F2 — Automaten und formale Sprachen

Berndt Farwer

Fachbereich Informatik

AB „Theoretische Grundlagen der Informatik“ (TGI)

Universität Hamburg

farwer@informatik.uni-hamburg.de

Potenzautomat

- Idee: Die Potenzmenge von Z wird als neue Zustandsmenge verwendet.

Potenzautomat

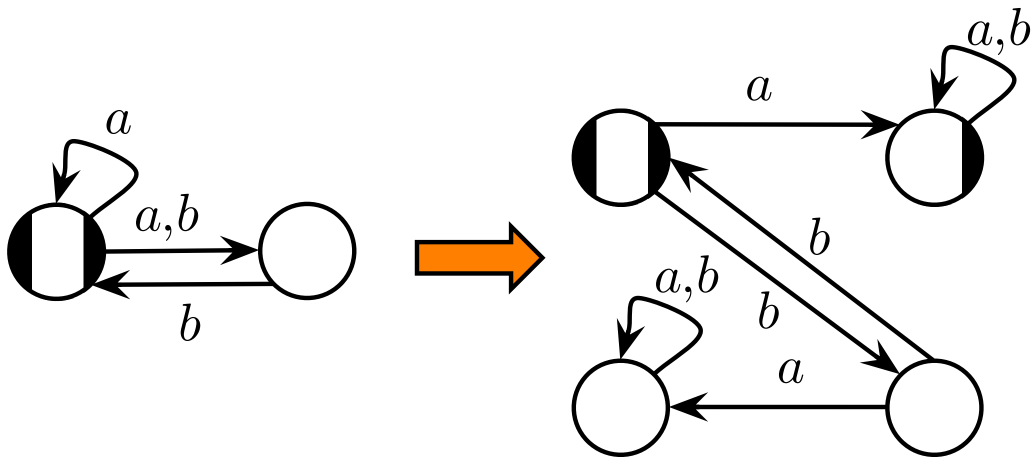
- Idee: Die Potenzmenge von Z wird als neue Zustandsmenge verwendet.
- Werden alle Zustände benötigt?

Potenzautomat

- Idee: Die Potenzmenge von Z wird als neue Zustandsmenge verwendet.
- Werden alle Zustände benötigt?
- Es gibt NFAs, die tatsächlich alle $2^{|Z|}$ Zustände erfordern!

Potenzautomat

- Idee: Die Potenzmenge von Z wird als neue Zustandsmenge verwendet.
- Werden alle Zustände benötigt?
- Es gibt NFAs, die tatsächlich alle $2^{|Z|}$ Zustände erfordern!
- **Beispiel:**



Endliche Sprachen $\in \mathcal{R}eg$

- **Theorem:** Jede endliche Sprache ist regulär.

Endliche Sprachen $\in \mathcal{R}eg$

- **Theorem:** Jede endliche Sprache ist regulär.
- **Beweis:** Sei $L = \{w_1, w_2, \ldots, w_n\}$ endlich, d.h. $|L| = n \in I\!N$.

Endliche Sprachen $\in \mathcal{R}eg$

- **Theorem:** Jede endliche Sprache ist regulär.
- **Beweis:** Sei $L = \{w_1, w_2, \ldots, w_n\}$ endlich, d.h. $|L| = n \in I\!N$.
  - Es gibt endliches Alphabet Σ mit $L \subseteq \Sigma$.

Endliche Sprachen $\in \mathcal{R}eg$

- **Theorem:** Jede endliche Sprache ist regulär.
- **Beweis:** Sei $L = \{w_1, w_2, \ldots, w_n\}$ endlich, d.h. $|L| = n \in I\!N$.
  - Es gibt endliches Alphabet Σ mit $L \subseteq \Sigma$.
  - NFA $A = (\{z, z'\}, \Sigma, K, \{z\}, \{z'\})$ mit

$$K := \{(z, w_i, z') \mid w_i \in L\}$$

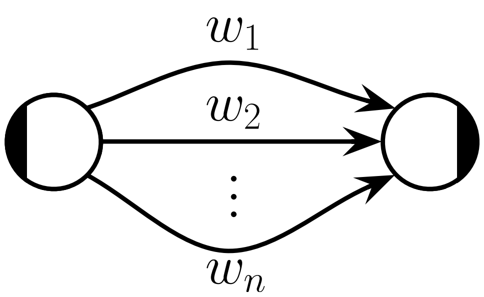
    akzeptiert L.

Endliche Sprachen $\in \mathcal{R}eg$

- **Theorem:** Jede endliche Sprache ist regulär.
- **Beweis:** Sei $L = \{w_1, w_2, \ldots, w_n\}$ endlich, d.h. $|L| = n \in \mathbb{N}$.
  - Es gibt endliches Alphabet Σ mit $L \subseteq \Sigma$.
  - NFA $A = (\{z, z'\}, \Sigma, K, \{z\}, \{z'\})$ mit

$$K := \{(z, w_i, z') \mid w_i \in L\}$$

    akzeptiert L.
  - graphisch:



Rationale Ausdrücke

Definition: Die **rationalen Ausdrücke** über einem endlichen Alphabet Σ sind definiert durch:

- $\emptyset$ ist ein rationaler Ausdruck, der die (*leere*) Menge $M_\emptyset := \emptyset$ beschreibt.

Rationale Ausdrücke

Definition: Die **rationalen Ausdrücke** über einem endlichen Alphabet Σ sind definiert durch:

- $\emptyset$ ist ein rationaler Ausdruck, der die (*leere*) Menge $M_\emptyset := \emptyset$ beschreibt.
- Für jedes $a \in \Sigma$ ist a ein rationaler Ausdruck, der die Menge $M_a := \{a\}$ beschreibt.

Rationale Ausdrücke

Definition: Die **rationalen Ausdrücke** über einem endlichen Alphabet Σ sind definiert durch:

- $\emptyset$ ist ein rationaler Ausdruck, der die (*leere*) Menge $M_\emptyset := \emptyset$ beschreibt.
- Für jedes $a \in \Sigma$ ist a ein rationaler Ausdruck, der die Menge $M_a := \{a\}$ beschreibt.
- Sind A und B rationale Ausdrücke, welche die Mengen M_A und M_B beschreiben, dann sind **induktiv** folgende rationalen Ausdrücke definiert:

Rationale Ausdrücke

Definition: Die **rationalen Ausdrücke** über einem endlichen Alphabet Σ sind definiert durch:

- $\emptyset$ ist ein rationaler Ausdruck, der die (*leere*) Menge $M_\emptyset := \emptyset$ beschreibt.
- Für jedes $a \in \Sigma$ ist a ein rationaler Ausdruck, der die Menge $M_a := \{a\}$ beschreibt.
- Sind A und B rationale Ausdrücke, welche die Mengen M_A und M_B beschreiben, dann sind **induktiv** folgende rationalen Ausdrücke definiert:
  - $(A + B)$ beschreibt die Menge $M_A \cup M_B$,

Rationale Ausdrücke

Definition: Die **rationalen Ausdrücke** über einem endlichen Alphabet Σ sind definiert durch:

- $\emptyset$ ist ein rationaler Ausdruck, der die (*leere*) Menge $M_\emptyset := \emptyset$ beschreibt.
- Für jedes $a \in \Sigma$ ist a ein rationaler Ausdruck, der die Menge $M_a := \{a\}$ beschreibt.
- Sind A und B rationale Ausdrücke, welche die Mengen M_A und M_B beschreiben, dann sind **induktiv** folgende rationalen Ausdrücke definiert:
  - $(A + B)$ beschreibt die Menge $M_A \cup M_B$,
  - $A \cdot B$ beschreibt die Menge $M_A \cdot M_B$

Rationale Ausdrücke

Definition: Die **rationalen Ausdrücke** über einem endlichen Alphabet Σ sind definiert durch:

- $\emptyset$ ist ein rationaler Ausdruck, der die (*leere*) Menge $M_\emptyset := \emptyset$ beschreibt.
- Für jedes $a \in \Sigma$ ist a ein rationaler Ausdruck, der die Menge $M_a := \{a\}$ beschreibt.
- Sind A und B rationale Ausdrücke, welche die Mengen M_A und M_B beschreiben, dann sind **induktiv** folgende rationalen Ausdrücke definiert:
  - $(A + B)$ beschreibt die Menge $M_A \cup M_B$,
  - $A \cdot B$ beschreibt die Menge $M_A \cdot M_B$
  - $(A)^*$ beschreibt die Menge M_A^*

Rationale Ausdrücke

Definition: Die **rationalen Ausdrücke** über einem endlichen Alphabet Σ sind definiert durch:

- $\emptyset$ ist ein rationaler Ausdruck, der die (*leere*) Menge $M_\emptyset := \emptyset$ beschreibt.
- Für jedes $a \in \Sigma$ ist a ein rationaler Ausdruck, der die Menge $M_a := \{a\}$ beschreibt.
- Sind A und B rationale Ausdrücke, welche die Mengen M_A und M_B beschreiben, dann sind **induktiv** folgende rationalen Ausdrücke definiert:
  - $(A + B)$ beschreibt die Menge $M_A \cup M_B$,
  - $A \cdot B$ beschreibt die Menge $M_A \cdot M_B$
  - $(A)^*$ beschreibt die Menge M_A^*
  - $(A)^+$ beschreibt die Menge M_A^+

Sprachfamilie $\mathcal{R}at$

- Es bezeichne $\mathcal{R}at(\Sigma)$ die Familie aller mit rationalen Ausdrücken beschreibbaren Teilmengen von Σ^*.

Sprachfamilie $\mathcal{R}at$

- Es bezeichne $\mathcal{R}at(\Sigma)$ die Familie aller mit rationalen Ausdrücken beschreibbaren Teilmengen von Σ^*.
- Zusammenfassend: $\mathcal{R}at := \bigcup_{\Sigma \text{ ist endl. Alphabet}} \mathcal{R}at(\Sigma)$.

Sprachfamilie $\mathcal{R}at$

- Es bezeichne $\mathcal{R}at(\Sigma)$ die Familie aller mit rationalen Ausdrücken beschreibbaren Teilmengen von Σ^*.
- Zusammenfassend: $\mathcal{R}at := \bigcup_{\Sigma \text{ ist endl. Alphabet}} \mathcal{R}at(\Sigma)$.
- Regeln zur Klammerersparnis:

Sprachfamilie $\mathcal{R}at$

- Es bezeichne $\mathcal{R}at(\Sigma)$ die Familie aller mit rationalen Ausdrücken beschreibbaren Teilmengen von Σ^*.
- Zusammenfassend: $\mathcal{R}at := \bigcup_{\Sigma \text{ ist endl. Alphabet}} \mathcal{R}at(\Sigma)$.
- Regeln zur Klammerersparnis:
  - unäre Operatoren vor binären ($*$ vor $\cdot$ und $+$)

Sprachfamilie $\mathcal{R}at$

- Es bezeichne $\mathcal{R}at(\Sigma)$ die Familie aller mit rationalen Ausdrücken beschreibbaren Teilmengen von Σ^*.
- Zusammenfassend: $\mathcal{R}at := \bigcup_{\Sigma \text{ ist endl. Alphabet}} \mathcal{R}at(\Sigma)$.
- Regeln zur Klammerersparnis:
  - unäre Operatoren vor binären ($*$ vor $\cdot$ und $+$)
  - Punkt vor Strich ($\cdot$ vor $+$)

Anwendung: Unix-CLI

- `grep` und `egrep` auf Unix-Betriebssystemen und in Editoren:

Anwendung: Unix-CLI

- `grep` und `egrep` auf Unix-Betriebssystemen und in Editoren:
  - Suche mit „regulären Ausdrücken“

Anwendung: Unix-CLI

- `grep` und `egrep` auf Unix-Betriebssystemen und in Editoren:
  - Suche mit „regulären Ausdrücken“
- Gibt man z.B. den Befehl

  ```
  egrep  ^Th...i.$   /usr/dict/duden
  ```

  ein, so könnten folgende Wörter gefunden und angezeigt werden:

Anwendung: Unix-CLI

- `grep` und `egrep` auf Unix-Betriebssystemen und in Editoren:
  - Suche mit „regulären Ausdrücken“
- Gibt man z.B. den Befehl

  ```
  egrep  ^Th...i.$  /usr/dict/duden
  ```

  ein, so könnten folgende Wörter gefunden und angezeigt werden:
  - `Thermik`

Anwendung: Unix-CLI

- `grep` und `egrep` auf Unix-Betriebssystemen und in Editoren:
  - Suche mit „regulären Ausdrücken“
- Gibt man z.B. den Befehl

  ```
  egrep ^Th...i.$ /usr/dict/duden
  ```

  ein, so könnten folgende Wörter gefunden und angezeigt werden:
  - `Thermik`
  - `Theorie`

Anwendung: Unix-CLI

- grep und egrep auf Unix-Betriebssystemen und in Editoren:
  - Suche mit „regulären Ausdrücken“
- Gibt man z.B. den Befehl

  ```
  egrep   ^Th...i.$   /usr/dict/duden
  ```

  ein, so könnten folgende Wörter gefunden und angezeigt werden:
  - `Thermik`
  - `Theorie`
  - `Thespis`

egrep-Notation

rationaler Ausdruck	egrep-Notation
Zeilenanfang	^
Zeilenende	$
c	c
(Gesamtalphabet:) Σ	.
$r + \emptyset^*$	$r?$
r^*	$r*$
r^+	$r+$
$r + s$	$r\|s$
$r \cdot s$	$r\ s$
(r)	(r)

Grundannahmen

- Sei L eine reguläre Sprache.

Grundannahmen

- Sei L eine reguläre Sprache.
- In allen weiteren Konstruktionen dürfen wir nach belieben voraussetzen, dass wir einen der folgenden Automaten vorliegen haben, der L akzeptiert:

Grundannahmen

- Sei L eine reguläre Sprache.
- In allen weiteren Konstruktionen dürfen wir nach belieben voraussetzen, dass wir einen der folgenden Automaten vorliegen haben, der L akzeptiert:
  - DFA

Grundannahmen

- Sei L eine reguläre Sprache.
- In allen weiteren Konstruktionen dürfen wir nach belieben voraussetzen, dass wir einen der folgenden Automaten vorliegen haben, der L akzeptiert:
  - DFA
  - vDFA

Grundannahmen

- Sei L eine reguläre Sprache.
- In allen weiteren Konstruktionen dürfen wir nach belieben voraussetzen, dass wir einen der folgenden Automaten vorliegen haben, der L akzeptiert:
  - DFA
  - vDFA
  - NFA

Grundannahmen

- Sei L eine reguläre Sprache.
- In allen weiteren Konstruktionen dürfen wir nach belieben voraussetzen, dass wir einen der folgenden Automaten vorliegen haben, der L akzeptiert:
  - DFA
  - vDFA
  - NFA
  - λ-freier NFA

Grundannahmen

- Sei L eine reguläre Sprache.
- In allen weiteren Konstruktionen dürfen wir nach belieben voraussetzen, dass wir einen der folgenden Automaten vorliegen haben, der L akzeptiert:
  - DFA
  - vDFA
  - NFA
  - λ-freier NFA
  - buchstabierender NFA

Abschlussoperator: Vereinigung

- **Theorem:** Mit $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$ ist $L_1 \cup L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.

Abschlussoperator: Vereinigung

- **Theorem:** Mit $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$ ist $L_1 \cup L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines NFA für $L_1 \cup L_2$)

Abschlussoperator: Vereinigung

- **Theorem:** Mit $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$ ist $L_1 \cup L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines NFA für $L_1 \cup L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ mit $L_1 = L(A)$ und $L_2 = L(B)$ zwei vDFAs.

Abschlussoperator: Vereinigung

- **Theorem:** Mit $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$ ist $L_1 \cup L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines NFA für $L_1 \cup L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ mit $L_1 = L(A)$ und $L_2 = L(B)$ zwei vDFAs.
  - Der NFA für $L_1 \cup L_2$ wird definiert durch: $C_{A \cup B} := (Z_1 \uplus Z_2, \Sigma_1 \cup \Sigma_2, K_3, Z_{3,0}, Z_{3,\text{end}})$

Abschlussoperator: Vereinigung

- **Theorem:** Mit $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$ ist $L_1 \cup L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines NFA für $L_1 \cup L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ mit $L_1 = L(A)$ und $L_2 = L(B)$ zwei vDFAs.
  - Der NFA für $L_1 \cup L_2$ wird definiert durch: $C_{A \cup B} := (Z_1 \uplus Z_2, \Sigma_1 \cup \Sigma_2, K_3, Z_{3,0}, Z_{3,\text{end}})$
    - $Z_{3,\text{start}} := \{z_{1,0}, z_{2,0}\}$

Abschlussoperator: Vereinigung

- **Theorem:** Mit $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$ ist $L_1 \cup L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines NFA für $L_1 \cup L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ mit $L_1 = L(A)$ und $L_2 = L(B)$ zwei vDFAs.
  - Der NFA für $L_1 \cup L_2$ wird definiert durch: $C_{A \cup B} := (Z_1 \uplus Z_2, \Sigma_1 \cup \Sigma_2, K_3, Z_{3,0}, Z_{3,\text{end}})$
    - $Z_{3,\text{start}} := \{z_{1,0}, z_{2,0}\}$
    - $Z_{3,\text{end}} := Z_{1,\text{end}} \cup Z_{2,\text{end}}$

Abschlussoperator: Vereinigung

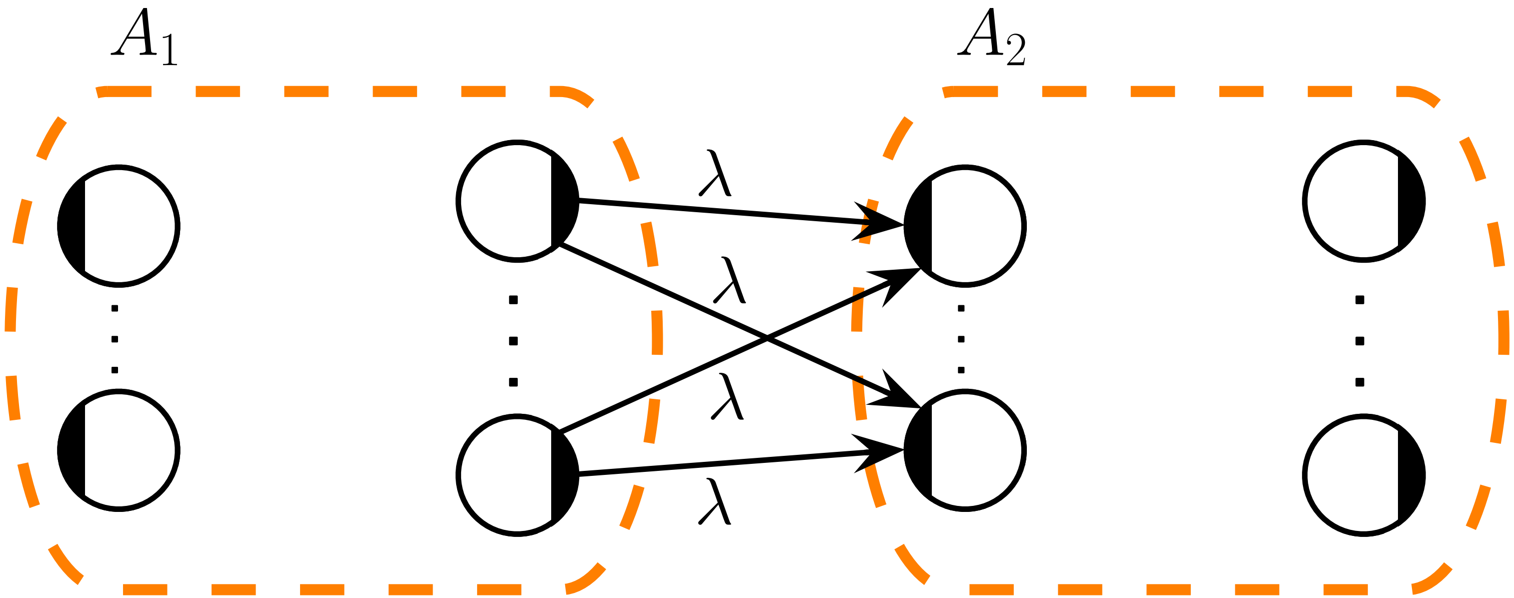
- **Theorem:** Mit $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$ ist $L_1 \cup L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines NFA für $L_1 \cup L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ mit $L_1 = L(A)$ und $L_2 = L(B)$ zwei vDFAs.
  - Der NFA für $L_1 \cup L_2$ wird definiert durch: $C_{A \cup B} := (Z_1 \uplus Z_2, \Sigma_1 \cup \Sigma_2, K_3, Z_{3,0}, Z_{3,\text{end}})$
    - $Z_{3,\text{start}} := \{z_{1,0}, z_{2,0}\}$
    - $Z_{3,\text{end}} := Z_{1,\text{end}} \cup Z_{2,\text{end}}$
    - $K_3 := \{(p_1, x, \delta_1(p_1, x)) \mid p_1 \in Z_1, x \in \Sigma_1\}$ $\cup \{(p_2, x, \delta_2(p_2, x)) \mid p_2 \in Z_2, x \in \Sigma_2\}$

Produkt (informell)

- Bereits gezeigt: Abschluss gegen Produktbildung.

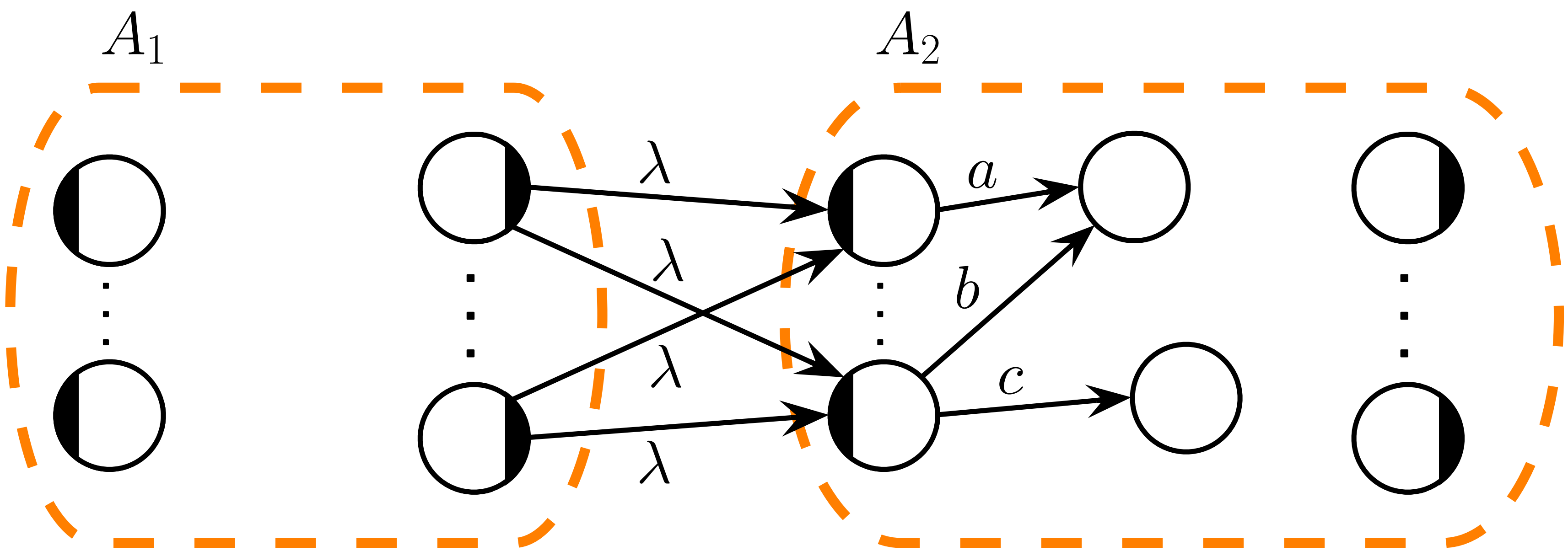
Produkt (informell)

- Bereits gezeigt: Abschluss gegen Produktbildung.
- Alternative Konstruktion führt zu einem buchstabierenden FA:



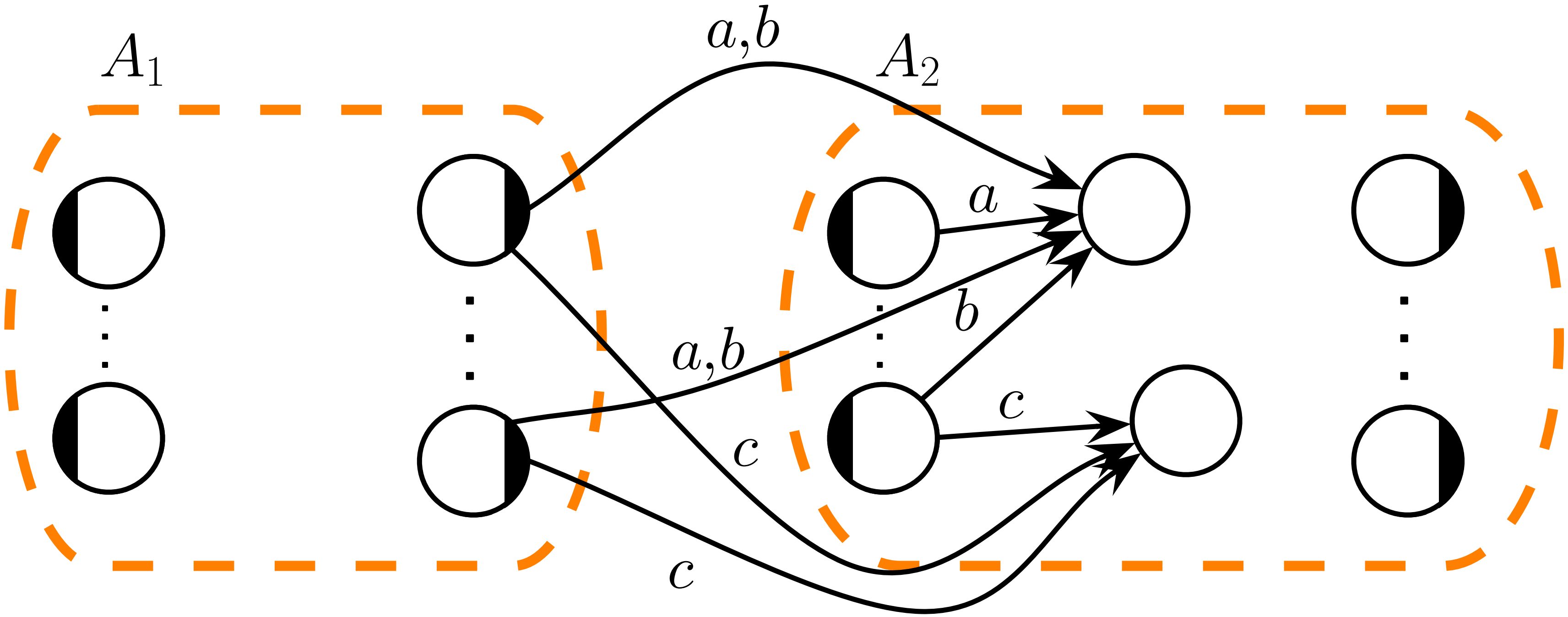
Produkt (informell)

- Bereits gezeigt: Abschluss gegen Produktbildung.
- Alternative Konstruktion führt zu einem buchstabierenden FA:



Produkt (informell)

- Bereits gezeigt: Abschluss gegen Produktbildung.
- Alternative Konstruktion führt zu einem buchstabierenden FA:



Abschlussoperator: Produkt

- **Theorem:** Seien $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist $L_1 \cdot L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.

Abschlussoperator: Produkt

- **Theorem:** Seien $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist $L_1 \cdot L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für $L_1 \cdot L_2$)

Abschlussoperator: Produkt

- **Theorem:** Seien $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist $L_1 \cdot L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für $L_1 \cdot L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\mathsf{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\mathsf{end}})$ vollständige DFA's mit $Z_1 \cap Z_2 = \emptyset$, $L_1 = L(A)$ und $L_2 = L(B)$.

Abschlussoperator: Produkt

- **Theorem:** Seien $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist $L_1 \cdot L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für $L_1 \cdot L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ vollständige DFA's mit $Z_1 \cap Z_2 = \emptyset$, $L_1 = L(A)$ und $L_2 = L(B)$.
  - Definiere NFA $C_{A \cdot B} :=$ $(Z_1 \cup Z_2, \Sigma_1 \cup \Sigma_2, K_1 \cup K_2 \cup K_3, z_{1,0}, Z_{2,\text{end}})$ mit

Abschlussoperator: Produkt

- **Theorem:** Seien $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist $L_1 \cdot L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für $L_1 \cdot L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ vollständige DFA's mit $Z_1 \cap Z_2 = \emptyset$, $L_1 = L(A)$ und $L_2 = L(B)$.
  - Definiere NFA $C_{A \cdot B} := (Z_1 \cup Z_2, \Sigma_1 \cup \Sigma_2, K_1 \cup K_2 \cup K_3, z_{1,0}, Z_{2,\text{end}})$ mit
    - $K_1 := \{(p_1, x, \delta_1(p_1, x)) \mid x \in \Sigma_1, p_1 \in Z_1\}$

Abschlussoperator: Produkt

- **Theorem:** Seien $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist $L_1 \cdot L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für $L_1 \cdot L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ vollständige DFA's mit $Z_1 \cap Z_2 = \emptyset$, $L_1 = L(A)$ und $L_2 = L(B)$.
  - Definiere NFA $C_{A \cdot B} :=$ $(Z_1 \cup Z_2, \Sigma_1 \cup \Sigma_2, K_1 \cup K_2 \cup K_3, z_{1,0}, Z_{2,\text{end}})$ mit
    - $K_1 := \{(p_1, x, \delta_1(p_1, x)) \mid x \in \Sigma_1, p_1 \in Z_1\}$
    - $K_2 := \{(p_2, x, \delta_2(p_2, x)) \mid x \in \Sigma_2, p_2 \in Z_2\}$

Abschlussoperator: Produkt

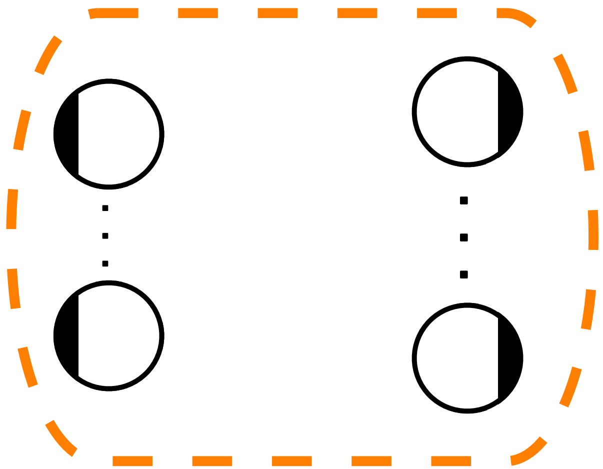
- **Theorem:** Seien $L_1 \in \mathcal{A}kz(\Sigma_1)$ und $L_2 \in \mathcal{A}kz(\Sigma_2)$, dann ist $L_1 \cdot L_2 \in \mathcal{A}kz(\Sigma_1 \cup \Sigma_2)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für $L_1 \cdot L_2$)
  - Seien $A := (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,\text{end}})$ und $B := (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,\text{end}})$ vollständige DFA's mit $Z_1 \cap Z_2 = \emptyset$, $L_1 = L(A)$ und $L_2 = L(B)$.
  - Definiere NFA $C_{A \cdot B} :=$ $(Z_1 \cup Z_2, \Sigma_1 \cup \Sigma_2, K_1 \cup K_2 \cup K_3, z_{1,0}, Z_{2,\text{end}})$ mit
    - $K_1 := \{(p_1, x, \delta_1(p_1, x)) \mid x \in \Sigma_1, p_1 \in Z_1\}$
    - $K_2 := \{(p_2, x, \delta_2(p_2, x)) \mid x \in \Sigma_2, p_2 \in Z_2\}$
    - $K_3 := \{(p_1, x, \delta_2(z_{2,0}, x)) \mid x \in \Sigma_2, p_1 \in Z_{1,\text{end}}\}$ die A und B verbindenden, Nicht-λ-Kanten.

Stern (informell)

- Gegeben: Ein endlicher Automat, der L akzeptiert.

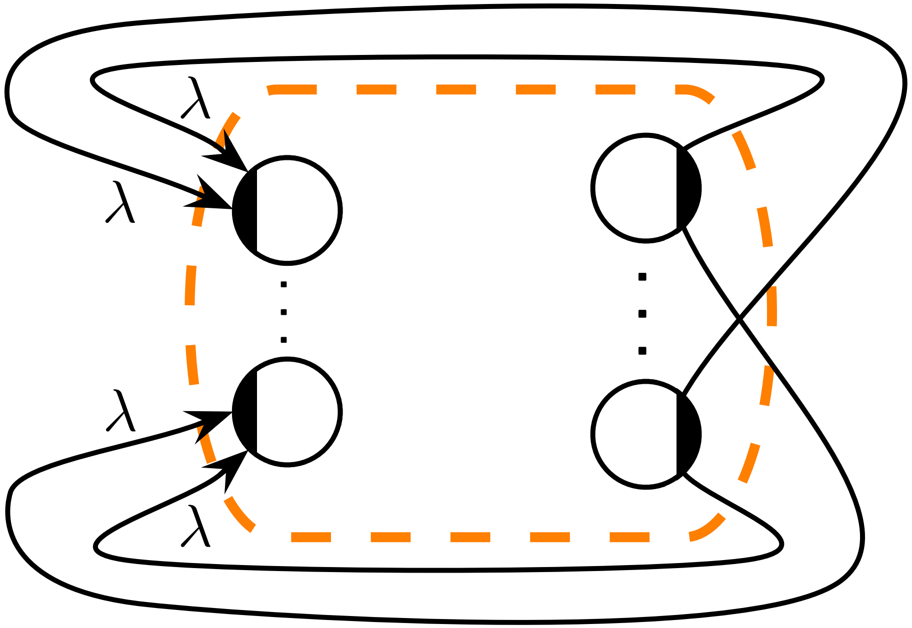
Stern (informell)

- Gegeben: Ein endlicher Automat, der L akzeptiert.
- Konstruktion eines (buchstabierenden) FA, der die Sprache L^* akzeptiert:



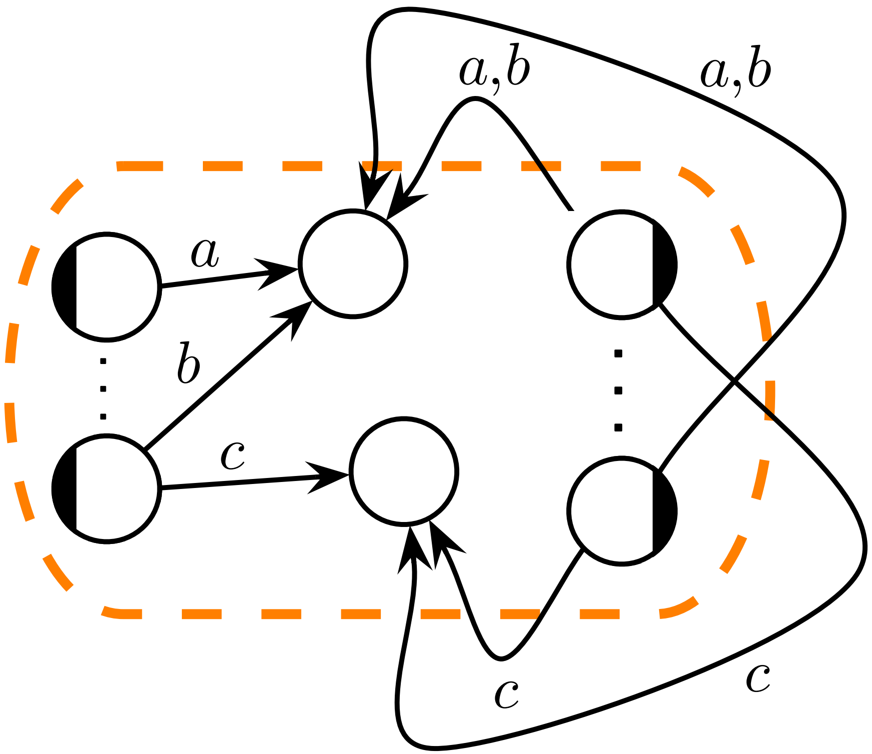
Stern (informell)

- Gegeben: Ein endlicher Automat, der L akzeptiert.
- Konstruktion eines (buchstabierenden) FA, der die Sprache L^* akzeptiert:



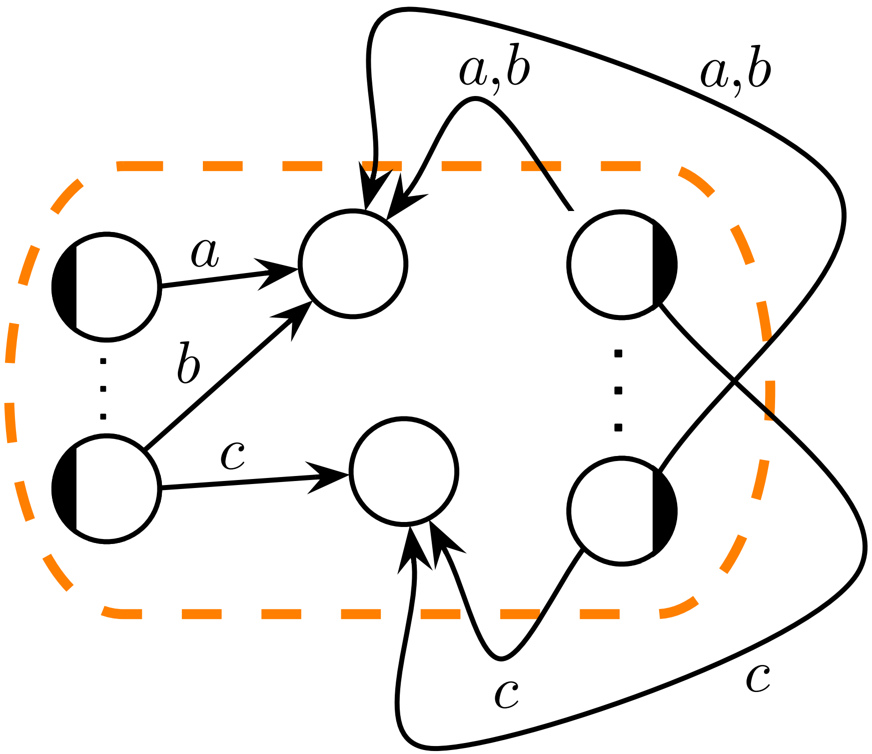
Stern (informell)

- Gegeben: Ein endlicher Automat, der L akzeptiert.
- Konstruktion eines (buchstabierenden) FA, der die Sprache L^* akzeptiert:



Stern (informell)

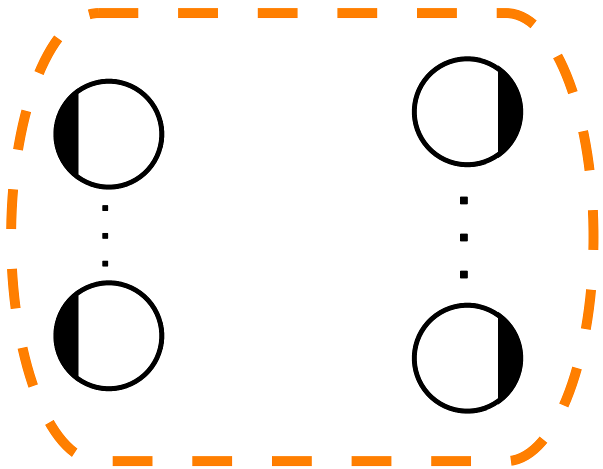
- Gegeben: Ein endlicher Automat, der L akzeptiert.
- Konstruktion eines (buchstabierenden) FA, der die Sprache L^* akzeptiert:



…aber dieser Automat akzeptiert nicht immer $\lambda \in L^*$!

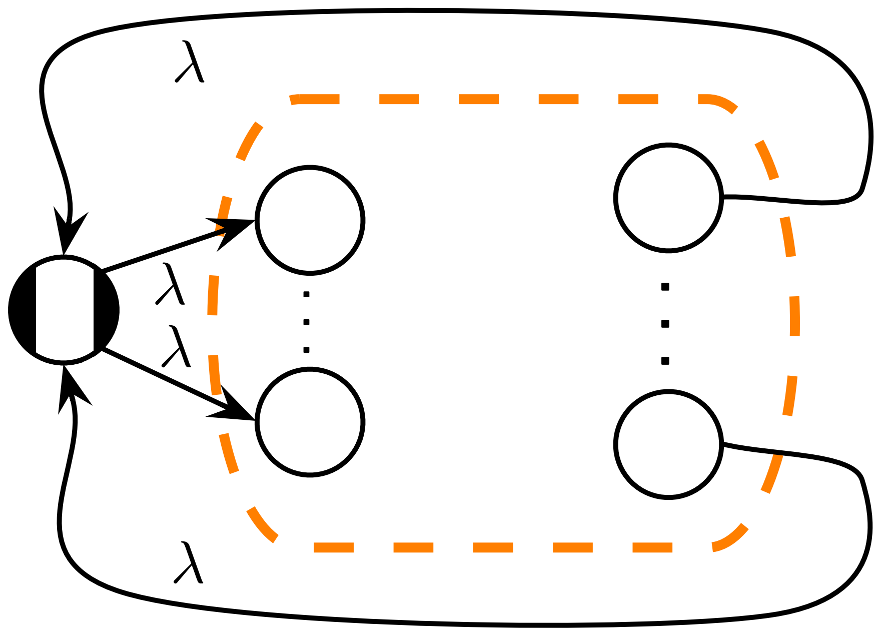
Stern (informell)

- Gegeben: Ein endlicher Automat, der L akzeptiert.
- Konstruktion eines (buchstabierenden) FA, der die Sprache L^* akzeptiert:



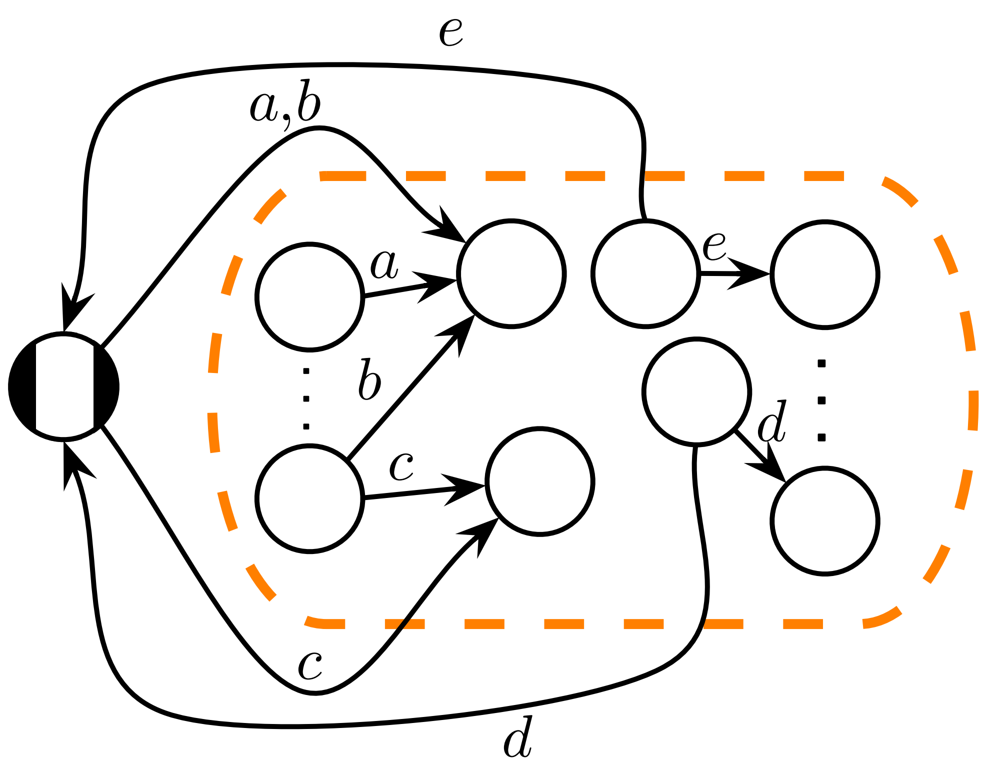
Stern (informell)

- Gegeben: Ein endlicher Automat, der L akzeptiert.
- Konstruktion eines (buchstabierenden) FA, der die Sprache L^* akzeptiert:



Stern (informell)

- Gegeben: Ein endlicher Automat, der L akzeptiert.
- Konstruktion eines (buchstabierenden) FA, der die Sprache L^* akzeptiert:



Abschlussoperator: Stern

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

Abschlussoperator: Stern

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $L^* \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für L^*)

Abschlussoperator: Stern

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $L^* \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für L^*)
  - Sei $A := (Z, \Sigma, \delta, z_0, Z_{\mathsf{end}})$ ein vDFA mit $L = L(A)$.

Abschlussoperator: Stern

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $L^* \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für L^*)
  - Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein vDFA mit $L = L(A)$.
  - Definiere buchstabierenden NFA $C_{A^*} := (Z \uplus \{p\}, \Sigma, K_1 \cup K_2 \cup K_3, \{p\}, \{p\})$ mit $K := K_1 \cup K_2 \cup K_3$, wobei

Abschlussoperator: Stern

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $L^* \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für L^*)
  - Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein vDFA mit $L = L(A)$.
  - Definiere buchstabierenden NFA $C_{A^*} := (Z \uplus \{p\}, \Sigma, K_1 \cup K_2 \cup K_3, \{p\}, \{p\})$ mit $K := K_1 \cup K_2 \cup K_3$, wobei
    - $K_1 := \{(z, x, \delta(p, x)) \mid x \in \Sigma, z \in Z\}$ die zum vDFA A gehörende Kantenmenge,

Abschlussoperator: Stern

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $L^* \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für L^*)
  - Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein vDFA mit $L = L(A)$.
  - Definiere buchstabierenden NFA $C_{A^*} := (Z \uplus \{p\}, \Sigma, K_1 \cup K_2 \cup K_3, \{p\}, \{p\})$ mit $K := K_1 \cup K_2 \cup K_3$, wobei
    - $K_1 := \{(z, x, \delta(p, x)) \mid x \in \Sigma, z \in Z\}$ die zum vDFA A gehörende Kantenmenge,
    - $K_2 := \{(p, x, \delta(z_0, x)) \mid x \in X\}$ und

Abschlussoperator: Stern

- Sei $L \in \mathcal{A}kz(\Sigma)$, dann ist auch $L^* \in \mathcal{A}kz(\Sigma)$.
- **Beweis:** (Konstruktion eines buchstabierenden NFA für L^*)
  - Sei $A := (Z, \Sigma, \delta, z_0, Z_{\text{end}})$ ein vDFA mit $L = L(A)$.
  - Definiere buchstabierenden NFA $C_{A^*} := (Z \uplus \{p\}, \Sigma, K_1 \cup K_2 \cup K_3, \{p\}, \{p\})$ mit $K := K_1 \cup K_2 \cup K_3$, wobei
    - $K_1 := \{(z, x, \delta(p, x)) \mid x \in \Sigma, z \in Z\}$ die zum vDFA A gehörende Kantenmenge,
    - $K_2 := \{(p, x, \delta(z_0, x)) \mid x \in X\}$ und
    - $K_3 := \{(z, x, p) \mid x \in \Sigma, \delta(z, x) \in Z_{\text{end}}\}$ die Verbindungen mit dem neuen Startzustand sind.

Einfache Mengen sind regulär!

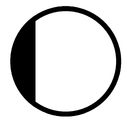
- Die leere Menge ist regulär.

Einfache Mengen sind regulär!

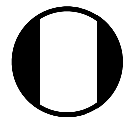
- Die leere Menge ist regulär.
- Die Menge $\emptyset^* = \{\lambda\}$ ist regulär.

Einfache Mengen sind regulär!

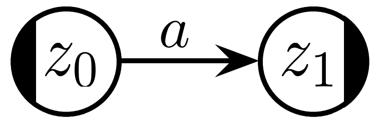
- Die leere Menge ist regulär.



- Die Menge $\emptyset^* = \{\lambda\}$ ist regulär.

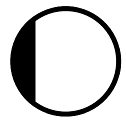


- Die Menge $\{a\}$ ist regulär.

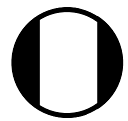


Einfache Mengen sind regulär!

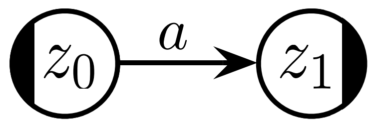
- Die leere Menge ist regulär.



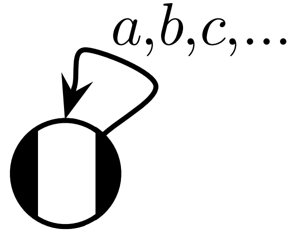
- Die Menge $\emptyset^* = \{\lambda\}$ ist regulär.



- Die Menge $\{a\}$ ist regulär.

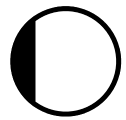


- Σ^* ist regulär.

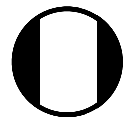


Einfache Mengen sind regulär!

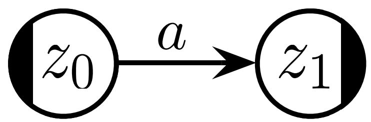
- Die leere Menge ist regulär.

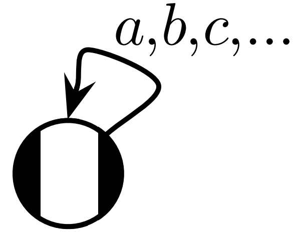


- Die Menge $\emptyset^* = \{\lambda\}$ ist regulär.



- Die Menge $\{a\}$ ist regulär.

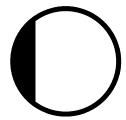




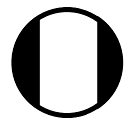
- Σ^* ist regulär.
- Für jede Menge M ist M^+ lediglich eine abkürzende Schreibweise für $M \cdot M^*$.

Einfache Mengen sind regulär!

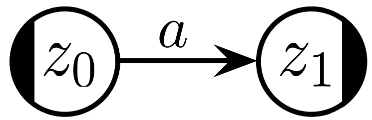
- Die leere Menge ist regulär.

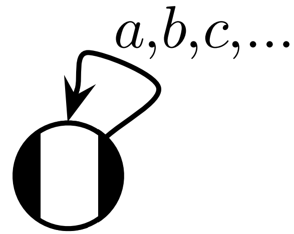


- Die Menge $\emptyset^* = \{\lambda\}$ ist regulär.



- Die Menge $\{a\}$ ist regulär.





- Σ^* ist regulär.
- Für jede Menge M ist M^+ lediglich eine abkürzende Schreibweise für $M \cdot M^*$.
- Für jede Menge M ist $M \cdot \emptyset = \emptyset \cdot M = \emptyset$.

Satz von Kleene

- Wegen der Abgeschlossenheit von $\mathcal{R}eg$ gegen $\cup$, $\cdot$ und $*$ ist klar, dass $\mathcal{R}at \subseteq \mathcal{R}eg$ gilt.

Satz von Kleene

- Wegen der Abgeschlossenheit von $\mathcal{R}eg$ gegen $\cup$, $\cdot$ und $*$ ist klar, dass $\mathcal{R}at \subseteq \mathcal{R}eg$ gilt.
- **Theorem:** Für jedes Alphabet Σ gilt $\mathcal{A}kz(\Sigma) = \mathcal{R}at(\Sigma)$ und folglich $\mathcal{R}eg = \mathcal{R}at$.

Satz von Kleene

- Wegen der Abgeschlossenheit von $\mathcal{R}eg$ gegen $\cup$, $\cdot$ und $*$ ist klar, dass $\mathcal{R}at \subseteq \mathcal{R}eg$ gilt.
- **Theorem:** Für jedes Alphabet Σ gilt $\mathcal{A}kz(\Sigma) = \mathcal{R}at(\Sigma)$ und folglich $\mathcal{R}eg = \mathcal{R}at$.
- **Beweisidee:** Wir stellen eine Rekursionsformel auf, die von jeder regulären Sprache zeigt, dass sie durch endlich häufiges Anwenden von $\cup$, $\cdot$ und $*$ darstellbar ist.

Satz von Kleene

- Wegen der Abgeschlossenheit von $\mathcal{R}eg$ gegen $\cup$, $\cdot$ und $*$ ist klar, dass $\mathcal{R}at \subseteq \mathcal{R}eg$ gilt.
- **Theorem:** Für jedes Alphabet Σ gilt $\mathcal{A}kz(\Sigma) = \mathcal{R}at(\Sigma)$ und folglich $\mathcal{R}eg = \mathcal{R}at$.
- **Beweisidee:** Wir stellen eine Rekursionsformel auf, die von jeder regulären Sprache zeigt, dass sie durch endlich häufiges Anwenden von $\cup$, $\cdot$ und $*$ darstellbar ist.
  - Konstruktion eines rationalen Ausdruckes für einen beliebigen DFA.

Satz von Kleene

- Wegen der Abgeschlossenheit von $\mathcal{R}eg$ gegen $\cup$, $\cdot$ und $*$ ist klar, dass $\mathcal{R}at \subseteq \mathcal{R}eg$ gilt.
- **Theorem:** Für jedes Alphabet Σ gilt $\mathcal{A}kz(\Sigma) = \mathcal{R}at(\Sigma)$ und folglich $\mathcal{R}eg = \mathcal{R}at$.
- **Beweisidee:** Wir stellen eine Rekursionsformel auf, die von jeder regulären Sprache zeigt, dass sie durch endlich häufiges Anwenden von $\cup$, $\cdot$ und $*$ darstellbar ist.
  - Konstruktion eines rationalen Ausdruckes für einen beliebigen DFA.
  - Konstruktion funktioniert auch für (buchstabierenden) NFA.

$\mathcal{A}kz(\Sigma) \subseteq \mathcal{R}at(\Sigma)$

- Sei $A = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, Z_{\textsf{end}})$ ein DFA

$\mathcal{A}kz(\Sigma) \subseteq \mathcal{R}at(\Sigma)$

- Sei $A = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, Z_{\text{end}})$ ein DFA
- Sei für $0 \leq k \leq n$ und $1 \leq i,\ j \leq n$:

$\mathcal{A}kz(\Sigma) \subseteq \mathcal{R}at(\Sigma)$

- Sei $A = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, Z_{\mathsf{end}})$ ein DFA
- Sei für $0 \leq k \leq n$ und $1 \leq i,\ j \leq n$:

$$R_{i,j}^k := \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \delta(z_i, w) = z_j \;\wedge\; [(w = uv \wedge u \neq \lambda \wedge \\ v \neq \lambda \wedge \delta(z_i, u) = z_r) \rightarrow r \leq k] \end{array} \right\}$$

$\mathcal{A}kz(\Sigma) \subseteq \mathcal{R}at(\Sigma)$

- Sei $A = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, Z_{\text{end}})$ ein DFA
- Sei für $0 \leq k \leq n$ und $1 \leq i,\ j \leq n$:

$$R^k_{i,j} := \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \delta(z_i, w) = z_j \;\; \wedge \;\; [(w = uv \wedge u \neq \lambda \wedge \\ v \neq \lambda \wedge \delta(z_i, u) = z_r) \rightarrow r \leq k] \end{array} \right\}$$

- $R^k_{i,j}$ enthält alle Wörter, die auf Pfaden von z_i nach z_j über die Menge $\{z_1, \ldots, z_k\}$ von **Zwischenzuständen** gelesen werden können.

$\mathcal{A}kz(\Sigma) \subseteq \mathcal{R}at(\Sigma)$

- Sei $A = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, Z_{\mathsf{end}})$ ein DFA
- Sei für $0 \leq k \leq n$ und $1 \leq i,\ j \leq n$:

$$R^k_{i,j} := \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \delta(z_i, w) = z_j \;\wedge\; [(w = uv \wedge u \neq \lambda \wedge \\ v \neq \lambda \wedge \delta(z_i, u) = z_r) \rightarrow r \leq k] \end{array} \right\}$$

- $R^k_{i,j}$ enthält alle Wörter, die auf Pfaden von z_i nach z_j über die Menge $\{z_1, \ldots, z_k\}$ von **Zwischenzuständen** gelesen werden können.
- Insbesondere $R^0_{i,j} = \{x \in \Sigma \cup \{\lambda\} \mid \delta(z_i, x) = z_j$ oder $(x = \lambda) \wedge (i = j)\}$

$Akz(\Sigma) \subseteq \mathcal{R}at(\Sigma)$

- Sei $A = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, Z_{\text{end}})$ ein DFA
- Sei für $0 \leq k \leq n$ und $1 \leq i,\ j \leq n$:

$$R_{i,j}^k := \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \delta(z_i, w) = z_j \;\wedge\; [(w = uv \wedge u \neq \lambda \wedge \\ v \neq \lambda \wedge \delta(z_i, u) = z_r) \rightarrow r \leq k] \end{array} \right\}$$

- $R_{i,j}^k$ enthält alle Wörter, die auf Pfaden von z_i nach z_j über die Menge $\{z_1, \ldots, z_k\}$ von **Zwischenzuständen** gelesen werden können.
- Insbesondere $R_{i,j}^0 = \{x \in \Sigma \cup \{\lambda\} \mid \delta(z_i, x) = z_j$ oder $(x = \lambda) \wedge (i = j)\}$
- Somit: $L(A) = \bigcup_{z_j \in Z_{\text{end}}} R_{1,j}^n$

Beweis: $R_{i,j}^k$-Rekursionsformel

$$R_{i,j}^k = \begin{cases} \{x \mid \delta(z_i, x) = z_j \text{ oder } (x = \lambda) \wedge (i = j)\}, & \text{falls } k = 0 \\ R_{i,j}^{k-1} \cup R_{i,k}^{k-1} \cdot (R_{k,k}^{k-1})^* \cdot R_{k,j}^{k-1}, & \text{falls } k \geq 1. \end{cases}$$

- **Induktions-Basis:** $R_{i,j}^0$ entspricht der Definition

Beweis: $R^k_{i,j}$-Rekursionsformel

$$R^k_{i,j} = \begin{cases} \{x \mid \delta(z_i, x) = z_j \text{ oder } (x = \lambda) \wedge (i = j)\}, & \text{falls } k = 0 \\ R^{k-1}_{i,j} \cup R^{k-1}_{i,k} \cdot (R^{k-1}_{k,k})^* \cdot R^{k-1}_{k,j}, & \text{falls } k \geq 1. \end{cases}$$

- **Induktions-Basis:** $R^0_{i,j}$ entspricht der Definition
- **Induktions-Schritt:** Sei $k = m \geq 0$.
  $R^m_{i,j}$ korrekt $\Rightarrow$ $R^{m+1}_{i,j}$ korrekt

Beweis: $R_{i,j}^k$-Rekursionsformel

$$R_{i,j}^k = \begin{cases} \{x \mid \delta(z_i, x) = z_j \text{ oder } (x = \lambda) \wedge (i = j)\}, & \text{falls } k = 0 \\ R_{i,j}^{k-1} \cup R_{i,k}^{k-1} \cdot (R_{k,k}^{k-1})^* \cdot R_{k,j}^{k-1}, & \text{falls } k \geq 1. \end{cases}$$

- **Induktions-Basis:** $R_{i,j}^0$ entspricht der Definition
- **Induktions-Schritt:** Sei $k = m \geq 0$.
  $R_{i,j}^m$ korrekt $\Rightarrow$ $R_{i,j}^{m+1}$ korrekt
  - Kommt z_{m+1} nicht vor $\Rightarrow w \in R_{i,j}^m$.

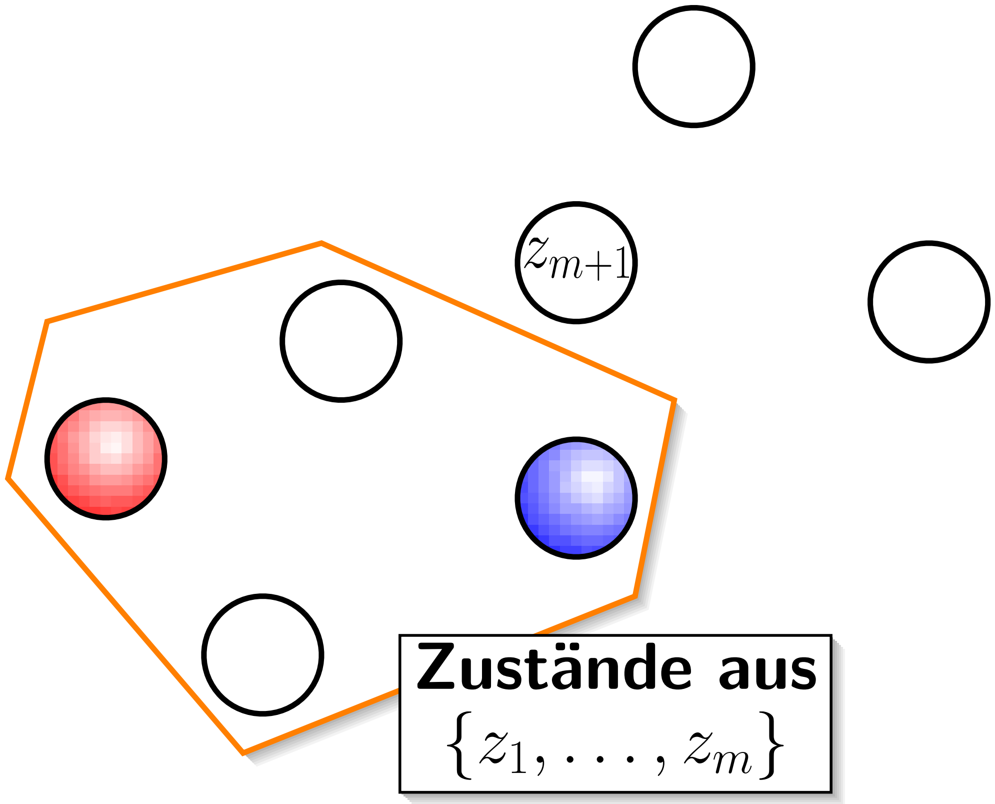
Beweis: $R^k_{i,j}$-Rekursionsformel

$$R^k_{i,j} = \begin{cases} \{x \mid \delta(z_i, x) = z_j \text{ oder } (x = \lambda) \wedge (i = j)\}, & \text{falls } k = 0 \\ R^{k-1}_{i,j} \cup R^{k-1}_{i,k} \cdot (R^{k-1}_{k,k})^* \cdot R^{k-1}_{k,j}, & \text{falls } k \geq 1. \end{cases}$$

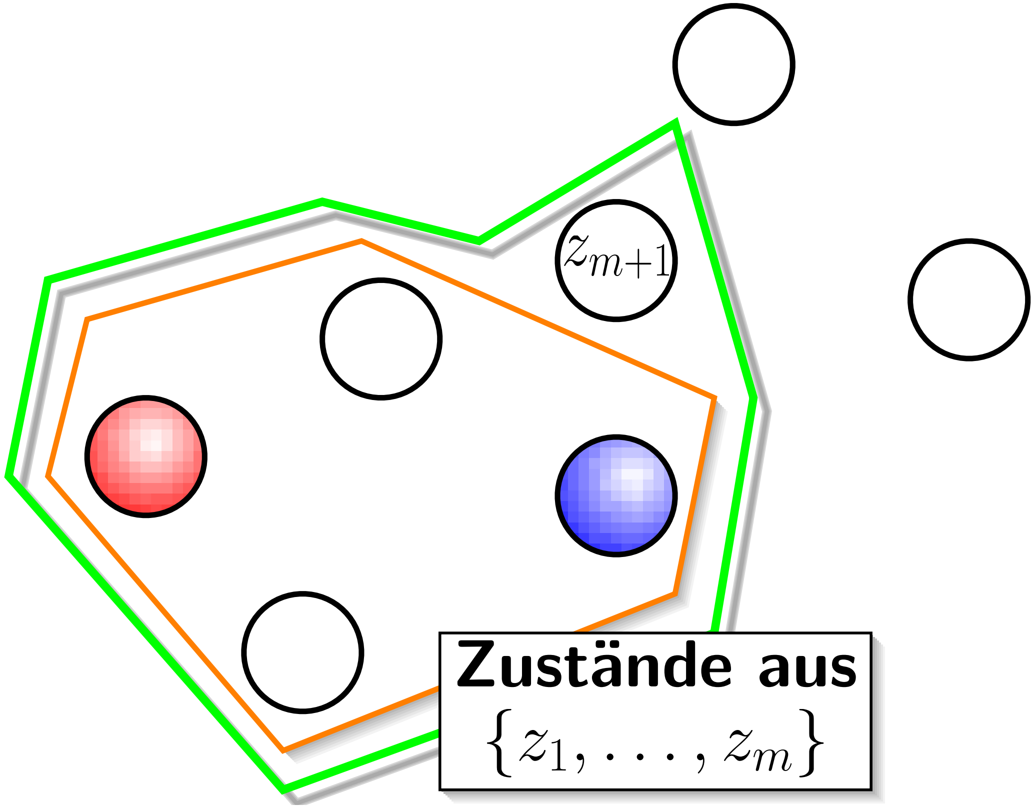
- **Induktions-Basis:** $R^0_{i,j}$ entspricht der Definition
- **Induktions-Schritt:** Sei $k = m \geq 0$.
  $R^m_{i,j}$ korrekt $\Rightarrow$ $R^{m+1}_{i,j}$ korrekt
  - Kommt z_{m+1} nicht vor $\Rightarrow w \in R^m_{i,j}$.
  - Kommt z_{m+1} auf dem Pfad vor $\Rightarrow$ Zerlegung:

$$z_i \xrightarrow[u]{*} z_{m+1} \xrightarrow[v_1]{*} z_{m+1} \xrightarrow[v_2]{*} \cdots \xrightarrow[v_r]{*} z_{m+1} \xrightarrow[w]{*} z_j$$

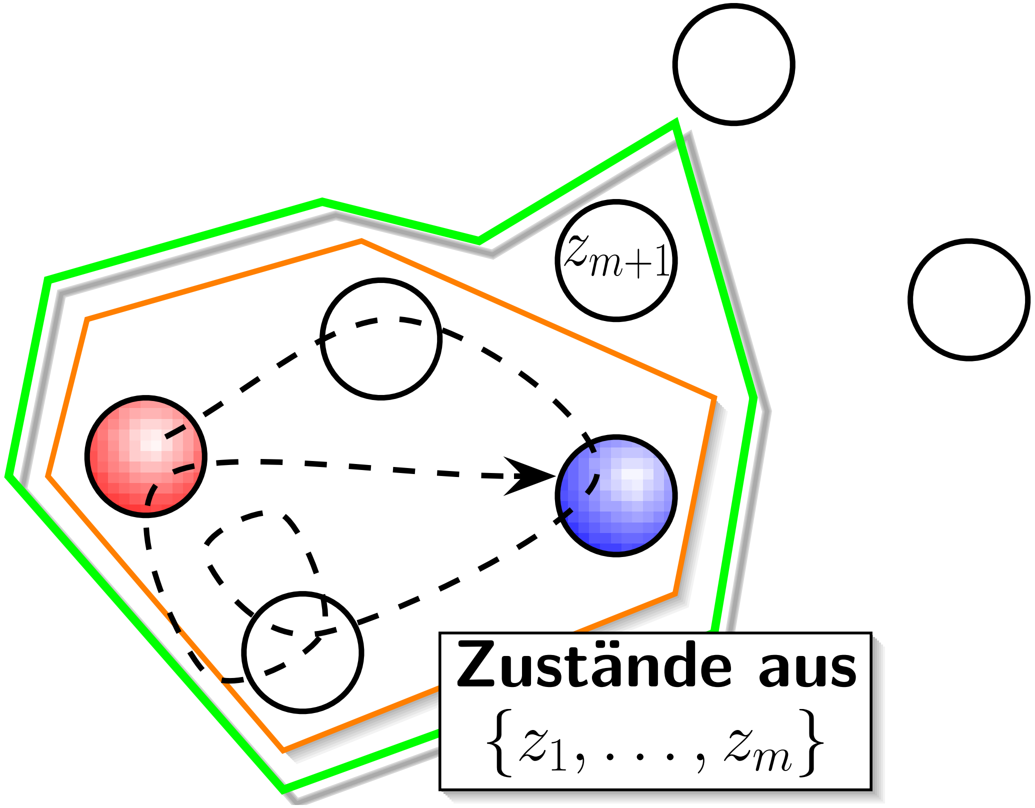
Pfade in einem FA



Pfade in einem FA

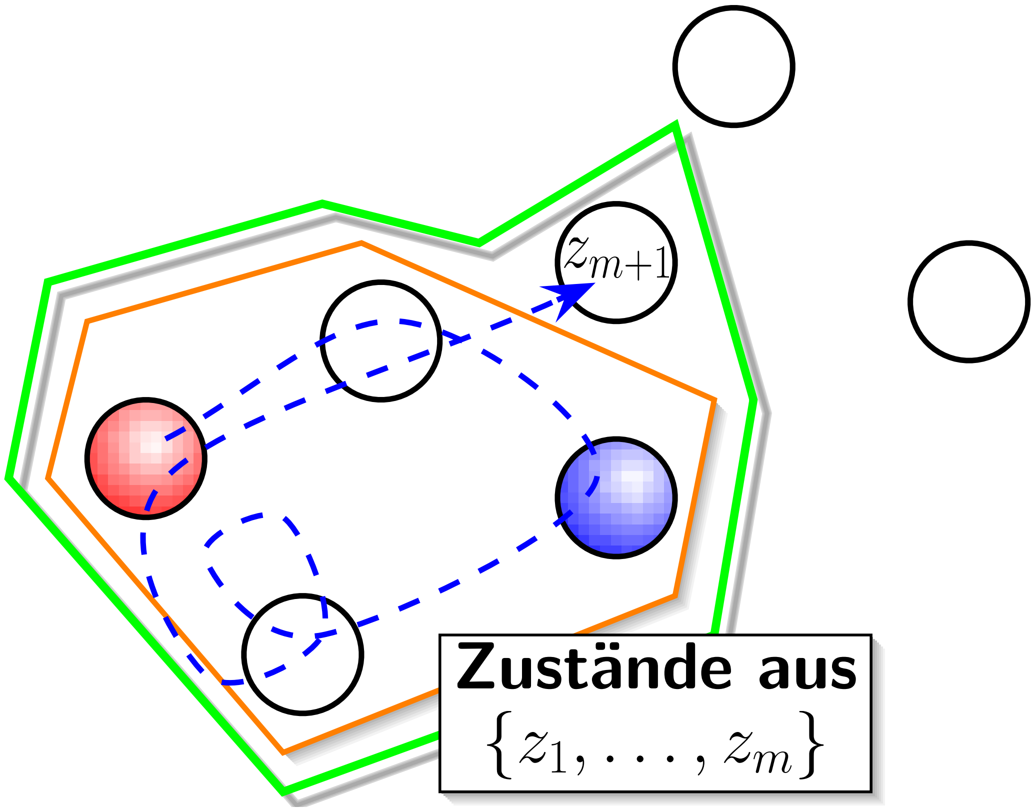


Pfade in einem FA



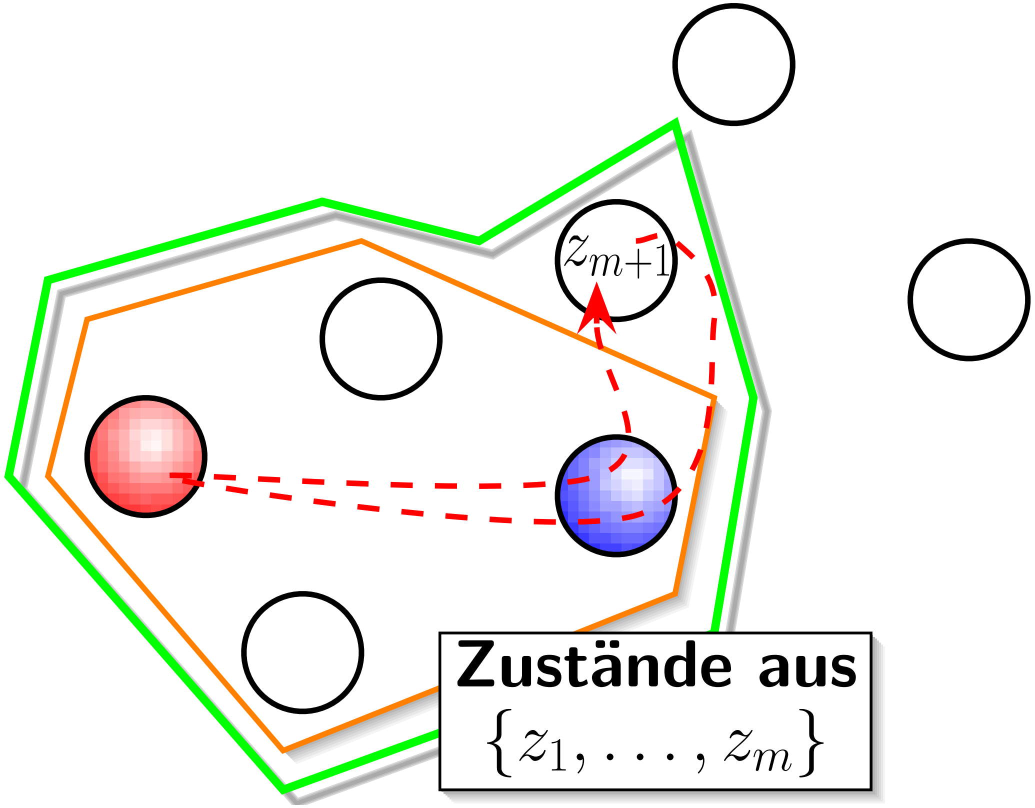
$R_{i,j}^m$

Pfade in einem FA



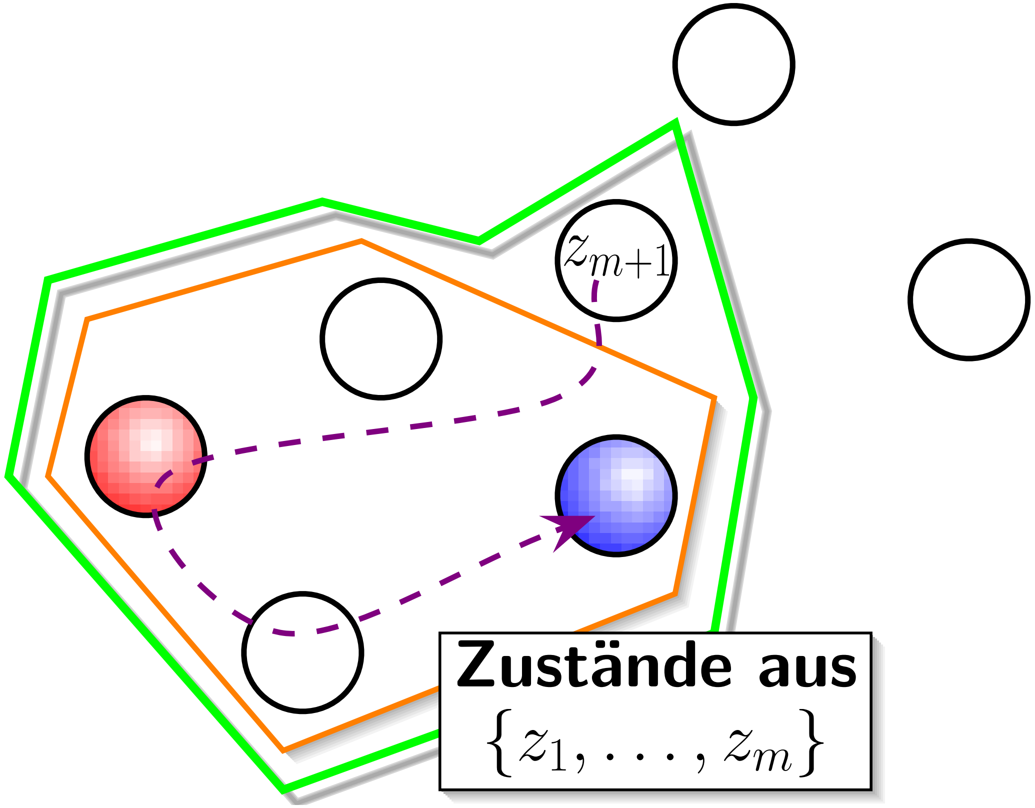
$R^m_{i,m+1}$

Pfade in einem FA



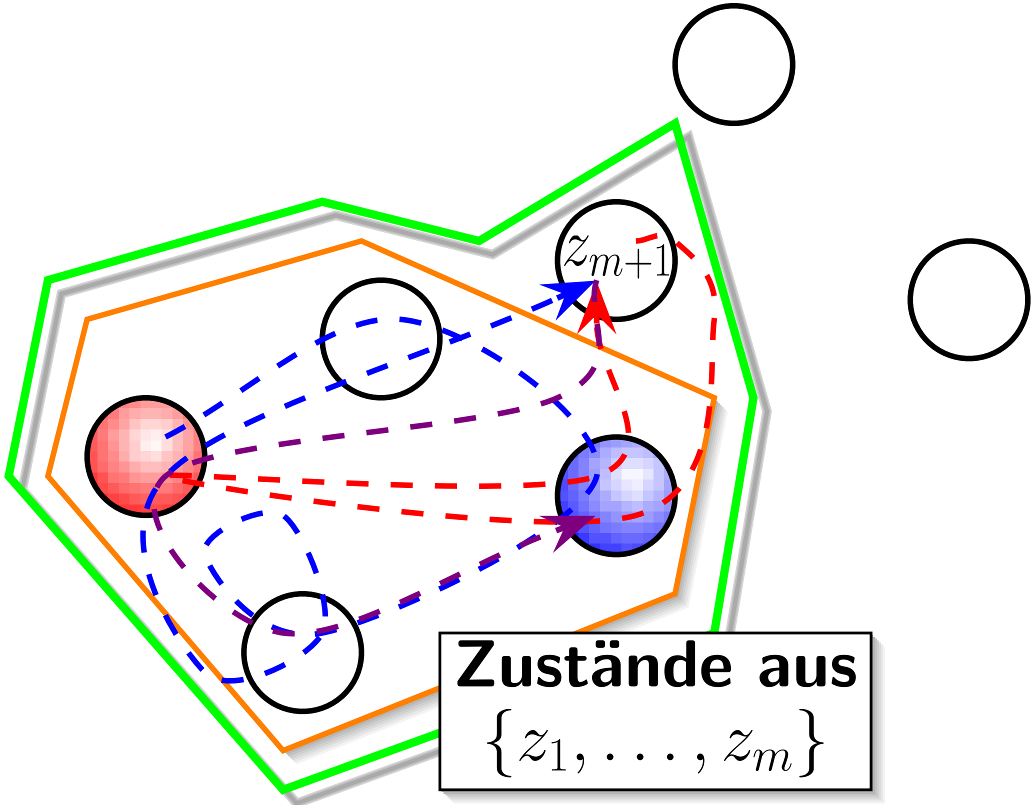
$$\left(R^m_{m+1,m+1}\right)^*$$

Pfade in einem FA



$R^m_{m+1,j}$

Pfade in einem FA



$$R^m_{i,j} \cup R^m_{i,m+1} \cdot (R^m_{m+1,m+1})^* \cdot R^m_{m+1,j}$$

Beweis (Forts.)

- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{z_1, \ldots, z_m\}$.

Beweis (Forts.)

- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{z_1, \ldots, z_m\}$.
- Offensichtlich gilt:

Beweis (Forts.)

- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{z_1, \ldots, z_m\}$.
- Offensichtlich gilt:
  - $u \in R^m_{i,m+1}$,

Beweis (Forts.)

- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{z_1, \ldots, z_m\}$.
- Offensichtlich gilt:
  - $u \in R^m_{i,m+1}$,
  - $\forall 1 \leq i \leq r: \ v_i \in R^m_{m+1,m+1}$ und

Beweis (Forts.)

- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{z_1, \ldots, z_m\}$.
- Offensichtlich gilt:
  - $u \in R^m_{i,m+1}$,
  - $\forall 1 \leq i \leq r: \; v_i \in R^m_{m+1,m+1}$ und
  - $w \in R^m_{m+1,j}$.

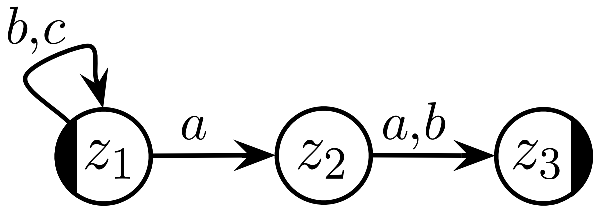
Beweis (Forts.)

- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{z_1, \ldots, z_m\}$.
- Offensichtlich gilt:
  - $u \in R^m_{i,m+1}$,
  - $\forall 1 \le i \le r: \; v_i \in R^m_{m+1,m+1}$ und
  - $w \in R^m_{m+1,j}$.
- Dies zeigt zunächst die Inklusion: $R^{m+1}_{i,j} \subseteq R^m_{i,j} \cup R^m_{i,m+1} \cdot (R^m_{m+1,m+1})^* \cdot R^m_{m+1,j}$.

Beweis (Forts.)

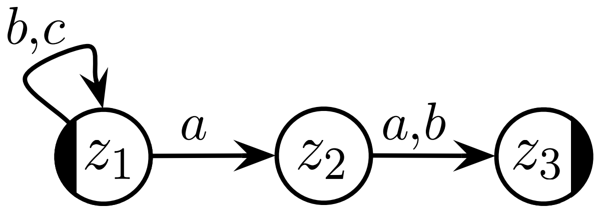
- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{z_1, \ldots, z_m\}$.
- Offensichtlich gilt:
  - $u \in R^m_{i,m+1}$,
  - $\forall 1 \leq i \leq r: \; v_i \in R^m_{m+1,m+1}$ und
  - $w \in R^m_{m+1,j}$.
- Dies zeigt zunächst die Inklusion:
  $R^{m+1}_{i,j} \subseteq R^m_{i,j} \cup R^m_{i,m+1} \cdot (R^m_{m+1,m+1})^* \cdot R^m_{m+1,j}$.
- Die Umkehrung dieser Inklusion ist leicht ersichtlich.

Beispiel: 1. ohne Zwischenzustand



$$R^0_{1,1} = \{\lambda, b, c\}$$

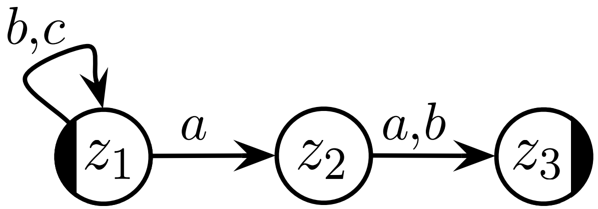
Beispiel: 1. ohne Zwischenzustand



$$R^0_{1,1} = \{\lambda, b, c\}$$

$$R^0_{1,2} = \{a\}$$

Beispiel: 1. ohne Zwischenzustand

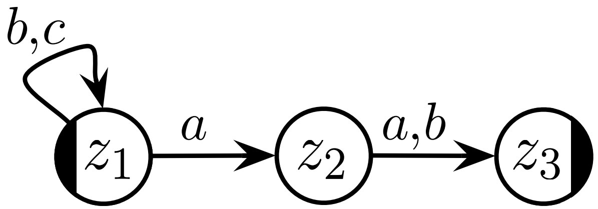


$$R^0_{1,1} = \{\lambda, b, c\}$$

$$R^0_{1,2} = \{a\}$$

$$R^0_{1,3} = R^0_{2,1} = R^0_{3,1} = R^0_{3,2} = \emptyset$$

Beispiel: 1. ohne Zwischenzustand



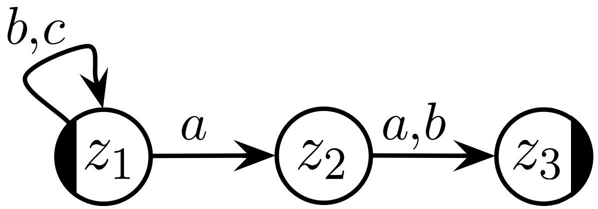
$$R^0_{1,1} = \{\lambda, b, c\}$$

$$R^0_{1,2} = \{a\}$$

$$R^0_{1,3} = R^0_{2,1} = R^0_{3,1} = R^0_{3,2} = \emptyset$$

$$R^0_{2,2} = \{\lambda\}$$

Beispiel: 1. ohne Zwischenzustand



$$R^0_{1,1} = \{\lambda, b, c\}$$

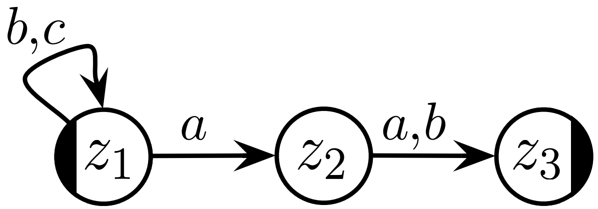
$$R^0_{1,2} = \{a\}$$

$$R^0_{1,3} = R^0_{2,1} = R^0_{3,1} = R^0_{3,2} = \emptyset$$

$$R^0_{2,2} = \{\lambda\}$$

$$R^0_{2,3} = \{a, b\}$$

Beispiel: 1. ohne Zwischenzustand



$$R^0_{1,1} = \{\lambda, b, c\}$$

$$R^0_{1,2} = \{a\}$$

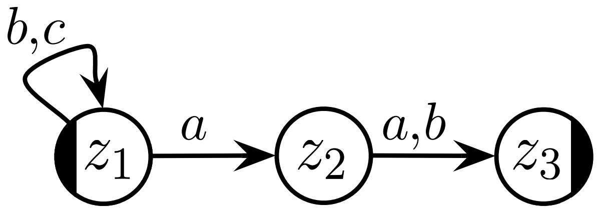
$$R^0_{1,3} = R^0_{2,1} = R^0_{3,1} = R^0_{3,2} = \emptyset$$

$$R^0_{2,2} = \{\lambda\}$$

$$R^0_{2,3} = \{a, b\}$$

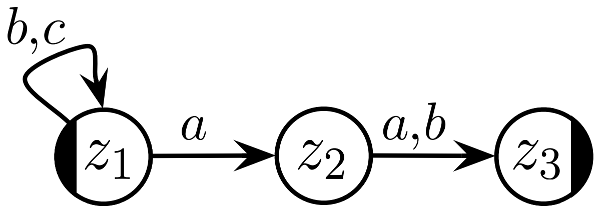
$$R^0_{3,3} = \{\lambda\}$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



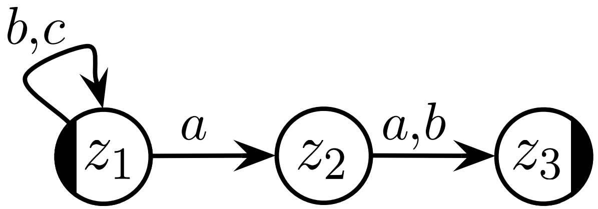
$$R^1_{1,1} \;=\; R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1}$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



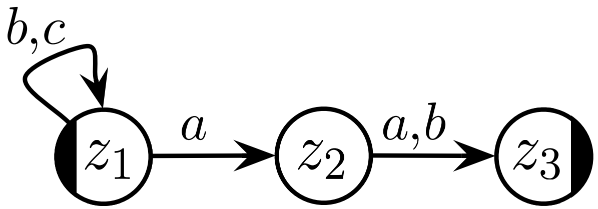
$$
\begin{aligned}
R^1_{1,1} &= R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1} \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^* \cdot R^0_{1,1})
\end{aligned}
$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



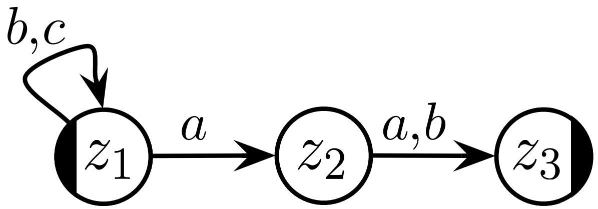
$$
\begin{aligned}
R^1_{1,1} &= R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1} \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^* \cdot R^0_{1,1}) \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^+)
\end{aligned}
$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



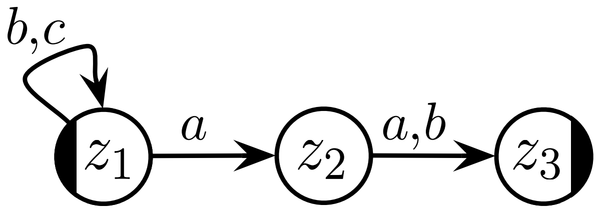
$$
\begin{aligned}
R^1_{1,1} &= R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1} \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^* \cdot R^0_{1,1}) \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^+) \\
&= R^0_{1,1} \cdot (R^0_{1,1})^*
\end{aligned}
$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



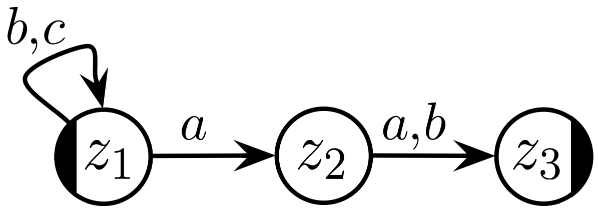
$$
\begin{aligned}
R^1_{1,1} &= R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1} \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^* \cdot R^0_{1,1}) \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^+) \\
&= R^0_{1,1} \cdot (R^0_{1,1})^* \\
&= (R^0_{1,1})^+
\end{aligned}
$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



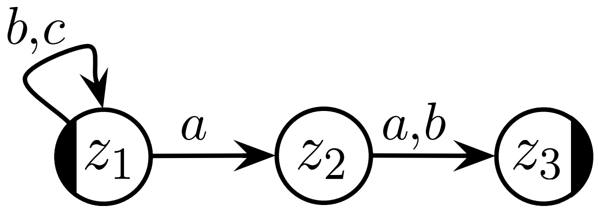
$$
\begin{aligned}
R^1_{1,1} &= R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1} \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^* \cdot R^0_{1,1}) \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^+) \\
&= R^0_{1,1} \cdot (R^0_{1,1})^* \\
&= (R^0_{1,1})^+ \\
&= \{\lambda, b, c\}^+
\end{aligned}
$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



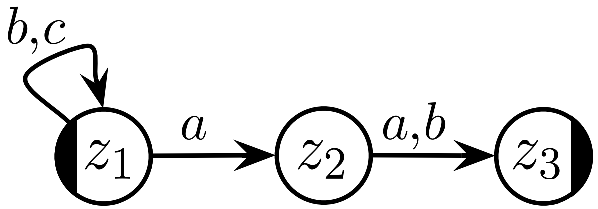
$$
\begin{aligned}
R^1_{1,1} &= R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1} \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^* \cdot R^0_{1,1}) \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^+) \\
&= R^0_{1,1} \cdot (R^0_{1,1})^* \\
&= (R^0_{1,1})^+ \\
&= \{\lambda, b, c\}^+ \\
&= \{b, c\}^*
\end{aligned}
$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



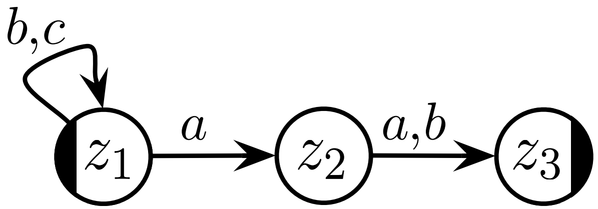
$$
\begin{aligned}
R^1_{1,1} &= R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1} \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^* \cdot R^0_{1,1}) \\
&= R^0_{1,1} \cdot (\{\lambda\} \cup (R^0_{1,1})^+) \\
&= R^0_{1,1} \cdot (R^0_{1,1})^* \\
&= (R^0_{1,1})^+ \\
&= \{\lambda, b, c\}^+ \\
&= \{b, c\}^* \\
&= (b + c)^*
\end{aligned}
$$

Beispiel: 2. Zwischenzustände $\{z_1\}$



$$
\begin{aligned}
R^1_{1,2} &= R^0_{1,2} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^{1,2}_0 \\
&= \{a\} \cup \{\lambda, b, c\} \cdot \{\lambda, b, c\}^* \cdot \{a\} \\
&= \{a\} \cup \{\lambda, b, c\}^+ \cdot \{a\} \\
&= \{b, c\}^* \cdot \{a\} \\
&= (b + c)^* a
\end{aligned}
$$

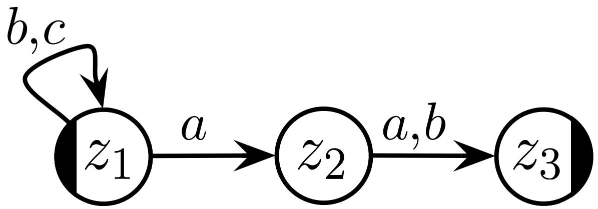
Beispiel: 2. Zwischenzustände $\{z_1\}$



$$
\begin{aligned}
R^1_{1,2} &= R^0_{1,2} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^{1,2}_0 \\
&= \{a\} \cup \{\lambda, b, c\} \cdot \{\lambda, b, c\}^* \cdot \{a\} \\
&= \{a\} \cup \{\lambda, b, c\}^+ \cdot \{a\} \\
&= \{b, c\}^* \cdot \{a\} \\
&= (b + c)^* a
\end{aligned}
$$

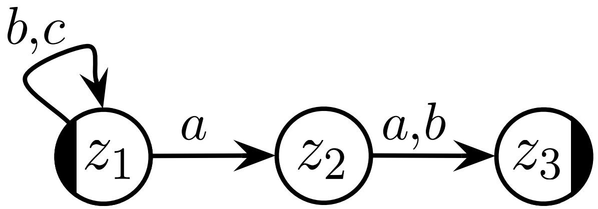
$$
\begin{aligned}
R^1_{2,3} &= R^0_{2,3} \cup R^0_{2,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,3} \\
&= (a + b)
\end{aligned}
$$

Beispiel: 3. alle Zwischenzustände



$$R^3_{1,3} \quad = \quad R^2_{1,3} \cup R^2_{1,3} \cdot (R^2_{3,3})^* \cdot R^2_{3,3}$$

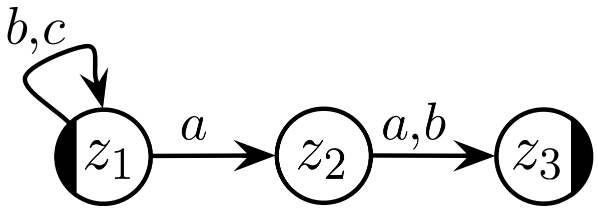
Beispiel: 3. alle Zwischenzustände



$$R^3_{1,3} = R^2_{1,3} \cup R^2_{1,3} \cdot (R^2_{3,3})^* \cdot R^2_{3,3}$$

$$\begin{aligned} R^2_{3,3} &= R^1_{3,3} \cup R^1_{3,2} \cdot (R^1_{2,2})^* \cdot R^1_{2,3} \\ &= R^1_{3,3} \cup \emptyset \cdot \quad \ldots \end{aligned}$$

Beispiel: 3. alle Zwischenzustände

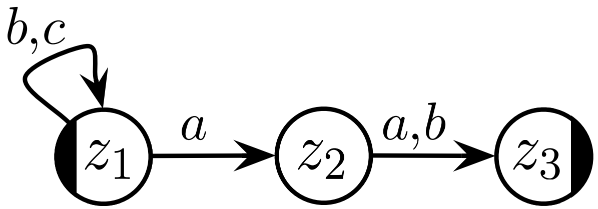


$$
\begin{aligned}
R^3_{1,3} &= R^2_{1,3} \cup R^2_{1,3} \cdot (R^2_{3,3})^* \cdot R^2_{3,3}
\end{aligned}
$$

$$
\begin{aligned}
R^2_{3,3} &= R^1_{3,3} \cup R^1_{3,2} \cdot (R^1_{2,2})^* \cdot R^1_{2,3} \\
&= R^1_{3,3} \cup \emptyset \cdot \quad \cdots
\end{aligned}
$$

$$
\begin{aligned}
R^1_{3,3} &= R^0_{3,3} \cup R^0_{3,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,3} \\
&= \{\lambda\} \cup \emptyset \cdot \quad \cdots \\
&= \{\lambda\}
\end{aligned}
$$

Beispiel: 3. alle Zwischenzustände

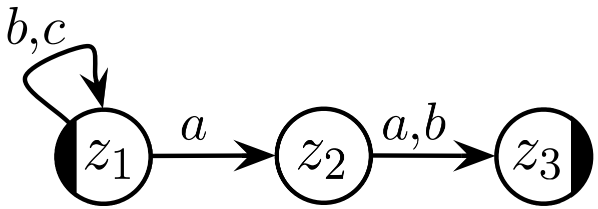


$$
\begin{aligned}
R^3_{1,3} &= R^2_{1,3} \cup R^2_{1,3} \cdot (R^2_{3,3})^* \cdot R^2_{3,3} \\
&= R^2_{1,3} \cup R^2_{1,3} \cdot \{\lambda\}^* \cdot \{\lambda\} \\
&= R^2_{1,3}
\end{aligned}
$$

$$
\begin{aligned}
R^2_{3,3} &= R^1_{3,3} \cup R^1_{3,2} \cdot (R^1_{2,2})^* \cdot R^1_{2,3} \\
&= R^1_{3,3} \cup \emptyset \cdot \quad \cdots
\end{aligned}
$$

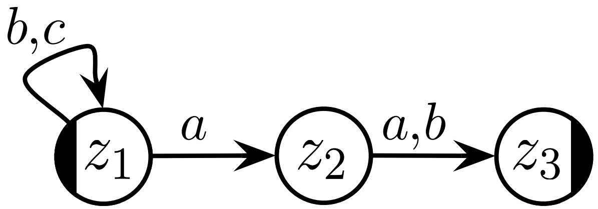
$$
\begin{aligned}
R^1_{3,3} &= R^1_{3,3} \cup R^0_{3,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,3} \\
&= \{\lambda\} \cup \emptyset \cdot \quad \cdots \\
&= \{\lambda\}
\end{aligned}
$$

Beispiel: 3. alle Zwischenzustände



Also $L(A) = R^3_{1,3} = R^2_{1,3}$ mit

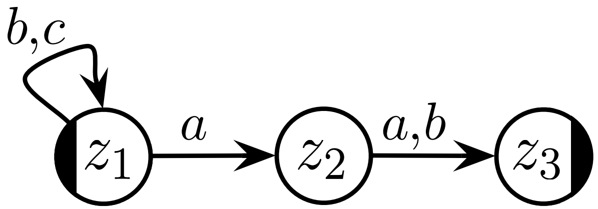
Beispiel: 3. alle Zwischenzustände



Also $L(A) = R^3_{1,3} = R^2_{1,3}$ mit

$$\begin{aligned} R^2_{1,3} &= R^1_{1,3} \cup R^1_{1,2} \cdot (R^1_{2,2})^* \cdot R^1_{2,3} \\ &= \emptyset + (b+c)^*(a)(a+b) \\ &= (b+c)^*(a)(a+b) \\ &= L(A) \end{aligned}$$

Beispiel: 3. alle Zwischenzustände



Also $L(A) = R^3_{1,3} = R^2_{1,3}$ mit

$$\begin{aligned} R^2_{1,3} &= R^1_{1,3} \cup R^1_{1,2} \cdot (R^1_{2,2})^* \cdot R^1_{2,3} \\ &= \emptyset + (b+c)^*(a)(a+b) \\ &= (b+c)^*(a)(a+b) \\ &= L(A) \end{aligned}$$

…hier hätte man den rationalen Ausdruck auch direkt vom Automaten „ablesen“ können!