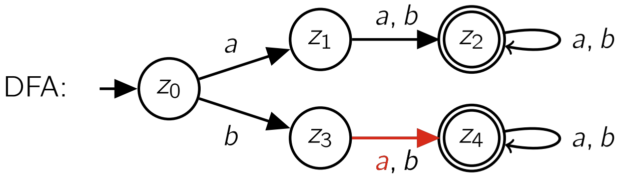
DFA:



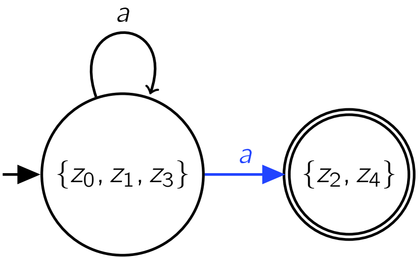

0.



# Beispiel für den graphischen Ansatz

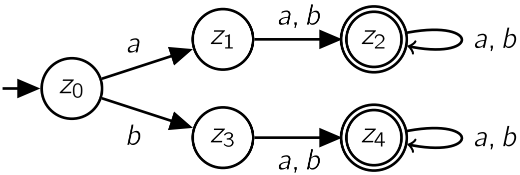



0.

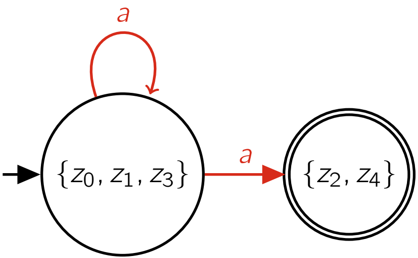


# Beispiel für den graphischen Ansatz

DFA:



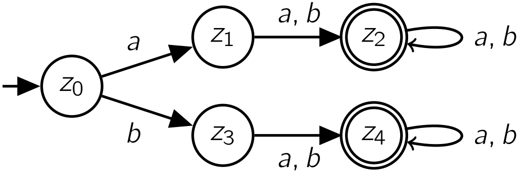

0.

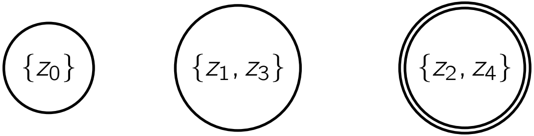



Wird kein DFA

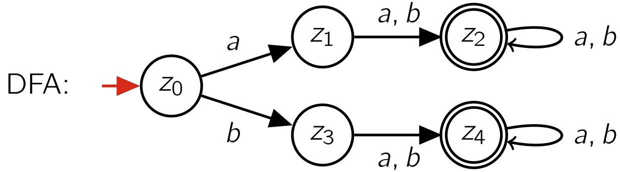
$\{z_0, z_1, z_3\}$ muss gespalten werden, damit ein DFA entsteht.

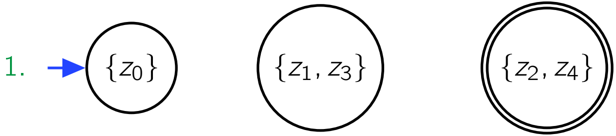
# Beispiel für den graphischen Ansatz

DFA: 


1. 

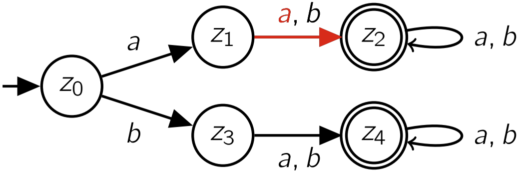
# Beispiel für den graphischen Ansatz



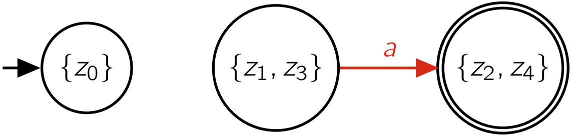



# Beispiel für den graphischen Ansatz

DFA:

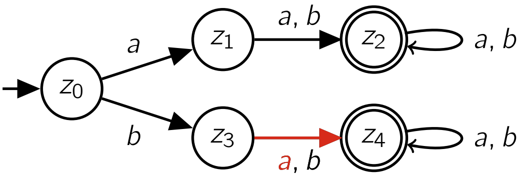



1.

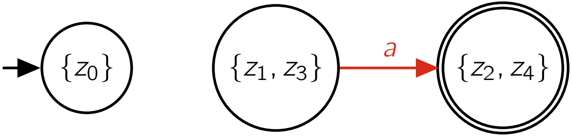


# Beispiel für den graphischen Ansatz

DFA:

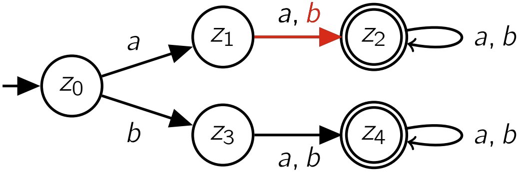



1.

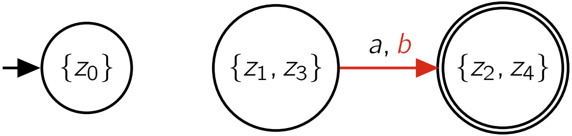


# Beispiel für den graphischen Ansatz

DFA:

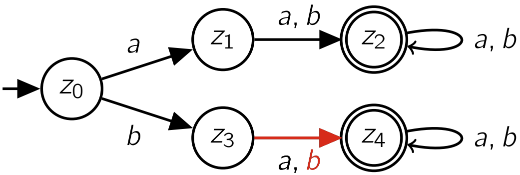



1.

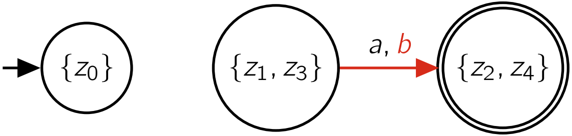


# Beispiel für den graphischen Ansatz

DFA:

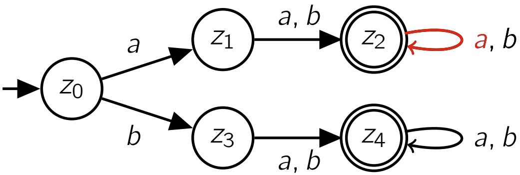



1.

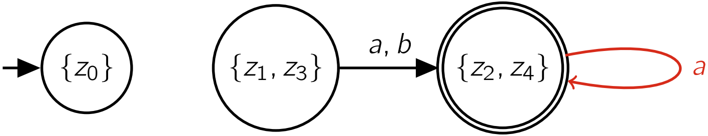


# Beispiel für den graphischen Ansatz

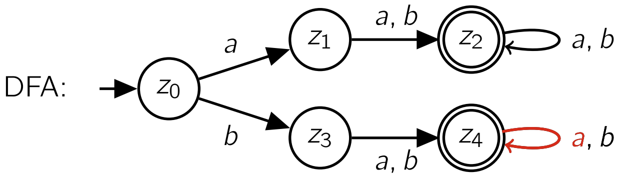
DFA:



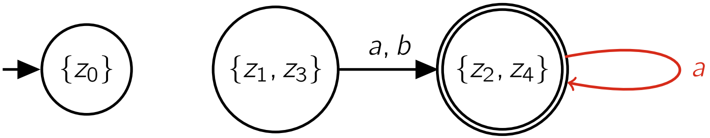

1.



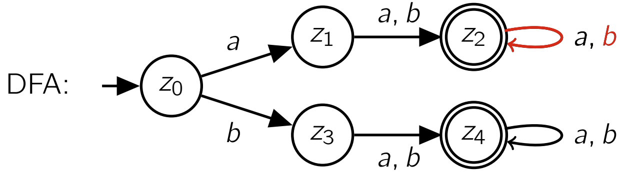
# Beispiel für den graphischen Ansatz

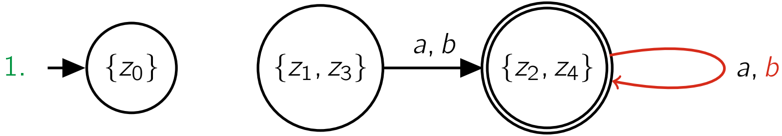



1.



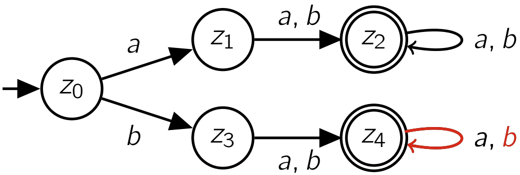
# Beispiel für den graphischen Ansatz



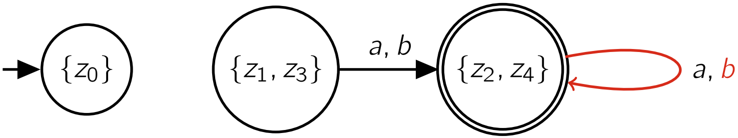



# Beispiel für den graphischen Ansatz

DFA:

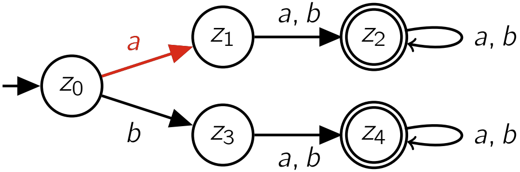



1.

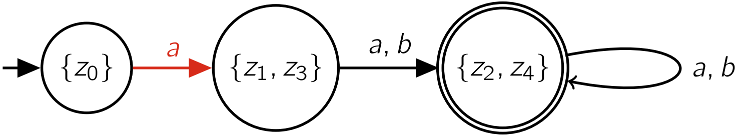


# Beispiel für den graphischen Ansatz

DFA:

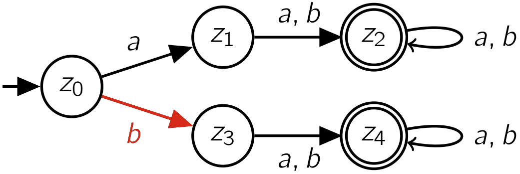



1.

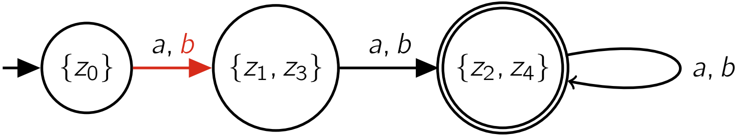


# Beispiel für den graphischen Ansatz

DFA:

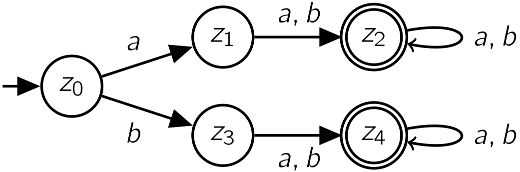



1.

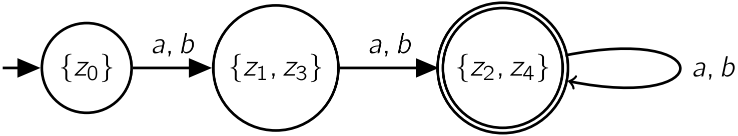


# Beispiel für den graphischen Ansatz

DFA:




1.




Minimaler DFA

# Minimierung eines DFAs mit dem tabellarischen Ansatz

Sei $M$ ein DFA.

Intuitiver Ansatz:

1. Entferne alle nicht erreichbaren Zustände von $M$.
2. Konstruiere die Partitionstabelle.
3. Bilde den minimalen DFA $M'$, indem Zustände derselben Klasse verschmolzen werden, basiert auf der letzten Reihe der Partitionstabelle.

# Der tabellarische Ansatz

DFA $M = (Z, \Sigma, z_0, E)$

# Der tabellarische Ansatz

DFA $M = (Z, \Sigma, z_0, E)$

Schritte:

2.1 Bilde eine Partition $\mathcal{P}$ von $Z$ mit folgenden Klassen:
Endzustände $E$ (falls nicht leer) und Nicht-Endzustände $Z \setminus E$ (falls nicht leer).

# Der tabellarische Ansatz

DFA $M = (Z, \Sigma, z_0, E)$

Schritte:

2.1 Bilde eine Partition $\mathcal{P}$ von $Z$ mit folgenden Klassen:
Endzustände $E$ (falls nicht leer) und Nicht-Endzustände $Z \setminus E$ (falls nicht leer).

2.2 Wiederhole bis $\mathcal{P}$ sich nicht mehr verändert:

2.2.1 Für jede Klasse $K \in \mathcal{P}$ mit $|K| \geq 2$ und für jedes $a \in \Sigma$:

2.2.1.1 Berechne für jeden Zustand $z \in K$ die Klasse $L \in \mathcal{P}$, sodass $\delta(z, a) \in L$.

2.2.1.2 Partitioniere $K$ in Teilklassen je nach $L$.

2.2.1.3 Falls es mehrere Teilklassen gibt, ersetze $K$ in $\mathcal{P}$ durch die Teilklassen.

# Der tabellarische Ansatz

DFA $M = (Z, \Sigma, z_0, E)$

Schritte:

2.1 Bilde eine Partition $\mathcal{P}$ von $Z$ mit folgenden Klassen:
Endzustände $E$ (falls nicht leer) und Nicht-Endzustände $Z \setminus E$ (falls nicht leer).

2.2 Wiederhole bis $\mathcal{P}$ sich nicht mehr verändert:

2.2.1 Für jede Klasse $K \in \mathcal{P}$ mit $|K| \geq 2$ und für jedes $a \in \Sigma$:

2.2.1.1 Berechne für jeden Zustand $z \in K$ die Klasse $L \in \mathcal{P}$, sodass $\delta(z, a) \in L$.

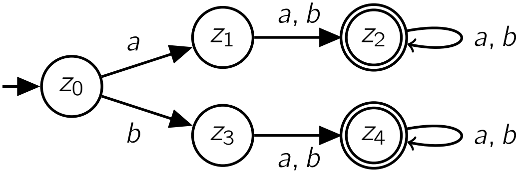
2.2.1.2 Partitioniere $K$ in Teilklassen je nach $L$.

2.2.1.3 Falls es mehrere Teilklassen gibt, ersetze $K$ in $\mathcal{P}$ durch die Teilklassen.

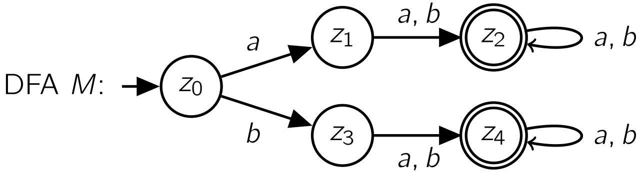
Am Ende besteht $\mathcal{P}$ aus den verschmolzenen Zuständen.

# Erstes Beispiel für den tabellarischen Ansatz

DFA $M$:



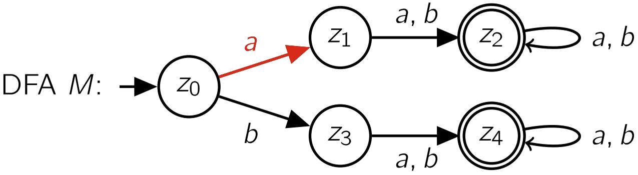
# Erstes Beispiel für den tabellarischen Ansatz




0.

| | | | | |
|---|---|---|---|---|
| $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |

# Erstes Beispiel für den tabellarischen Ansatz



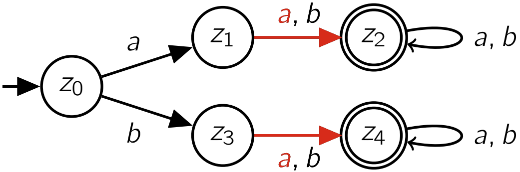

0.

| | | | | |
|---|---|---|---|---|
| $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |

- $z_0$ mit $a$ landet in der ersten Klasse.

# Erstes Beispiel für den tabellarischen Ansatz

DFA $M$:

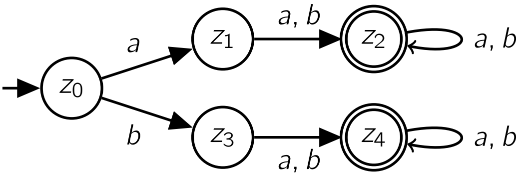



0.

| | | | | |
|---|---|---|---|---|
| $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |

- $z_0$ mit $a$ landet in der ersten Klasse.
- $z_1$ und $z_3$ mit $a$ landen in der zweiten Klasse.

# Erstes Beispiel für den tabellarischen Ansatz

DFA $M$: 

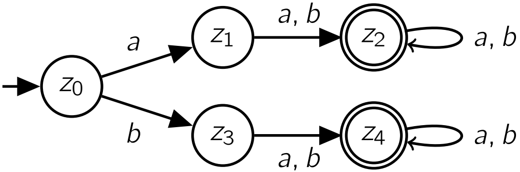

0.

| | | | | |
|---|---|---|---|---|
| $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |

- $z_0$ mit $a$ landet in der ersten Klasse.
- $z_1$ und $z_3$ mit $a$ landen in der zweiten Klasse.
- $z_0$ muss daher von $z_1$ und $z_3$ getrennt werden.

# Erstes Beispiel für den tabellarischen Ansatz

DFA $M$:




0.

| | | | | |
|---|---|---|---|---|
| $z_0$ | $z_1$ | $z_3$ \| | $z_2$ | $z_4$ |

- $z_0$ mit $a$ landet in der ersten Klasse.
- $z_1$ und $z_3$ mit $a$ landen in der zweiten Klasse.
- $z_0$ muss daher von $z_1$ und $z_3$ getrennt werden.

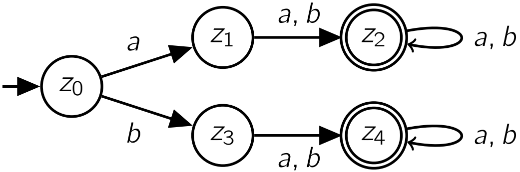
1.

| | | | | |
|---|---|---|---|---|
| $z_0$ \| | $z_1$ | $z_3$ \| | $z_2$ | $z_4$ |

mit $a$

# Erstes Beispiel für den tabellarischen Ansatz

DFA $M$:




0. | $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |
   |---|---|---|---|---|

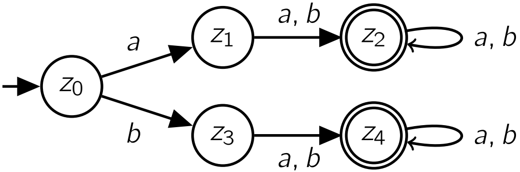
   - $z_0$ mit $a$ landet in der ersten Klasse.
   - $z_1$ und $z_3$ mit $a$ landen in der zweiten Klasse.
   - $z_0$ muss daher von $z_1$ und $z_3$ getrennt werden.

1. | $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |
   |---|---|---|---|---|

   mit $a$

   - Keine weitere Partition möglich.

# Erstes Beispiel für den tabellarischen Ansatz

DFA $M$: 


0. 

| | | | | |
|---|---|---|---|---|
| $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |

- $z_0$ mit $a$ landet in der ersten Klasse.
- $z_1$ und $z_3$ mit $a$ landen in der zweiten Klasse.
- $z_0$ muss daher von $z_1$ und $z_3$ getrennt werden.

1. 

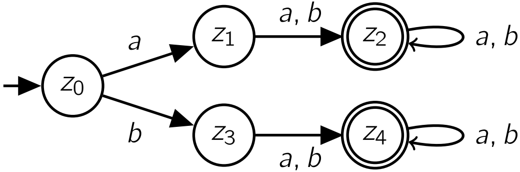
| | | | | |
|---|---|---|---|---|
| $z_0$ | $z_1$ | $z_3$ | $z_2$ | $z_4$ |

mit $a$

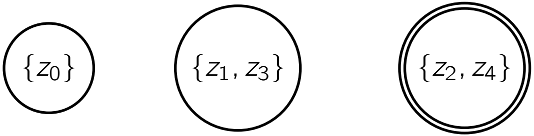
- Keine weitere Partition möglich.
- Der minimale DFA hat drei Zustände: $\{z_0\}$, $\{z_1, z_3\}$, $\{z_2, z_4\}$.

# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

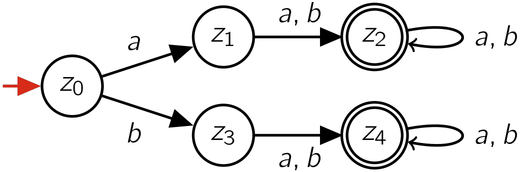



Minimaler DFA $M'$:

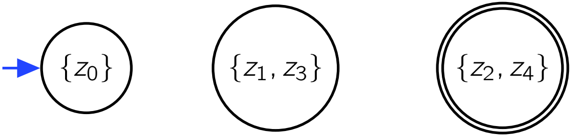


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

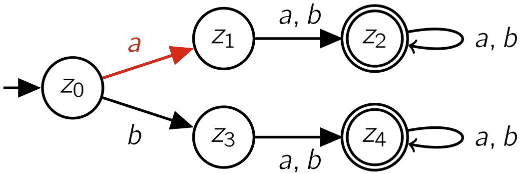



Minimaler DFA $M'$:

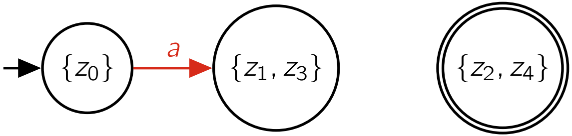


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

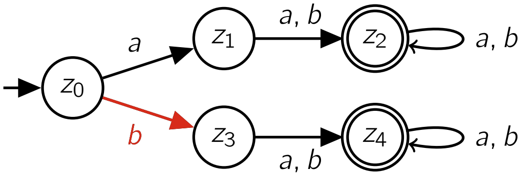



Minimaler DFA $M'$:

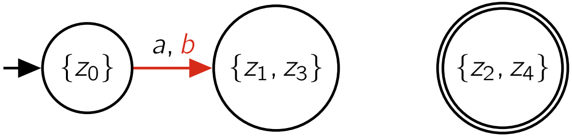


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

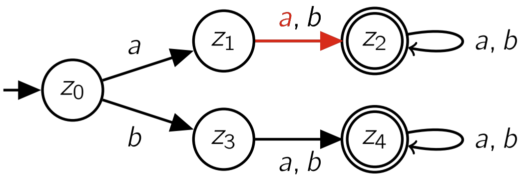



Minimaler DFA $M'$:

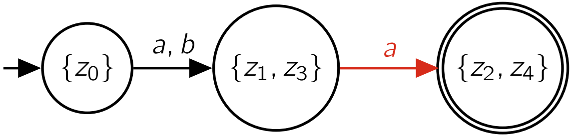


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

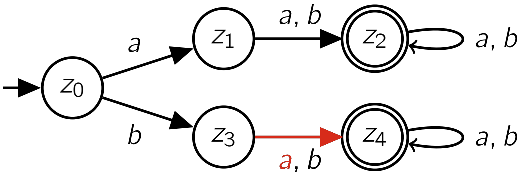



Minimaler DFA $M'$:

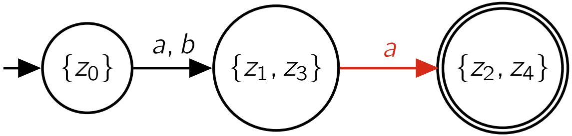


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

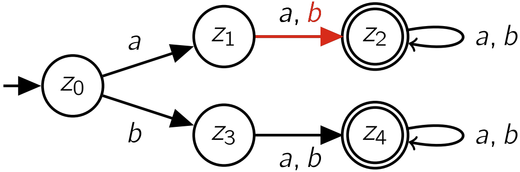



Minimaler DFA $M'$:

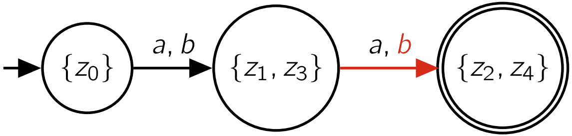


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

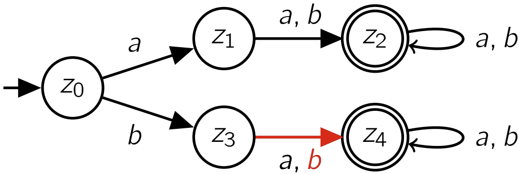



Minimaler DFA $M'$:

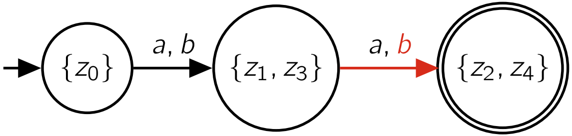


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

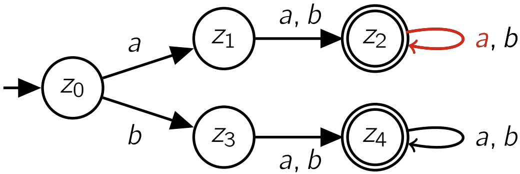



Minimaler DFA $M'$:

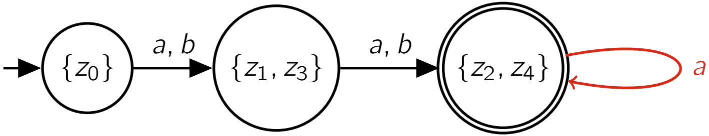


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

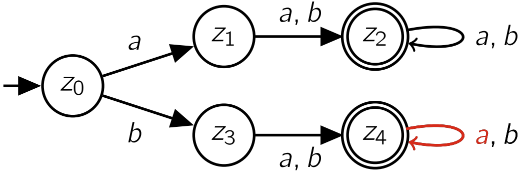



Minimaler DFA $M'$:

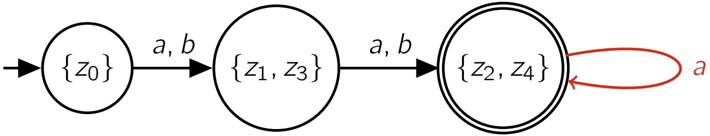


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

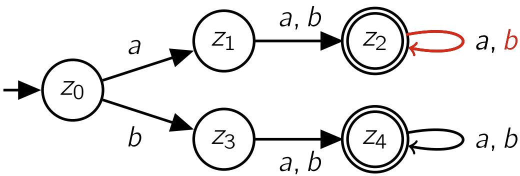



Minimaler DFA $M'$:

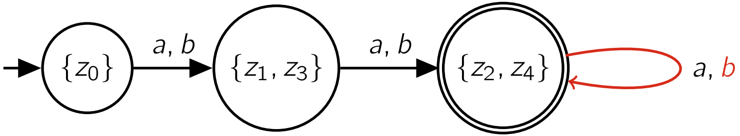


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

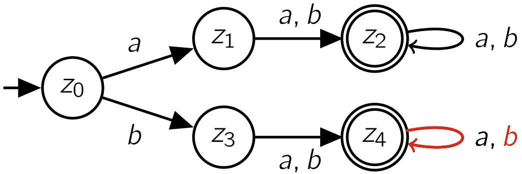



Minimaler DFA $M'$:

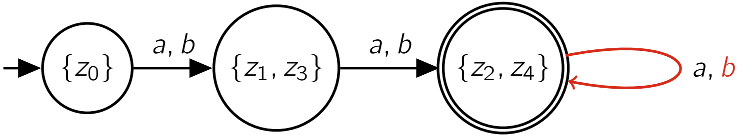


# Beispiel für die Konstruktion des minimalen DFA

DFA $M$:

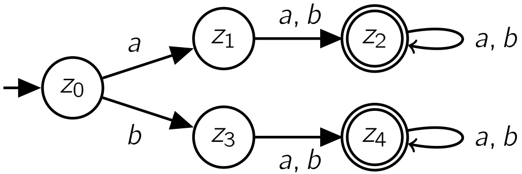



Minimaler DFA $M'$:

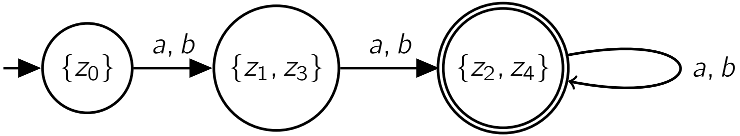


# Beispiel für die Konstruktion des minimalen DFA

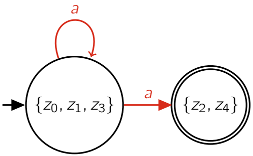
DFA $M$:



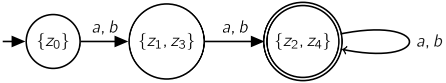

Minimaler DFA $M'$:



# Äquivalenz des graphischen und des tabellarischen Ansatzes



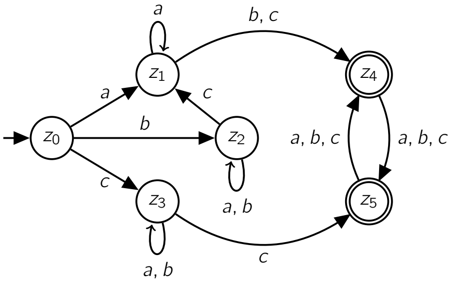

$$= \quad z_0 \quad z_1 \quad z_3 \;\Big|\; z_2 \quad z_4$$




$$= \quad z_0 \;\Big|\; z_1 \quad z_3 \;\Big|\; z_2 \quad z_4$$

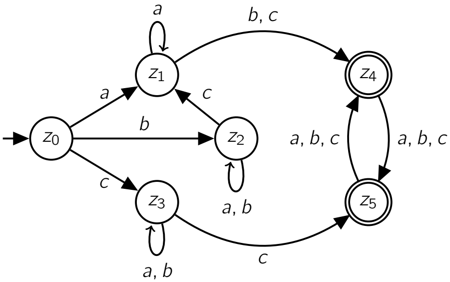
# Zweites Beispiel für den tabellarischen Ansatz

DFA $M$:



# Zweites Beispiel für den tabellarischen Ansatz

DFA $M$:

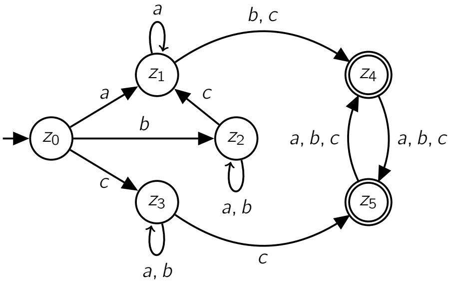



Partitionstabelle:

| | | | | | | |
|---|---|---|---|---|---|---|
| 0. | $z_0$ | $z_1$ | $z_2$ | $z_3$ | $z_4$ | $z_5$ |

# Zweites Beispiel für den tabellarischen Ansatz

DFA $M$:

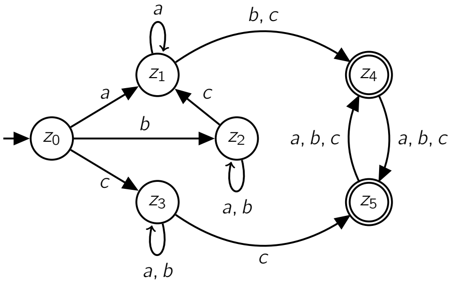



Partitionstabelle:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0. | $z_0$ | $z_1$ | $z_2$ | $z_3$ \| | $z_4$ | $z_5$ | |
| 1. | $z_1$ \| | $z_0$ | $z_2$ | $z_3$ \| | $z_4$ | $z_5$ | mit $b$ |

# Zweites Beispiel für den tabellarischen Ansatz

DFA $M$:

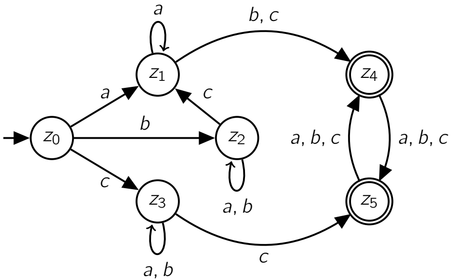



Partitionstabelle:

0. $z_0 \;\; z_1 \;\; z_2 \;\; z_3 \mid z_4 \;\; z_5$
1. $z_1 \mid z_0 \;\; z_2 \;\; z_3 \mid z_4 \;\; z_5$ mit $b$
2. $z_1 \mid z_0 \mid z_2 \mid z_3 \mid z_4 \;\; z_5$ mit $c$

# Zweites Beispiel für den tabellarischen Ansatz

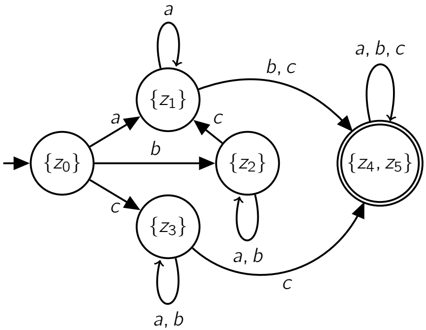
DFA $M$:




Partitionstabelle:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0. | $z_0$ | $z_1$ | $z_2$ | $z_3$ | $z_4$ | $z_5$ | |
| 1. | $z_1$ | $z_0$ | $z_2$ | $z_3$ | $z_4$ | $z_5$ | mit $b$ |
| 2. | $z_1$ | $z_0$ | $z_2$ | $z_3$ | $z_4$ | $z_5$ | mit $c$ |

Minimaler DFA $M'$:



# Äquivalenzklassenautomat

Beide Ansätze basieren auf dem Äquivalenzklassenautomaten.

**Definition**

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA.
Wir nennen zwei Zustände $z, z' \in Z$ äquivalent und schreiben $z \equiv_M z'$ (alternativ $z \equiv z'$), falls gilt: $\widetilde{\delta}(z, w) \in E$ g.d.w. $\widetilde{\delta}(z', w) \in E$ für alle $w \in \Sigma^*$.

# Äquivalenzklassenautomat

Beide Ansätze basieren auf dem Äquivalenzklassenautomaten.

**Definition**

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA.
Wir nennen zwei Zustände $z, z' \in Z$ äquivalent und schreiben $z \equiv_M z'$ (alternativ $z \equiv z'$), falls gilt: $\widetilde{\delta}(z, w) \in E$ g.d.w. $\widetilde{\delta}(z', w) \in E$ für alle $w \in \Sigma^*$.

Der Äquivalenzklassenautomat zu $M$ ist der DFA $M' = (Z', \Sigma, \delta', z_0', E')$ mit

$$
\begin{aligned}
Z' &:= \{[z]_\equiv \mid z \in Z\} \\
z_0' &:= [z_0]_\equiv \\
E' &:= \{[z]_\equiv \mid z \in E\} \\
\delta'([z]_\equiv, a) &:= [\delta(z, a)]_\equiv
\end{aligned}
$$

# Äquivalenzklassenautomat

Beide Ansätze basieren auf dem Äquivalenzklassenautomaten.

**Definition**

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA.
Wir nennen zwei Zustände $z, z' \in Z$ äquivalent und schreiben $z \equiv_M z'$ (alternativ $z \equiv z'$), falls gilt: $\widetilde{\delta}(z, w) \in E$ g.d.w. $\widetilde{\delta}(z', w) \in E$ für alle $w \in \Sigma^*$.

Der Äquivalenzklassenautomat zu $M$ ist der DFA $M' = (Z', \Sigma, \delta', z_0', E')$ mit

$$
\begin{aligned}
Z' &:= \{[z]_\equiv \mid z \in Z\} \\
z_0' &:= [z_0]_\equiv \\
E' &:= \{[z]_\equiv \mid z \in E\} \\
\delta'([z]_\equiv, a) &:= [\delta(z, a)]_\equiv
\end{aligned}
$$

Informell: Zwei Zustände sind äquivalent, wenn sie die gleiche „Sprache“ darstellen.

# Korrektheit und Minimalität des Äquivalenzklassenautomaten

**Satz**

Seien $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA und $M' = (Z', \Sigma, \delta', z_0', E')$ der Äquivalenzklassenautomat zu $M$. Dann gilt

1. $L(M') = L(M)$.
2. Falls alle Zustände in $Z$ von $z_0$ erreichbar sind, dann ist $M'$ minimal, d.h. jeder DFA $M''$ mit $L(M'') = L(M')$ hat mindestens so viele Zustände wie $M'$.

# Korrektheit und Minimalität des Äquivalenzklassenautomaten

**Satz**

Seien $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA und $M' = (Z', \Sigma, \delta', z_0', E')$ der Äquivalenzklassenautomat zu $M$. Dann gilt

1. $L(M') = L(M)$.
2. Falls alle Zustände in $Z$ von $z_0$ erreichbar sind, dann ist $M'$ minimal, d.h. jeder DFA $M''$ mit $L(M'') = L(M')$ hat mindestens so viele Zustände wie $M'$.

**Beweis**

1. Wir zeigen: $w \in L(M')$ g.d.w. $w \in L(M)$. Sei $w \in \Sigma^*$.

# Korrektheit und Minimalität des Äquivalenzklassenautomaten

**Satz**

Seien $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA und $M' = (Z', \Sigma, \delta', z_0', E')$ der Äquivalenzklassenautomat zu $M$. Dann gilt

1. $L(M') = L(M)$.
2. Falls alle Zustände in $Z$ von $z_0$ erreichbar sind, dann ist $M'$ minimal, d.h. jeder DFA $M''$ mit $L(M'') = L(M')$ hat mindestens so viele Zustände wie $M'$.

**Beweis**

1. Wir zeigen: $w \in L(M')$ g.d.w. $w \in L(M)$. Sei $w \in \Sigma^*$.

   $M$ durchläuft $z_0, \ldots, z_{|w|}$ entlang $w$ und akzeptiert $w$ g.d.w. $z_{|w|} \in E$.

# Korrektheit und Minimalität des Äquivalenzklassenautomaten

**Satz**

Seien $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA und $M' = (Z', \Sigma, \delta', z_0', E')$ der Äquivalenzklassenautomat zu $M$. Dann gilt

1. $L(M') = L(M)$.
2. Falls alle Zustände in $Z$ von $z_0$ erreichbar sind, dann ist $M'$ minimal, d.h. jeder DFA $M''$ mit $L(M'') = L(M')$ hat mindestens so viele Zustände wie $M'$.

**Beweis**

1. Wir zeigen: $w \in L(M')$ g.d.w. $w \in L(M)$. Sei $w \in \Sigma^*$.

   $M$ durchläuft $z_0, \dots, z_{|w|}$ entlang $w$ und akzeptiert $w$ g.d.w. $z_{|w|} \in E$.
   $M'$ durchläuft $[z_0]_\equiv, \dots, [z_{|w|}]_\equiv$ und akzeptiert $w$ g.d.w. $[z_{|w|}]_\equiv \in E'$.

# Korrektheit und Minimalität des Äquivalenzklassenautomaten

**Satz**

Seien $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA und $M' = (Z', \Sigma, \delta', z_0', E')$ der Äquivalenzklassenautomat zu $M$. Dann gilt

1. $L(M') = L(M)$.
2. Falls alle Zustände in $Z$ von $z_0$ erreichbar sind, dann ist $M'$ minimal, d.h. jeder DFA $M''$ mit $L(M'') = L(M')$ hat mindestens so viele Zustände wie $M'$.

**Beweis**

1. Wir zeigen: $w \in L(M')$ g.d.w. $w \in L(M)$. Sei $w \in \Sigma^*$.

   $M$ durchläuft $z_0, \ldots, z_{|w|}$ entlang $w$ und akzeptiert $w$ g.d.w. $z_{|w|} \in E$.
   $M'$ durchläuft $[z_0]_\equiv, \ldots, [z_{|w|}]_\equiv$ und akzeptiert $w$ g.d.w. $[z_{|w|}]_\equiv \in E'$.

   Da per Definition $[z_{|w|}]_\equiv \in E'$ äquivalent zu $z_{|w|} \in E$ ist, folgt, dass $L(M') = L(M)$.

# Korrektheit und Minimalität des Äquivalenzklassenautomaten

**Satz**

Seien $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA und $M' = (Z', \Sigma, \delta', z_0', E')$ der Äquivalenzklassenautomat zu $M$. Dann gilt

1. $L(M') = L(M)$.
2. Falls alle Zustände in $Z$ von $z_0$ erreichbar sind, dann ist $M'$ minimal, d.h. jeder DFA $M''$ mit $L(M'') = L(M')$ hat mindestens so viele Zustände wie $M'$.

**Beweis** (Fortsetzung)

2. Wird in späterer Vorlesung gezeigt (nur FSK). $\square$

# Algorithmus 3: Berechnung aller äquivalenten Zustände

**Eingabe:** DFA $M = (Z, \Sigma, \delta, z_0, E)$, der keine unerreichbaren Zustände hat
**Ausgabe:** Partition $\mathcal{P} = \{[z_1]_\equiv, \ldots, [z_m]_\equiv\}$ von $Z$
**Beginn**
  initialisiere Partition $\mathcal{P}$ mit $E$ (falls nicht leer) und $Z \setminus E$ (falls nicht leer);
  **wiederhole**
    **für jedes** $K = \{z_1, \ldots, z_m\} \in \mathcal{P}$ *mit* $|m| \geq 2$ **tue**
      **für jedes** $a \in \Sigma$ **tue**
        berechne die Partition $\mathcal{Q}$ von $\{z_1, \ldots, z_m\}$ über dem Wert von $[\delta(z_i, a)]$ für jedes $i$;
        $\mathcal{P} := (\mathcal{P} \setminus \{K\}) \cup \mathcal{Q}$;
      **Ende**
    **Ende**
  **bis** *sich* $\mathcal{P}$ *nicht mehr verändert*;
  **return** $\mathcal{P}$
**Ende**

# Korrektheit von Algorithmus 3

**Satz**

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA, der keine unerreichbaren Zustände hat.
Algorithmus 3 berechnet die Äquivalenzklassen bezüglich $\equiv$.

# Korrektheit von Algorithmus 3

**Satz**

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA, der keine unerreichbaren Zustände hat. Algorithmus 3 berechnet die Äquivalenzklassen bezüglich $\equiv$.

**Beweis** Wir führen den Beweis in zwei Schritten:

1. Wenn $z \equiv z'$, dann sind $z$ und $z'$ in derselben Klasse von $\mathcal{P}$.
2. Wenn $z \not\equiv z'$, dann sind $z$ und $z'$ in verschiedenen Klassen von $\mathcal{P}$.

# Korrektheit des tabellarischen Ansatzes

**Beweis** (Fortsetzung)

1. Wir zeigen die Kontraposition: Wenn $z$ und $z'$ in verschiedenen Klassen von $\mathcal{P}$ sind, dann $z \not\equiv z'$.

# Korrektheit des tabellarischen Ansatzes

**Beweis** (Fortsetzung)

1. Wir zeigen die Kontraposition: Wenn $z$ und $z'$ in verschiedenen Klassen von $\mathcal{P}$ sind, dann $z \not\equiv z'$. Und um $z \not\equiv z'$ zu zeigen, reicht es, ein Wort $w$ zu finden, sodass $\widetilde{\delta}(z, w) \in E$ und $\widetilde{\delta}(z', w) \notin E$ oder umgekehrt.

# Korrektheit des tabellarischen Ansatzes

**Beweis** (Fortsetzung)

1. Wir zeigen die Kontraposition: Wenn $z$ und $z'$ in verschiedenen Klassen von $\mathcal{P}$ sind, dann $z \not\equiv z'$. Und um $z \not\equiv z'$ zu zeigen, reicht es, ein Wort $w$ zu finden, sodass $\widetilde{\delta}(z, w) \in E$ und $\widetilde{\delta}(z', w) \notin E$ oder umgekehrt.

   Durch Induktion über die Anzahl $n$ der Schleifeniterationen bis $z$ und $z'$ getrennt wurden.

# Korrektheit des tabellarischen Ansatzes

**Beweis** (Fortsetzung)

1. Wir zeigen die Kontraposition: Wenn $z$ und $z'$ in verschiedenen Klassen von $\mathcal{P}$ sind, dann $z \not\equiv z'$. Und um $z \not\equiv z'$ zu zeigen, reicht es, ein Wort $w$ zu finden, sodass $\widetilde{\delta}(z, w) \in E$ und $\widetilde{\delta}(z', w) \notin E$ oder umgekehrt.

   Durch Induktion über die Anzahl $n$ der Schleifeniterationen bis $z$ und $z'$ getrennt wurden.

   - Fall $n = 0$: Da $z$ und $z'$ schon vor der ersten Iteration getrennt wurden, müssen $z \in E$ und $z' \notin E$ oder umgekehrt gelten. Wir nehmen $w = \varepsilon$: $\widetilde{\delta}(z, \varepsilon) \in E$ und $\widetilde{\delta}(z', \varepsilon) \notin E$ oder umgekehrt.

# Korrektheit des tabellarischen Ansatzes

**Beweis** (Fortsetzung)

1. Wir zeigen die Kontraposition: Wenn $z$ und $z'$ in verschiedenen Klassen von $\mathcal{P}$ sind, dann $z \not\equiv z'$. Und um $z \not\equiv z'$ zu zeigen, reicht es, ein Wort $w$ zu finden, sodass $\widetilde{\delta}(z, w) \in E$ und $\widetilde{\delta}(z', w) \notin E$ oder umgekehrt.

   Durch Induktion über die Anzahl $n$ der Schleifeniterationen bis $z$ und $z'$ getrennt wurden.

   - Fall $n = 0$: Da $z$ und $z'$ schon vor der ersten Iteration getrennt wurden, müssen $z \in E$ und $z' \notin E$ oder umgekehrt gelten. Wir nehmen $w = \varepsilon$: $\widetilde{\delta}(z, \varepsilon) \in E$ und $\widetilde{\delta}(z', \varepsilon) \notin E$ oder umgekehrt.
   - Fall $n > 0$: Da $z$ und $z'$ in Iteration $n$ getrennt wurden, muss es $a \in \Sigma$ geben, sodass $\delta(z, a)$ und $\delta(z', a)$ schon in Iteration $n - 1$ getrennt waren. Die Induktionshypothese liefert ein Wort $w'$ mit $\widetilde{\delta}(\delta(z, a), w') \in E$ und $\widetilde{\delta}(\delta(z', a), w') \notin E$ oder umgekehrt. Wir nehmen $w = aw'$: $\widetilde{\delta}(z, aw') \in E$ und $\widetilde{\delta}(z', aw') \notin E$ oder umgekehrt.

# Korrektheit des tabellarischen Ansatzes

**Beweis** (Fortsetzung)

2. Wir müssen zeigen, dass wenn $z \not\equiv z'$, dann sind $z$ und $z'$ in zwei verschiedenen Klassen von $\mathcal{P}$. Sei $w$ ein Wort sodass $\widetilde{\delta}(z, w) \in E$ und $\widetilde{\delta}(z', w) \notin E$ oder umgekehrt.

   Durch Induktion über $|w|$.

   - Fall $w = \varepsilon$: Dann haben wir $\widetilde{\delta}(z, \varepsilon) \in E$ und $\widetilde{\delta}(z', \varepsilon) \notin E$ oder umgekehrt. Die Initialisierungsphase sorgt dafür, dass $z$ und $z'$ getrennt werden.
   - Fall $w$ ist von der Form $aw'$: Dann haben wir $\widetilde{\delta}(z, aw') = \widetilde{\delta}(\delta(z, a), w') \in E$ und $\widetilde{\delta}(z', aw') = \widetilde{\delta}(\delta(z', a), w') \notin E$ oder umgekehrt. Per Induktionshypothese müssen $\delta(z, a)$ und $\delta(z', a)$ getrennt sein. Der Algorithmus muss $z$ und $z'$ trennen, wenn $a$ betrachtet wird. $\square$

# Algorithmus 4: Minimierung von DFAs

**Eingabe:** DFA $M = (Z, \Sigma, \delta, z_0, E)$
**Ausgabe:** Minimaler DFA $M'$ mit $L(M') = L(M)$
**Beginn**

entferne Zustände aus $M$, die nicht vom Startzustand aus erreichbar sind;
berechne äquivalente Zustände mit Algorithmus 3;
erzeuge den Äquivalenzklassenautomat, indem die berechneten äquivalenten Zustände verschmolzen werden;

**Ende**