F2 — Automaten und formale Sprachen

Berndt Farwer

Fachbereich Informatik

AB „Theoretische Grundlagen der Informatik“ (TGI)

Universität Hamburg

farwer@informatik.uni-hamburg.de

Moore-Automat

- Ein **Moore-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$, mit

Moore-Automat

- Ein **Moore-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$, mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,

Moore-Automat

- Ein **Moore-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$, mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,
  - Γ ist **Ausgabealphabet** und

Moore-Automat

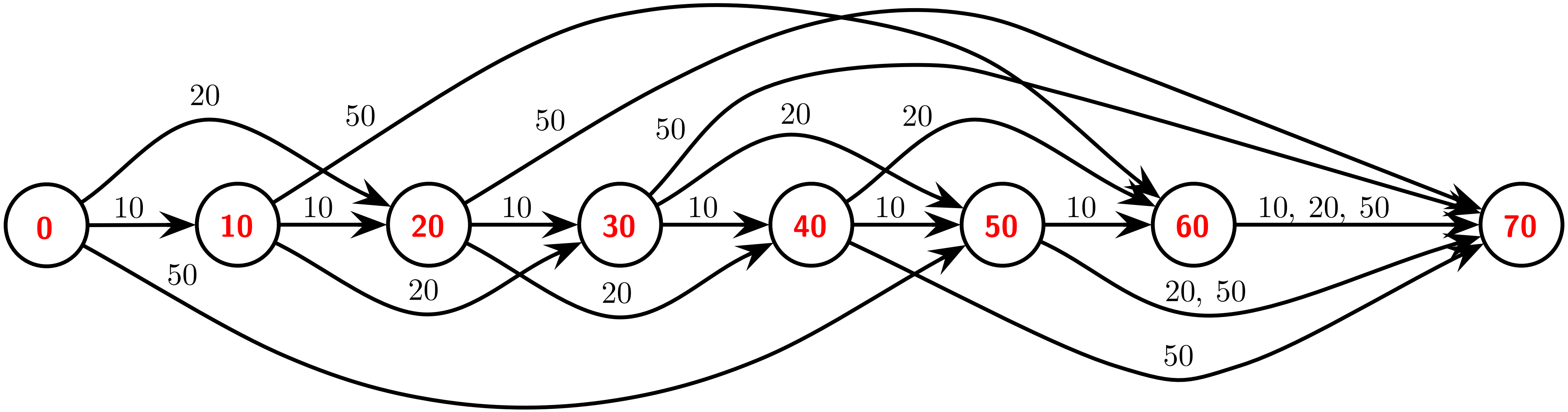
- Ein **Moore-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$, mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,
  - Γ ist **Ausgabealphabet** und
  - $\rho : Z \longrightarrow \Gamma$ ist **Ausgabefunktion**.

Moore-Automat

- Ein **Moore-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$, mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,
  - Γ ist **Ausgabealphabet** und
  - $\rho : Z \longrightarrow \Gamma$ ist **Ausgabefunktion**.
- Die **Ausgabe** von M für die Eingabe $w := x_1 x_2 \ldots x_n$, mit $x_i \in \Sigma$, ist: $\rho(z_0)\rho(z_1)\ldots\rho(z_n)$ wenn für $z_0, z_1, \ldots, z_n$ gilt: $\delta(z_i, x_{i+1}) = z_{i+1}, 0 \leq i \leq n$.

  $(z_0, x_1, z_1), (z_1, x_2, z_2), \ldots, (z_{n-1}, x_n, z_n)$ ist also eine **Rechnung** von A für die Eingabe $x_1 x_2 \ldots x_n$.

Beispiel: Getränkeautomat



- Ausgabe des bisher gezahlten Betrags im Zustand.
- Es gibt keine Ausgaben an den Kanten.

Beispiel: Moore-Automat

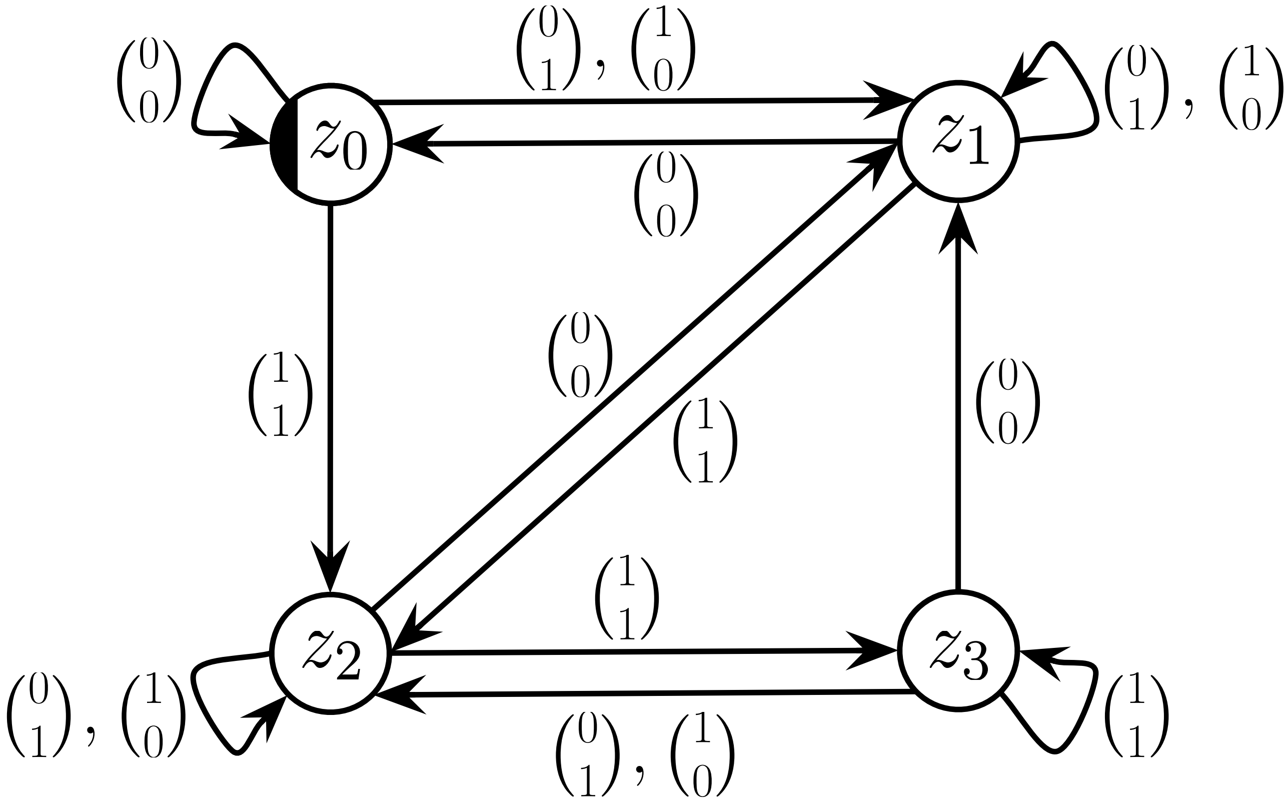
Ein Serienaddierer . . . (von hinten)

$$11101 + 1011 = 101000$$

Beispiel: Moore-Automat

Ein Serienaddierer . . . (von hinten)

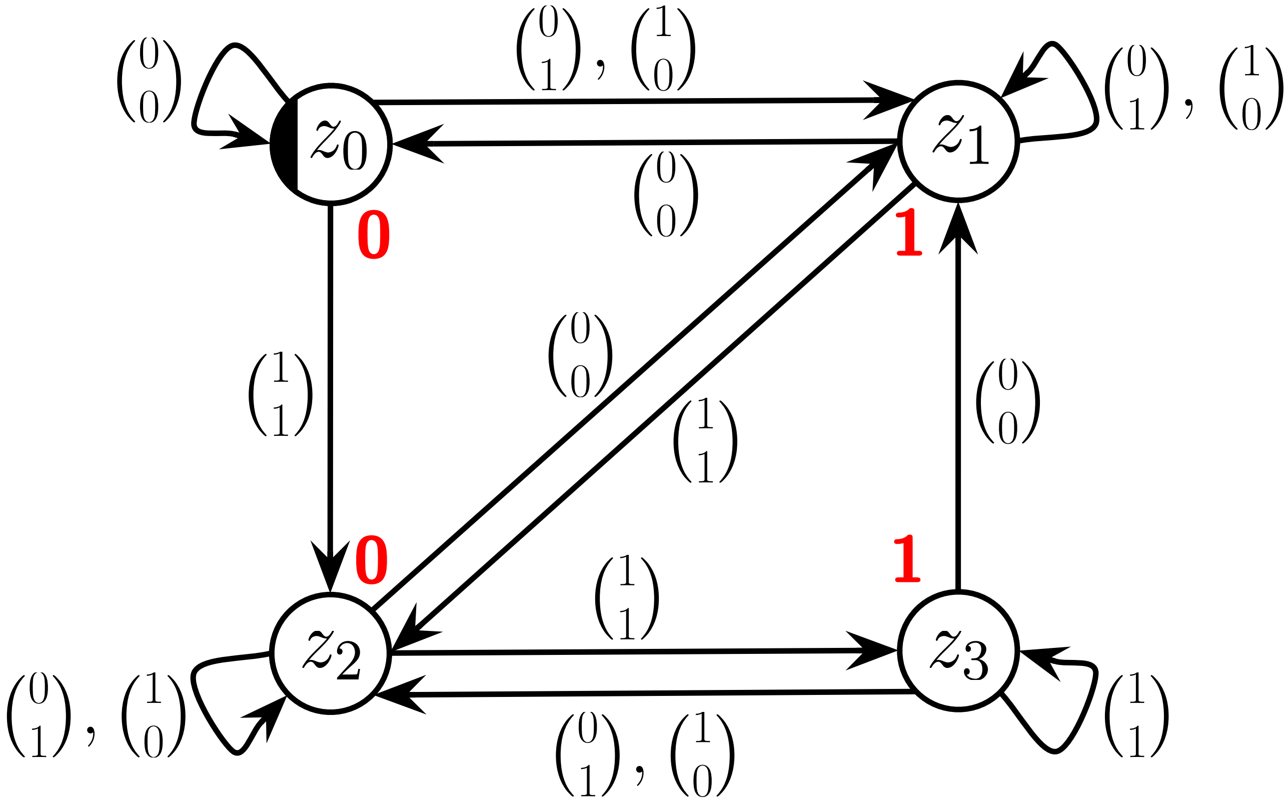
$$11101 + 1011 = 101000$$



Beispiel: Moore-Automat

Ein Serienaddierer . . . (von hinten)

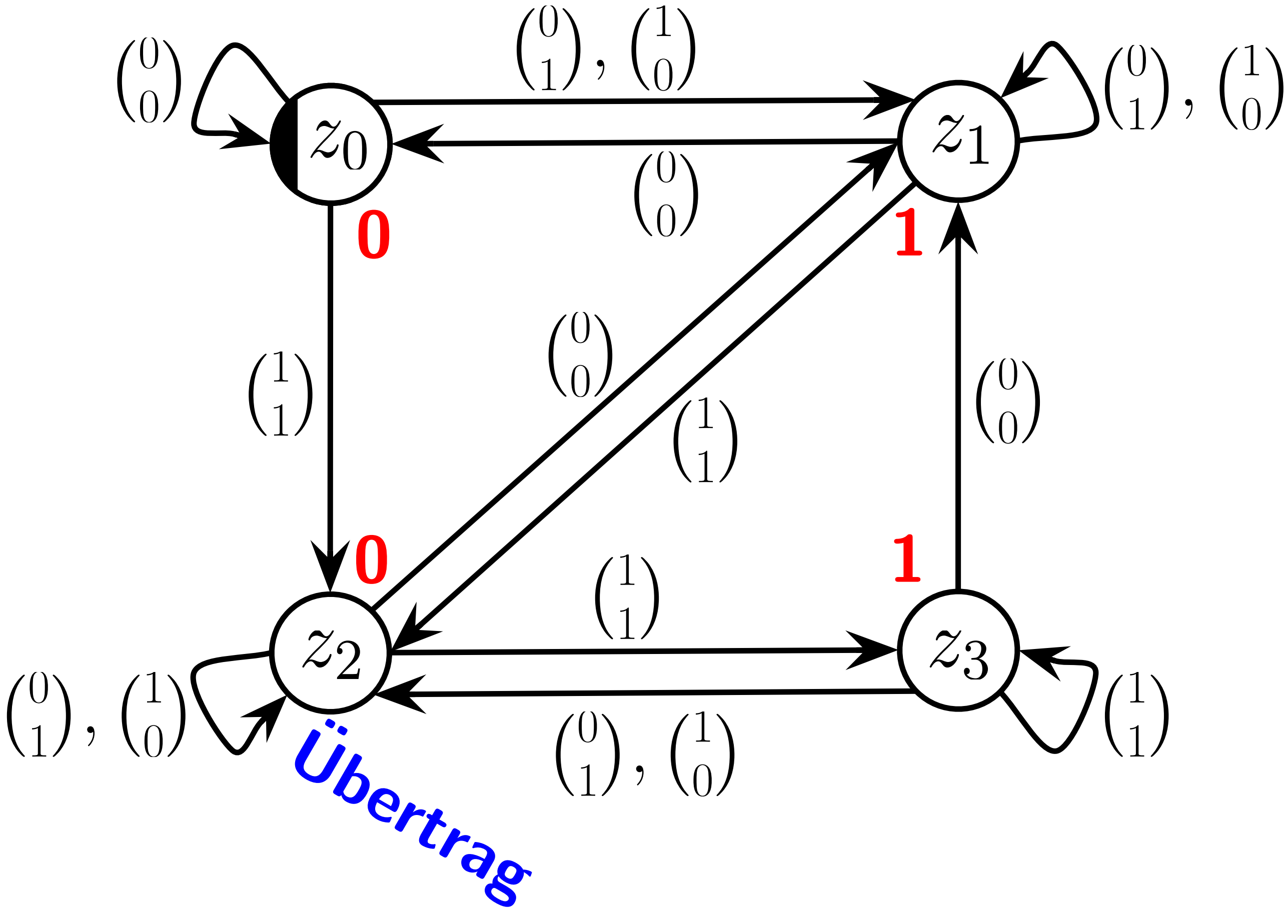
$$11101 + 1011 = 101000$$



Beispiel: Moore-Automat

Ein Serienaddierer . . . (von hinten)

$$11101 + 1011 = 101000$$



Mealy-Automat

- Ein **Mealy-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$ mit

Mealy-Automat

- Ein **Mealy-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$ mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,

Mealy-Automat

- Ein **Mealy-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$ mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,
  - Γ ist **Ausgabealphabet** und

Mealy-Automat

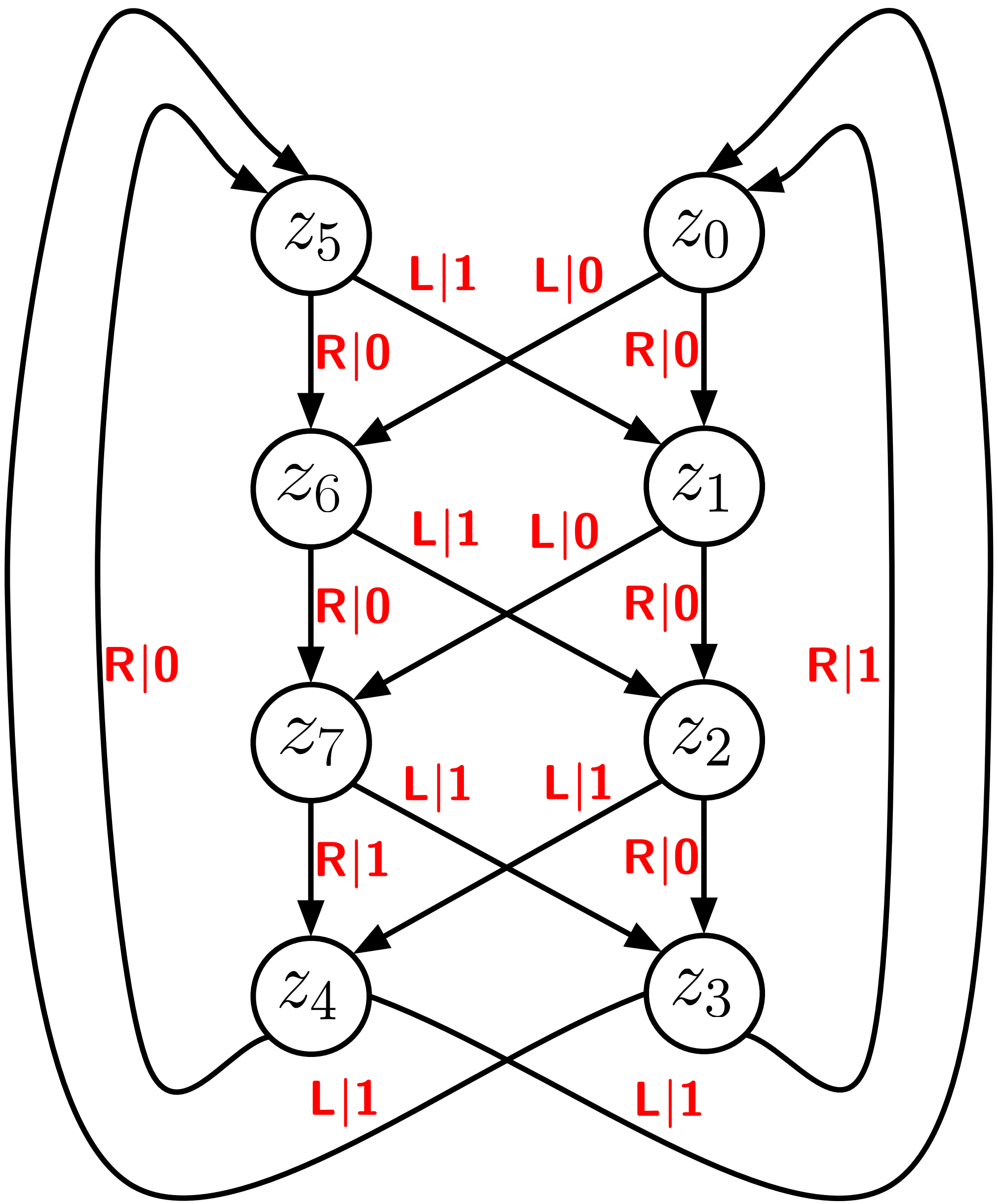
- Ein **Mealy-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$ mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,
  - Γ ist **Ausgabealphabet** und
  - $\rho : Z \times \Sigma \longrightarrow \Gamma$ **Ausgabefunktion**.

Mealy-Automat

- Ein **Mealy-Automat** wird beschrieben durch $M = (Z, \Sigma, \Gamma, \delta, \rho, z_0)$ mit
  - $A := (Z, \Sigma, \delta, z_0, Z)$ ist vollständiger DFA,
  - Γ ist **Ausgabealphabet** und
  - $\rho : Z \times \Sigma \longrightarrow \Gamma$ **Ausgabefunktion**.
- Die **Ausgabe** von M für die Eingabe $w := x_1 \ldots x_n$, mit $x_i \in \Sigma$, ist: $\rho(z_0, x_1)\rho(z_1, x_2) \ldots \rho(z_{n-1}, x_n)$ wenn für $z_0, z_1, \ldots, z_{n-1}$ gilt:
  $\delta(z_i, x_{i+1}) = z_{i+1}, 0 \leq i < n - 1$

  $(z_0, x_1, z_1), (z_1, x_2, z_2), \ldots, (z_{n-1}, x_n, z_n)$ ist also eine **Rechnung** von A für die Eingabe $x_1 x_2 \ldots x_n$.

Beispiel: Kugelautomat



Beispiel: Mealy-Automat

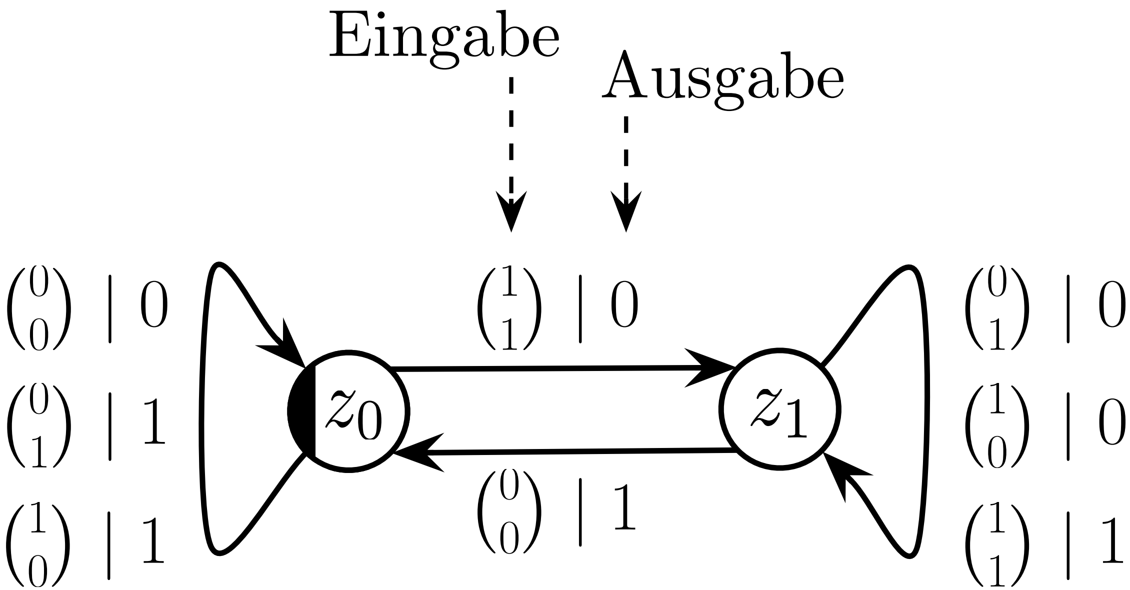
Ein Serienaddierer . . .

$$11101 + 1011 = 101000$$

Beispiel: Mealy-Automat

Ein Serienaddierer . . .

$$11101 + 1011 = 101000$$



Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:

Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:
  - DFA und NFA zur Akzeptierung von Sprachen

Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:
  - DFA und NFA zur Akzeptierung von Sprachen
  - eine Sprache, die von keinem NFA akzeptiert wird

Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:
  - DFA und NFA zur Akzeptierung von Sprachen
  - eine Sprache, die von keinem NFA akzeptiert wird
- Wir benötigen:

Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:
  - DFA und NFA zur Akzeptierung von Sprachen
  - eine Sprache, die von keinem NFA akzeptiert wird
- Wir benötigen:
  - mehr Informationen über die von NFA und DFA akzeptierten Sprachen

Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:
  - DFA und NFA zur Akzeptierung von Sprachen
  - eine Sprache, die von keinem NFA akzeptiert wird
- Wir benötigen:
  - mehr Informationen über die von NFA und DFA akzeptierten Sprachen
    - Was unterscheidet die von NFAs und DFAs akzeptierten Sprachen?

Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:
  - DFA und NFA zur Akzeptierung von Sprachen
  - eine Sprache, die von keinem NFA akzeptiert wird
- Wir benötigen:
  - mehr Informationen über die von NFA und DFA akzeptierten Sprachen
    - Was unterscheidet die von NFAs und DFAs akzeptierten Sprachen?
    - Abschlußeigenschaften

Von FAs akzeptierte Sprachen

- Wir haben kennengelernt:
  - DFA und NFA zur Akzeptierung von Sprachen
  - eine Sprache, die von keinem NFA akzeptiert wird
- Wir benötigen:
  - mehr Informationen über die von NFA und DFA akzeptierten Sprachen
    - Was unterscheidet die von NFAs und DFAs akzeptierten Sprachen?
    - Abschlußeigenschaften
    - Entscheidbarkeitsresultate

Sprachfamilien

- Eine Klasse $\mathcal{L}$ von formalen Sprachen wird **Sprachfamilie** genannt, wenn folgende Bedingungen erfüllt sind:

Sprachfamilien

- Eine Klasse $\mathcal{L}$ von formalen Sprachen wird **Sprachfamilie** genannt, wenn folgende Bedingungen erfüllt sind:

  1. $\mathcal{L} \neq \emptyset$.

Sprachfamilien

- Eine Klasse $\mathcal{L}$ von formalen Sprachen wird **Sprachfamilie** genannt, wenn folgende Bedingungen erfüllt sind:
  1. $\mathcal{L} \neq \emptyset$.
  2. Es existiert ein *abzählbar unendliches* Alphabet Γ, so dass für jede Sprache $L \in \mathcal{L}$ ein *endliches* Alphabet $\Sigma \subseteq \Gamma$ existiert mit $L \subseteq \Sigma^*$.

Sprachfamilien

- Eine Klasse $\mathcal{L}$ von formalen Sprachen wird **Sprachfamilie** genannt, wenn folgende Bedingungen erfüllt sind:
  1. $\mathcal{L} \neq \emptyset$.
  2. Es existiert ein *abzählbar unendliches* Alphabet Γ, so dass für jede Sprache $L \in \mathcal{L}$ ein *endliches* Alphabet $\Sigma \subseteq \Gamma$ existiert mit $L \subseteq \Sigma^*$.
  3. $\exists L \in \mathcal{L} : L \neq \emptyset$.

Sprachfamilien

- Eine Klasse $\mathcal{L}$ von formalen Sprachen wird **Sprachfamilie** genannt, wenn folgende Bedingungen erfüllt sind:
  1. $\mathcal{L} \neq \emptyset$.
  2. Es existiert ein *abzählbar unendliches* Alphabet Γ, so dass für jede Sprache $L \in \mathcal{L}$ ein *endliches* Alphabet $\Sigma \subseteq \Gamma$ existiert mit $L \subseteq \Sigma^*$.
  3. $\exists L \in \mathcal{L} : L \neq \emptyset$.

Die Klasse *aller* Wortmengen ist **keine** Sprachfamilie, da es für diese unendlich vielen Sprachen kein gemeinsames endliches Alphabet Σ gibt.

Die regulären Sprachen

- Die Mengen von Wörtern, die von einem DFA akzeptiert werden können, heißen **reguläre Mengen**.

Die regulären Sprachen

- Die Mengen von Wörtern, die von einem DFA akzeptiert werden können, heißen **reguläre Mengen**.
- Die Familie der regulären Mengen wird mit $\mathcal{R}eg$ bezeichnet.

Die regulären Sprachen

- Die Mengen von Wörtern, die von einem DFA akzeptiert werden können, heißen **reguläre Mengen**.
- Die Familie der regulären Mengen wird mit $\mathcal{R}eg$ bezeichnet.
- Mit $\mathcal{A}kz(\Sigma)$ wird die Familie aller Sprachen $L \subseteq \Sigma^*$ bezeichnet, die von DFAs mit Eingabe-Alphabet Σ akzeptiert werden können.

Die regulären Sprachen

- Die Mengen von Wörtern, die von einem DFA akzeptiert werden können, heißen **reguläre Mengen**.
- Die Familie der regulären Mengen wird mit $\mathcal{R}eg$ bezeichnet.
- Mit $\mathcal{A}kz(\Sigma)$ wird die Familie aller Sprachen $L \subseteq \Sigma^*$ bezeichnet, die von DFAs mit Eingabe-Alphabet Σ akzeptiert werden können.
- Es gilt:

  $\mathcal{A}kz(\Sigma) = \{L \subseteq \Sigma^* \mid L = L(A)$ für einen DFA $A\}$

  und

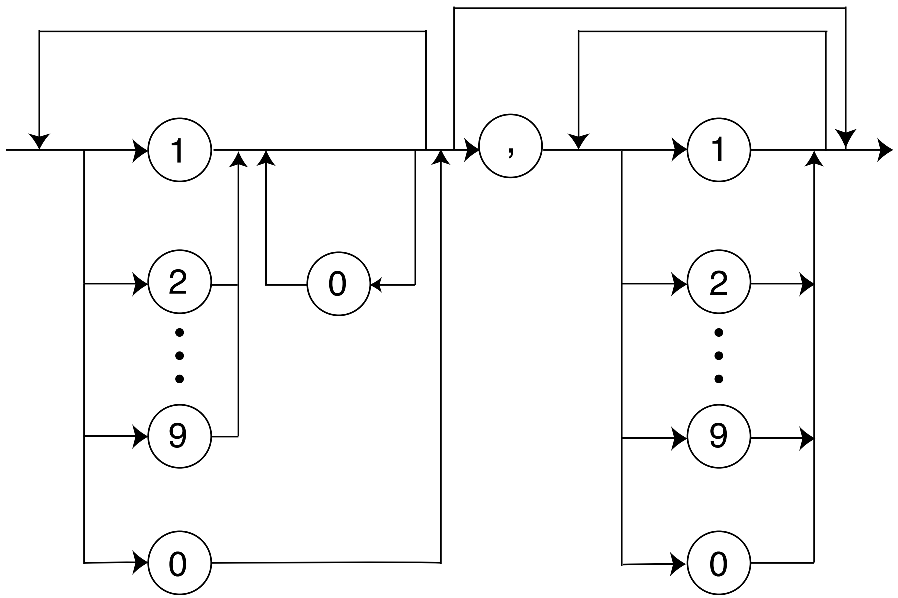
  $$\mathcal{R}eg = \bigcup_{\substack{\Sigma \text{ ist endl.} \\ \text{Alphabet}}} \mathcal{A}kz(\Sigma).$$

Wozu NFAs?

Einige Gründe für die Betrachtung von nichtdeteministischen endlichen Automaten:

- Analogie zu Syntaxdiagrammen:

Dezimalzahl:

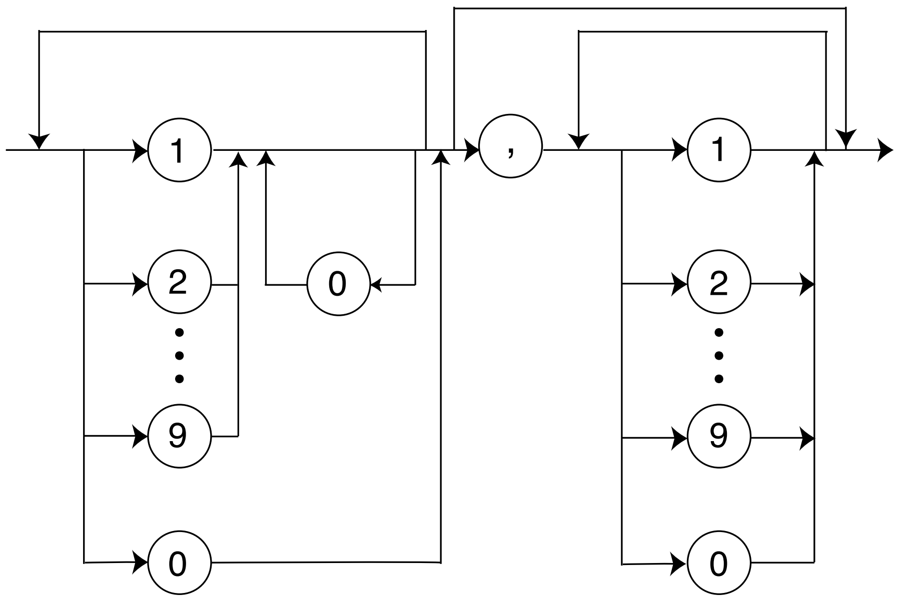


Wozu NFAs?

Einige Gründe für die Betrachtung von nichtdeteministischen endlichen Automaten:

- Analogie zu Syntaxdiagrammen:

Dezimalzahl:



- kleinere und häufig übersichtlichere Darstellung gegenüber DFAs

NFA = DFA

- **Theorem:** Jede von einem NFA akzeptierte Menge kann auch von einem initial zusammenhängenden, vollständigen DFA akzeptiert werden und ist daher regulär.

NFA = DFA

- **Theorem:** Jede von einem NFA akzeptierte Menge kann auch von einem initial zusammenhängenden, vollständigen DFA akzeptiert werden und ist daher regulär.
- Für den Beweis benötigen wir einige Umformungen des NFA, von dem wir ausgehen.

NFA = DFA

- **Theorem:** Jede von einem NFA akzeptierte Menge kann auch von einem initial zusammenhängenden, vollständigen DFA akzeptiert werden und ist daher regulär.
- Für den Beweis benötigen wir einige Umformungen des NFA, von dem wir ausgehen.
- Konstruktion in drei Schritten:

NFA = DFA

- **Theorem:** Jede von einem NFA akzeptierte Menge kann auch von einem initial zusammenhängenden, vollständigen DFA akzeptiert werden und ist daher regulär.
- Für den Beweis benötigen wir einige Umformungen des NFA, von dem wir ausgehen.
- Konstruktion in drei Schritten:
  - λ-frei

NFA = DFA

- **Theorem:** Jede von einem NFA akzeptierte Menge kann auch von einem initial zusammenhängenden, vollständigen DFA akzeptiert werden und ist daher regulär.
- Für den Beweis benötigen wir einige Umformungen des NFA, von dem wir ausgehen.
- Konstruktion in drei Schritten:
  - λ-frei
  - buchstabierend

NFA = DFA

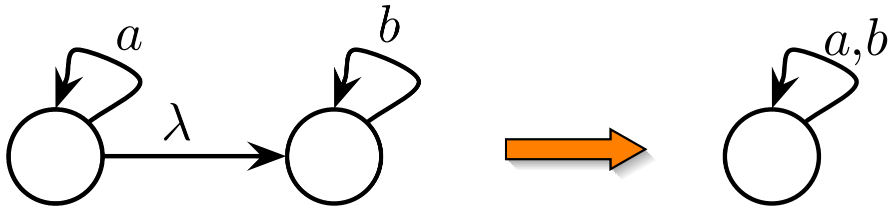
- **Theorem:** Jede von einem NFA akzeptierte Menge kann auch von einem initial zusammenhängenden, vollständigen DFA akzeptiert werden und ist daher regulär.
- Für den Beweis benötigen wir einige Umformungen des NFA, von dem wir ausgehen.
- Konstruktion in drei Schritten:
  - λ-frei
  - buchstabierend
  - vollständig und deterministisch

λ-frei

- Der **naive Ansatz**, mit λ-Kanten verbundene Zustände zu verschmelzen, führt nicht zum Ziel.

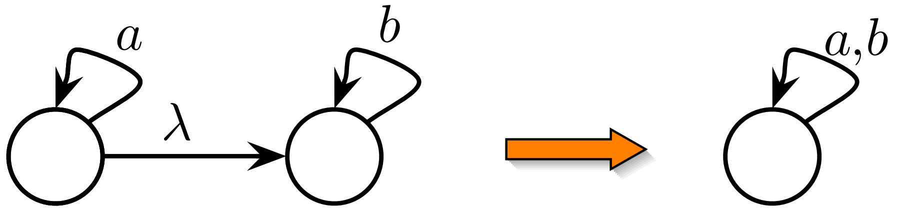
λ-frei

- Der **naive Ansatz**, mit λ-Kanten verbundene Zustände zu verschmelzen, führt nicht zum Ziel.
- Gegenbeispiel:



λ-frei

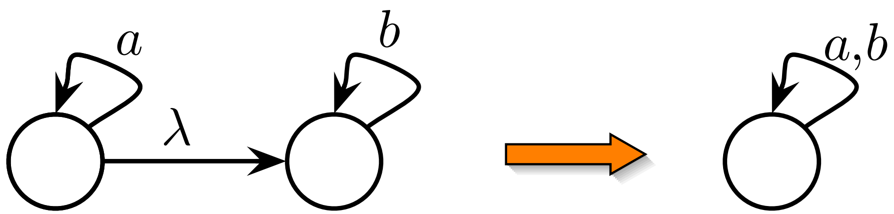
- Der **naive Ansatz**, mit λ-Kanten verbundene Zustände zu verschmelzen, führt nicht zum Ziel.
- Gegenbeispiel:



- Wir betrachten deshalb zunächst alle Nicht-λ-Kanten . . .

λ-frei

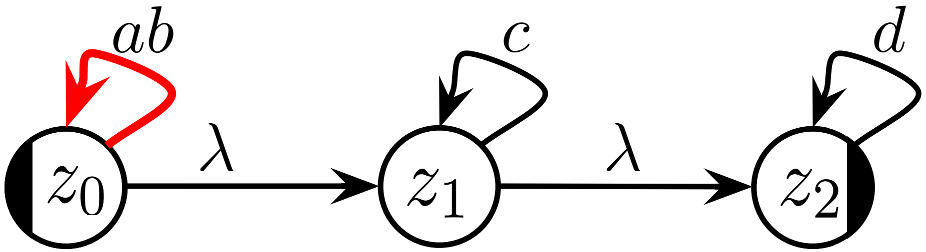
- Der **naive Ansatz**, mit λ-Kanten verbundene Zustände zu verschmelzen, führt nicht zum Ziel.
- Gegenbeispiel:



- Wir betrachten deshalb zunächst alle Nicht-λ-Kanten ...
- und bilden alle möglichen Kombinationen mit λ-Pfaden des Automaten.

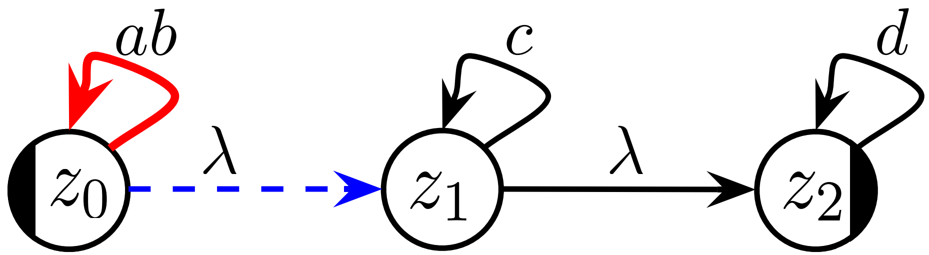
λ-Elimination

1. Betrachten wir die Kante (z_0, ab, z_0):



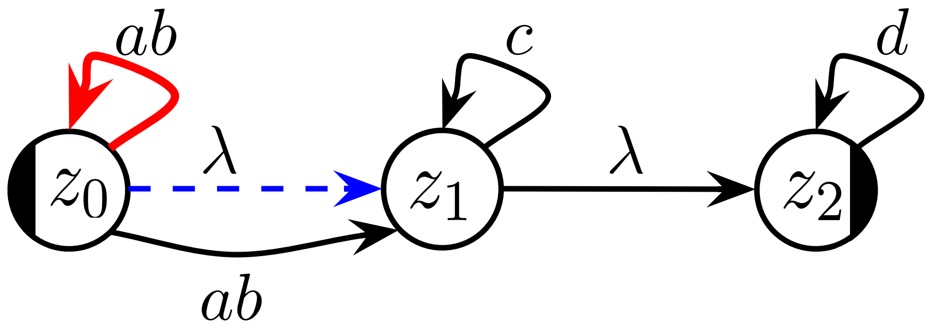
λ-Elimination

1. Betrachten wir die Kante (z_0, ab, z_0):



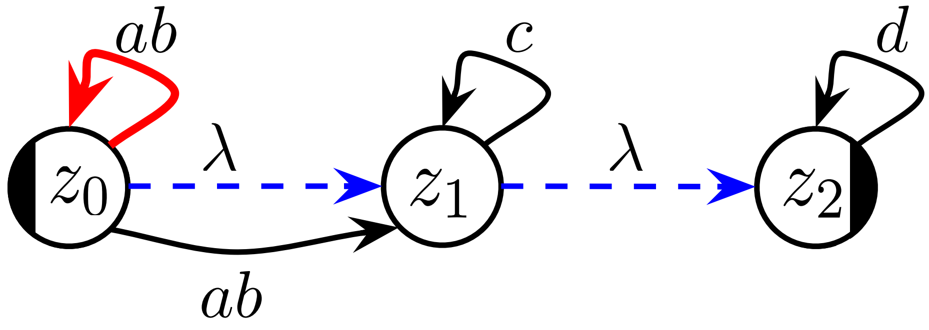
λ-Elimination

1. Betrachten wir die Kante (z_0, ab, z_0):



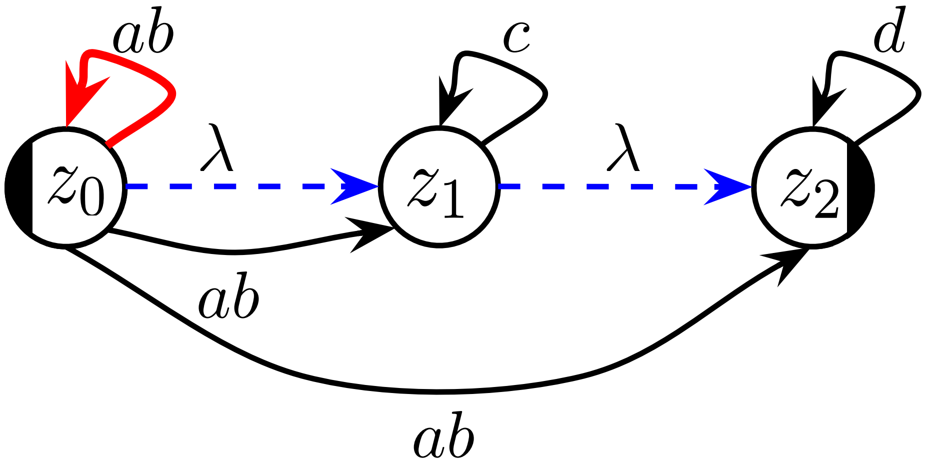
λ-Elimination

1. Betrachten wir die Kante (z_0, ab, z_0):



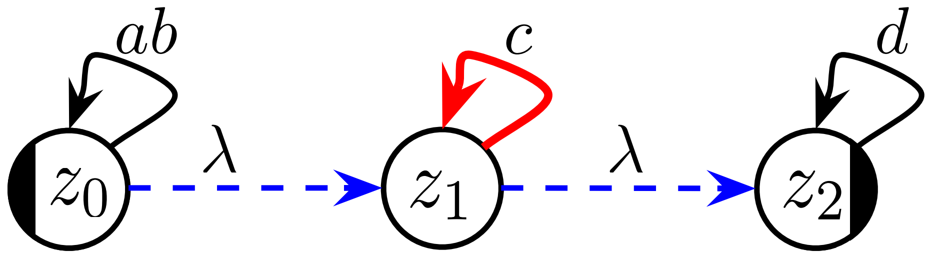
λ-Elimination

1. Betrachten wir die Kante (z_0, ab, z_0):



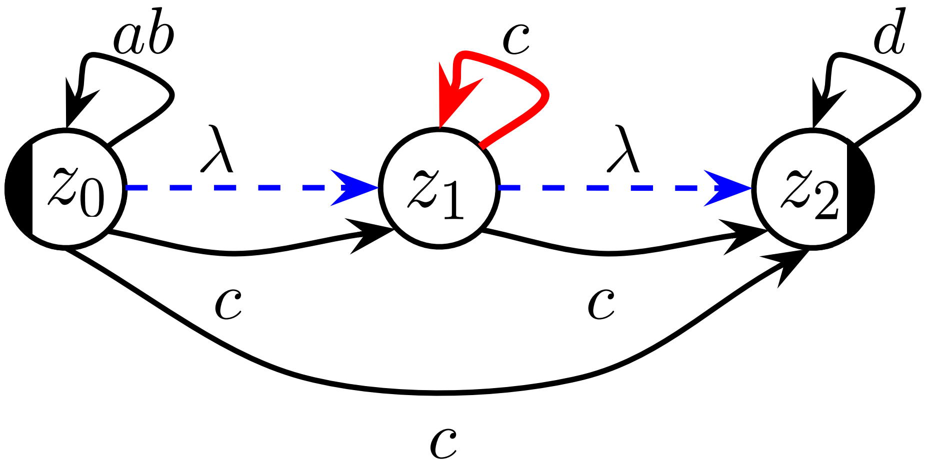
λ-Elimination

2. Betrachten wir die Kante (z_1, c, z_1):



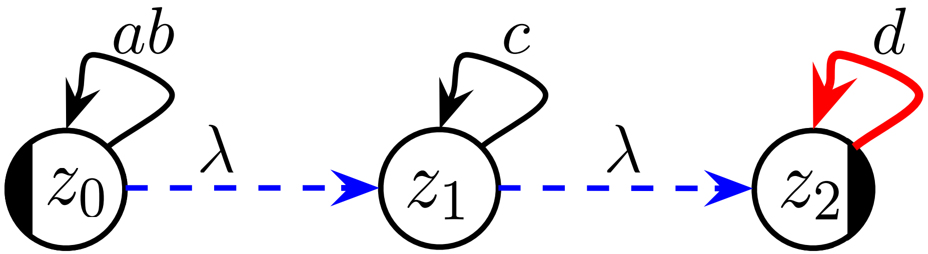
λ-Elimination

2. Betrachten wir die Kante (z_1, c, z_1):



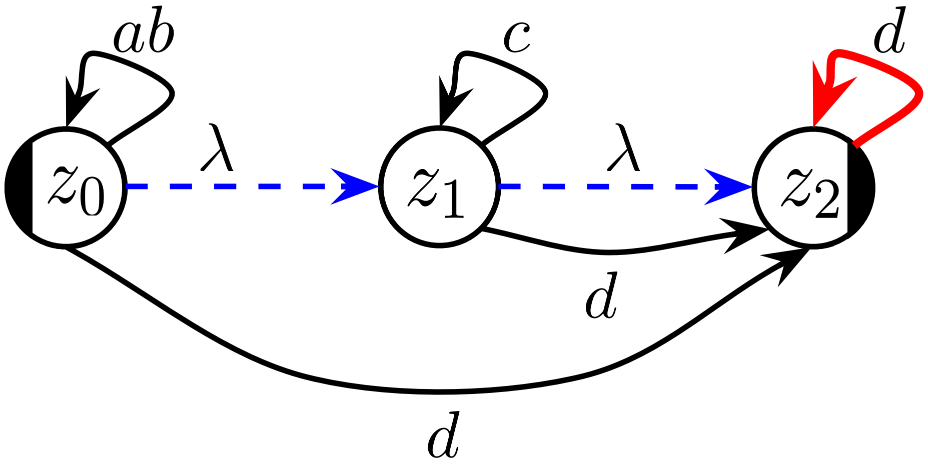
λ-Elimination

3. Betrachten wir die Kante (z_2, d, z_2):



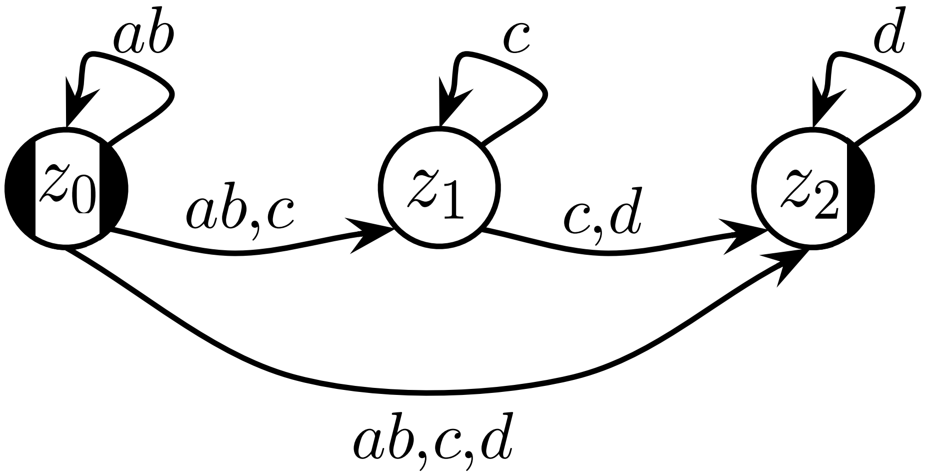
λ-Elimination

3. Betrachten wir die Kante (z_2, d, z_2):



λ-Elimination

Insgesamt ergibt sich folgendes Bild:



z_0 wird Endzustand, da ein λ-Pfad zu einem Endzustand existiert.

λ-frei (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein NFA, der möglicherweise λ-Kanten besitzt.

λ-frei (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein NFA, der möglicherweise λ-Kanten besitzt.
- Ein neuer NFA $B := (Z, \Sigma, K', Z_{\mathsf{start}}, Z'_{\mathsf{end}})$ ohne λ-Kanten wird wie folgt erklärt:

λ-frei (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein NFA, der möglicherweise λ-Kanten besitzt.
- Ein neuer NFA $B := (Z, \Sigma, K', Z_{\mathsf{start}}, Z'_{\mathsf{end}})$ ohne λ-Kanten wird wie folgt erklärt:

$$K' := \{(z, w, z''') \in Z \times \Sigma^* \times Z \mid \exists (z', w, z'') \in K$$

$$w \neq \lambda \wedge z \xrightarrow[\lambda]{*} z' \wedge z'' \xrightarrow[\lambda]{*} z''' \text{ (in A)}\}$$

λ-frei (formal)

- Sei $A := (Z, \Sigma, K, Z_{\text{start}}, Z_{\text{end}})$ ein NFA, der möglicherweise λ-Kanten besitzt.
- Ein neuer NFA $B := (Z, \Sigma, K', Z_{\text{start}}, Z'_{\text{end}})$ ohne λ-Kanten wird wie folgt erklärt:

$$
\begin{aligned}
K' \quad := \quad & \{(z, w, z''') \in Z \times \Sigma^* \times Z \mid \exists (z', w, z'') \in K \\
& w \neq \lambda \wedge z \xrightarrow[\lambda]{*} z' \wedge z'' \xrightarrow[\lambda]{*} z''' \text{ (in A)}\}
\end{aligned}
$$

und

$$Z'_{\text{end}} := Z_{\text{end}} \cup \{z \in Z_{\text{start}} \mid \exists z' \in Z_{\text{end}} : z \xrightarrow[\lambda]{*} z'\}.$$

Beweis: $L(A) = L(B)$

- $L(B) \subseteq L(A)$

Beweis: $L(A) = L(B)$

- $L(B) \subseteq L(A)$
  - (z, w, z') ist nur dann in K', wenn $z \xrightarrow[w]{*} z'$ im NFA A möglich ist.

Beweis: $L(A) = L(B)$

- $L(B) \subseteq L(A)$
  - (z, w, z') ist nur dann in K', wenn $z \xrightarrow[w]{*} z'$ im NFA A möglich ist.
  - ... wenn im Zustandsgraphen von A ein Pfad von z nach z' existiert, auf dem das Wort w gelesen wird.

Beweis: $L(A) = L(B)$

- $L(B) \subseteq L(A)$
  - (z, w, z') ist nur dann in K', wenn $z \xrightarrow[w]{*} z'$ im NFA A möglich ist.
  - ...wenn im Zustandsgraphen von A ein Pfad von z nach z' existiert, auf dem das Wort w gelesen wird.
  - $\Rightarrow$ zu jedem Erfolgspfad in B existiert auch einer in A.

Beweis: $L(A) = L(B)$

- $L(B) \subseteq L(A)$
  - (z, w, z') ist nur dann in K', wenn $z \xrightarrow[w]{*} z'$ im NFA A möglich ist.
  - …wenn im Zustandsgraphen von A ein Pfad von z nach z' existiert, auf dem das Wort w gelesen wird.
  - $\Rightarrow$ zu jedem Erfolgspfad in B existiert auch einer in A.
- $L(A) \subseteq L(B)$

Beweis: $L(A) = L(B)$

- $L(B) \subseteq L(A)$
  - (z, w, z') ist nur dann in K', wenn $z \xrightarrow[w]{*} z'$ im NFA A möglich ist.
  - ... wenn im Zustandsgraphen von A ein Pfad von z nach z' existiert, auf dem das Wort w gelesen wird.
  - $\Rightarrow$ zu jedem Erfolgspfad in B existiert auch einer in A.
- $L(A) \subseteq L(B)$
  - $\lambda \in L(A) \quad \Rightarrow \quad \lambda \in L(B)$, da dann einer der Startzustände von B per Definition in Z'_{end} ist.

Beweis: $L(A) = L(B)$

- $L(B) \subseteq L(A)$
  - (z, w, z') ist nur dann in K', wenn $z \xrightarrow[w]{*} z'$ im NFA A möglich ist.
  - ...wenn im Zustandsgraphen von A ein Pfad von z nach z' existiert, auf dem das Wort w gelesen wird.
  - $\Rightarrow$ zu jedem Erfolgspfad in B existiert auch einer in A.
- $L(A) \subseteq L(B)$
  - $\lambda \in L(A) \quad \Rightarrow \quad \lambda \in L(B)$, da dann einer der Startzustände von B per Definition in Z'_{end} ist.
  - Jede Kante $(z', w, z'') \in K$ mit $w \neq \lambda$ ist in K'!

Beweis: $L(A) = L(B)$

- $L(A) \subseteq L(B)$ [Fortsetzung]

Beweis: $L(A) = L(B)$

- $L(A) \subseteq L(B)$ [Fortsetzung]
  - Erfolgspfad p in A:

$$z_1 \xrightarrow[\lambda]{*} z_1' \xrightarrow[w_1]{} z_2 \xrightarrow[\lambda]{*} z_2' \xrightarrow[w_2]{} z_3 \xrightarrow[\lambda]{*} \ldots \xrightarrow[\lambda]{*} z_n' \xrightarrow[w_n]{} z_{n+1} \xrightarrow[\lambda]{*}$$

Beweis: $L(A) = L(B)$

- $L(A) \subseteq L(B)$ [Fortsetzung]
  - Erfolgspfad p in A:

$$z_1 \xrightarrow[\lambda]{*} z_1' \xrightarrow[w_1]{} z_2 \xrightarrow[\lambda]{*} z_2' \xrightarrow[w_2]{} z_3 \xrightarrow[\lambda]{*} \ldots \xrightarrow[\lambda]{*} z_n' \xrightarrow[w_n]{} z_{n+1} \xrightarrow[\lambda]{*}$$

  - $z_i' \xrightarrow[w_i]{} z_{i+1}$ ist Übergang mit Kante (z_i', w_i, z_{i+1}) von A, $(w_i \neq \lambda)$.

Beweis: $L(A) = L(B)$

- $L(A) \subseteq L(B)$ [Fortsetzung]
  - Erfolgspfad p in A:

$$z_1 \xrightarrow[\lambda]{*} z_1' \xrightarrow[w_1]{} z_2 \xrightarrow[\lambda]{*} z_2' \xrightarrow[w_2]{} z_3 \xrightarrow[\lambda]{*} \cdots \xrightarrow[\lambda]{*} z_n' \xrightarrow[w_n]{} z_{n+1} \xrightarrow[\lambda]{*}$$

  - $z_i' \xrightarrow[w_i]{} z_{i+1}$ ist Übergang mit Kante (z_i', w_i, z_{i+1}) von A, $(w_i \neq \lambda)$.
  - Kante muss in Pfad p mit $|p| \neq \lambda$ vorkommen

Beweis: $L(A) = L(B)$

- $L(A) \subseteq L(B)$ [Fortsetzung]
  - Erfolgspfad p in A:

$$z_1 \xrightarrow[\lambda]{*} z_1' \xrightarrow[w_1]{} z_2 \xrightarrow[\lambda]{*} z_2' \xrightarrow[w_2]{} z_3 \xrightarrow[\lambda]{*} \ldots \xrightarrow[\lambda]{*} z_n' \xrightarrow[w_n]{} z_{n+1} \xrightarrow[\lambda]{*}$$

  - $z_i' \xrightarrow[w_i]{} z_{i+1}$ ist Übergang mit Kante (z_i', w_i, z_{i+1}) von A, $(w_i \neq \lambda)$.
  - Kante muss in Pfad p mit $|p| \neq \lambda$ vorkommen
  - Nach Definition von K' ist $(z_i, w_i, z_{i+1}) \in K'$ für $1 \leq i < n$ sowie $(z_n, w_n, z_{n+1}') \in K'$.

Beweis: $L(A) = L(B)$

- $L(A) \subseteq L(B)$ [Fortsetzung]
  - Erfolgspfad p in A:

$$z_1 \xrightarrow[\lambda]{*} z_1' \xrightarrow[w_1]{} z_2 \xrightarrow[\lambda]{*} z_2' \xrightarrow[w_2]{} z_3 \xrightarrow[\lambda]{*} \ldots \xrightarrow[\lambda]{*} z_n' \xrightarrow[w_n]{} z_{n+1} \xrightarrow[\lambda]{*}$$

  - $z_i' \xrightarrow[w_i]{} z_{i+1}$ ist Übergang mit Kante (z_i', w_i, z_{i+1}) von A, $(w_i \neq \lambda)$.
  - Kante muss in Pfad p mit $|p| \neq \lambda$ vorkommen
  - Nach Definition von K' ist $(z_i, w_i, z_{i+1}) \in K'$ für $1 \leq i < n$ sowie $(z_n, w_n, z_{n+1}') \in K'$.
  - Also: $z_1 \xrightarrow[|p|]{*} z_{n+1}'$ in B Erfolgspfad in B.

buchstabierend

- **Theorem:** Zu jedem λ-freien NFA gibt es einen äquivalenten buchstabierenden NFA.

buchstabierend

- **Theorem:** Zu jedem λ-freien NFA gibt es einen äquivalenten buchstabierenden NFA.
- Zu jeder Kante $k := (z, w, z') \in K$ des NFA $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ mit $|w| \geq 2$ werden $|w| - 1$ neue (Zwischen-)Zustände z_{k_i} definiert.

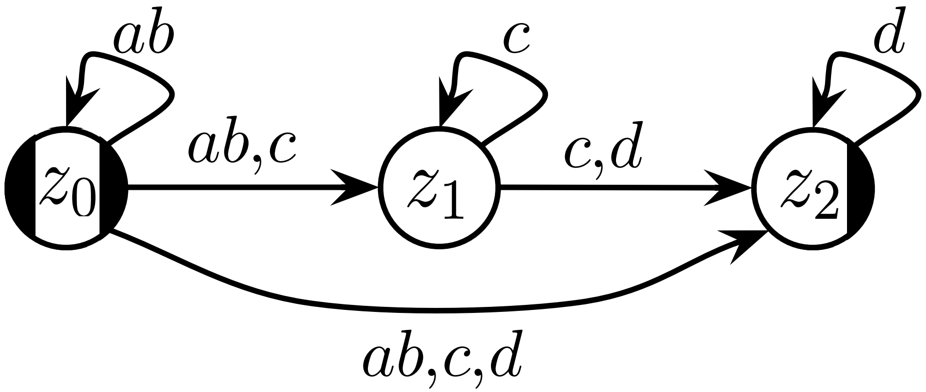
buchstabierend

- **Theorem:** Zu jedem λ-freien NFA gibt es einen äquivalenten buchstabierenden NFA.
- Zu jeder Kante $k := (z, w, z') \in K$ des NFA $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ mit $|w| \geq 2$ werden $|w| - 1$ neue (Zwischen-)Zustände z_{k_i} definiert.
- Die Kante $k = (z, w, z')$ mit $w = x_1 x_2 \cdots x_n$ wird dann ersetzt durch die Kanten der Menge

$$\{(z, x_1, z_{k_1}), (z_{k_1}, x_2, z_{k_2}), \ldots, (z_{k_{n-1}}, x_n, z')\}.$$

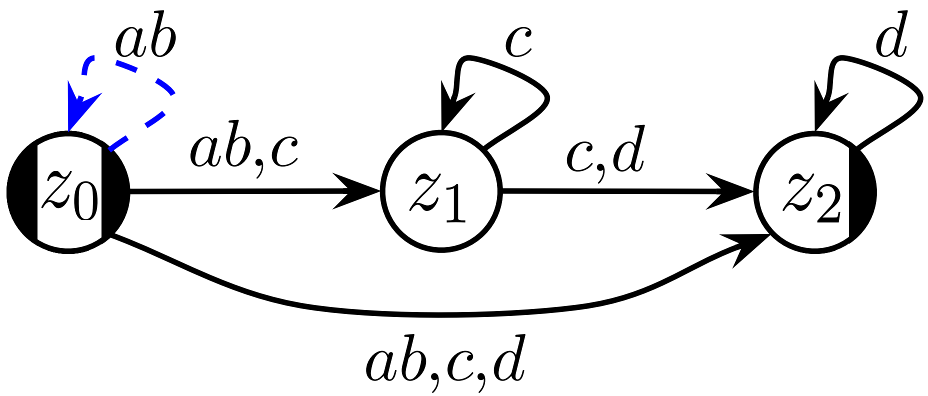
Konstr.: buchstabierender NFA

Wir gehen vom vorigen λ-freien Automaten aus:



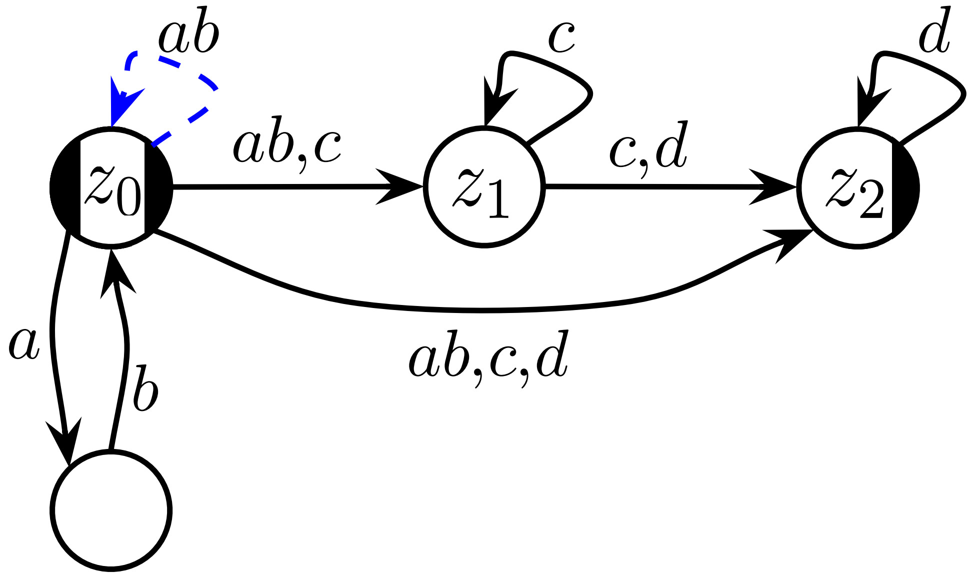
Konstr.: buchstabierender NFA

1. Betrachten wir die Kante (z_0, ab, z_0):



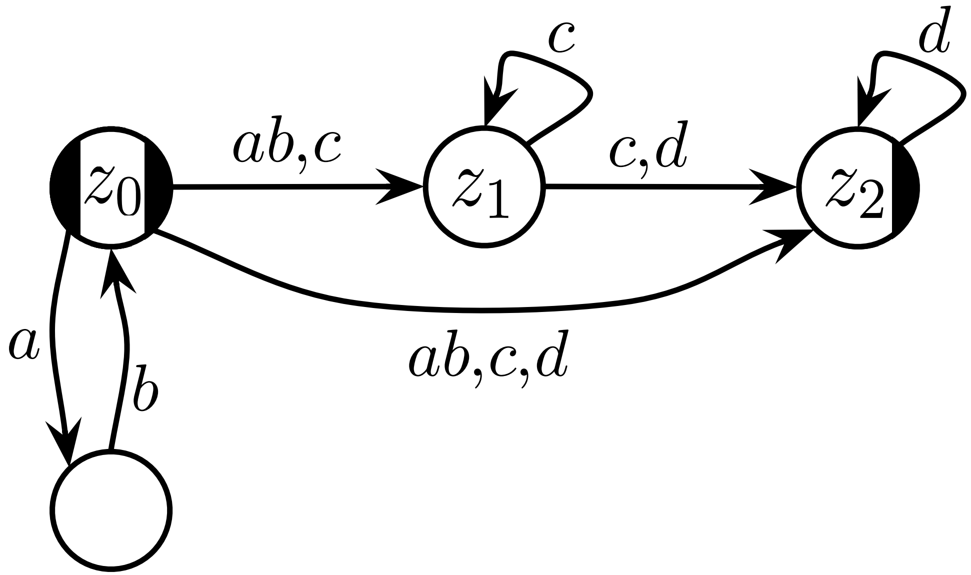
Konstr.: buchstabierender NFA

1. Betrachten wir die Kante (z_0, ab, z_0):



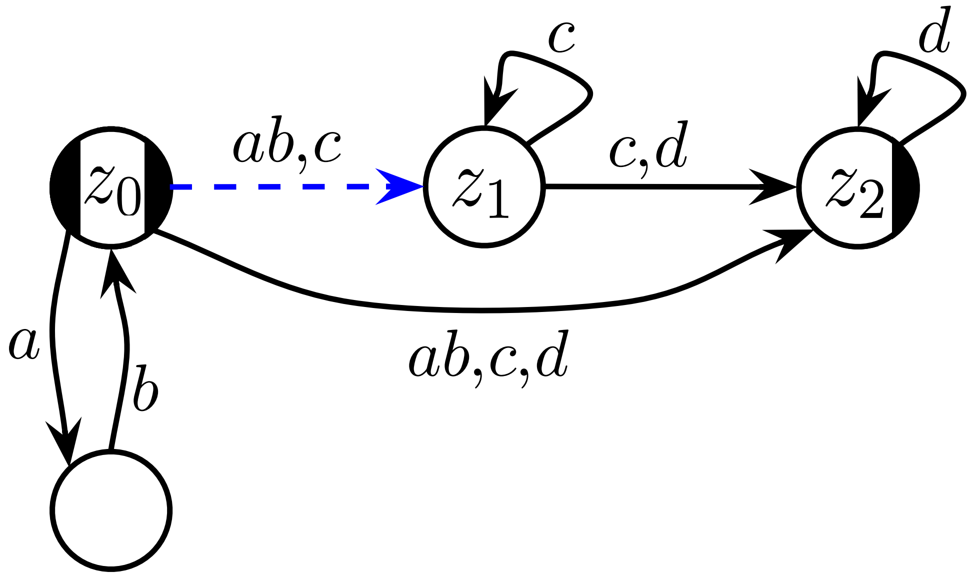
Konstr.: buchstabierender NFA

1. Betrachten wir die Kante (z_0, ab, z_0):



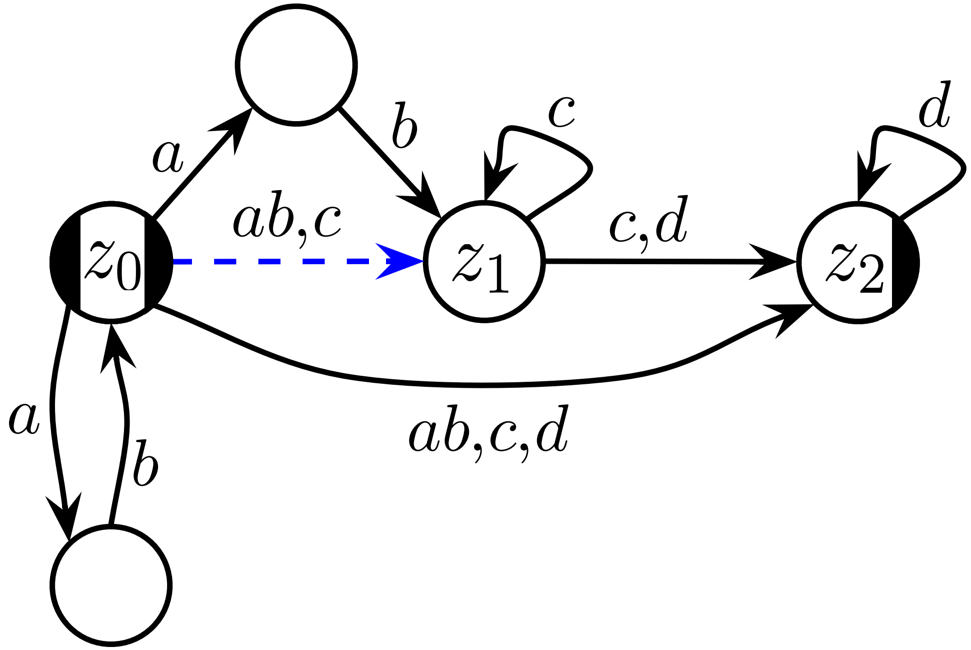
Konstr.: buchstabierender NFA

2. Betrachten wir die Kante (z_0, ab, z_1):



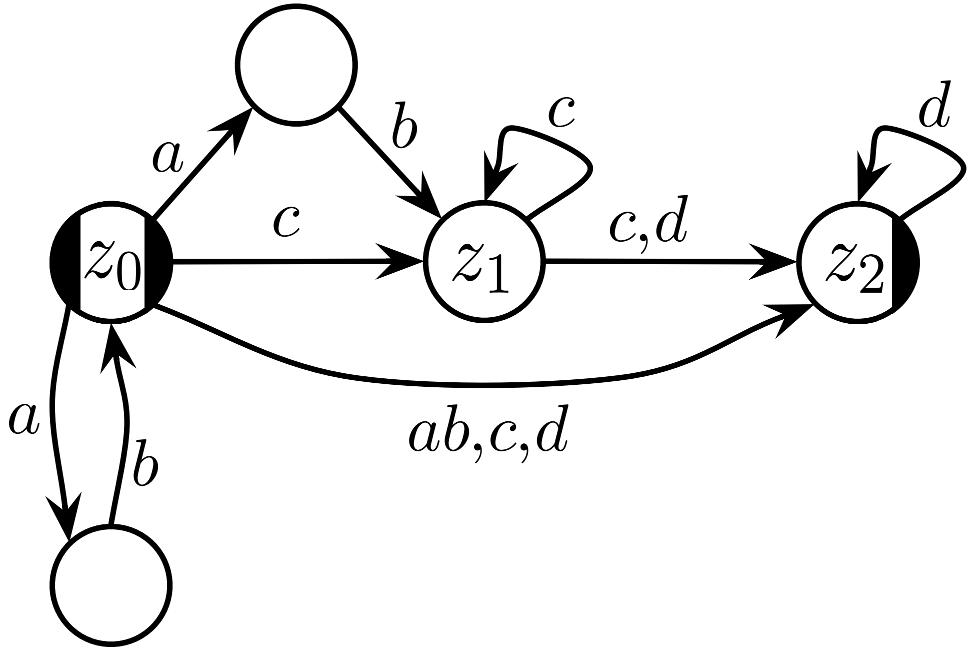
Konstr.: buchstabierender NFA

2. Betrachten wir die Kante (z_0, ab, z_1):



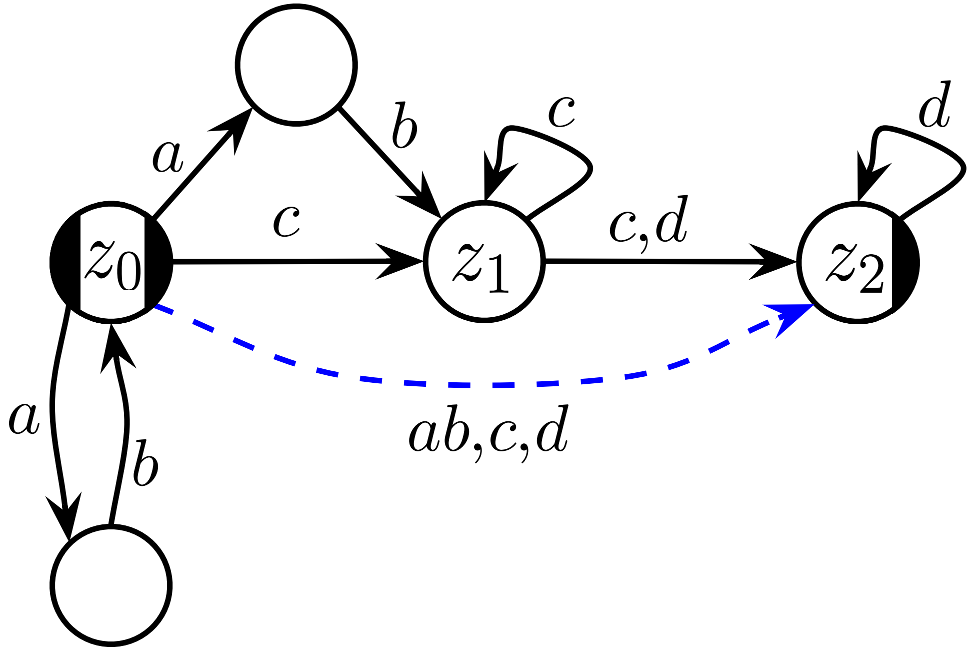
Konstr.: buchstabierender NFA

2. Betrachten wir die Kante (z_0, ab, z_1):



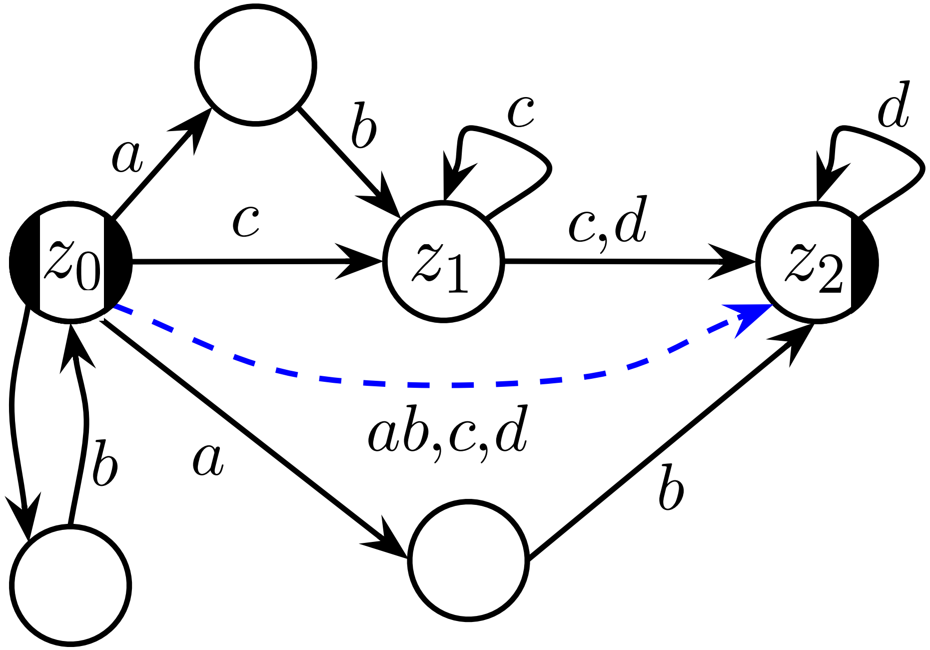
Konstr.: buchstabierender NFA

3. Betrachten wir die Kante (z_0, ab, z_2):



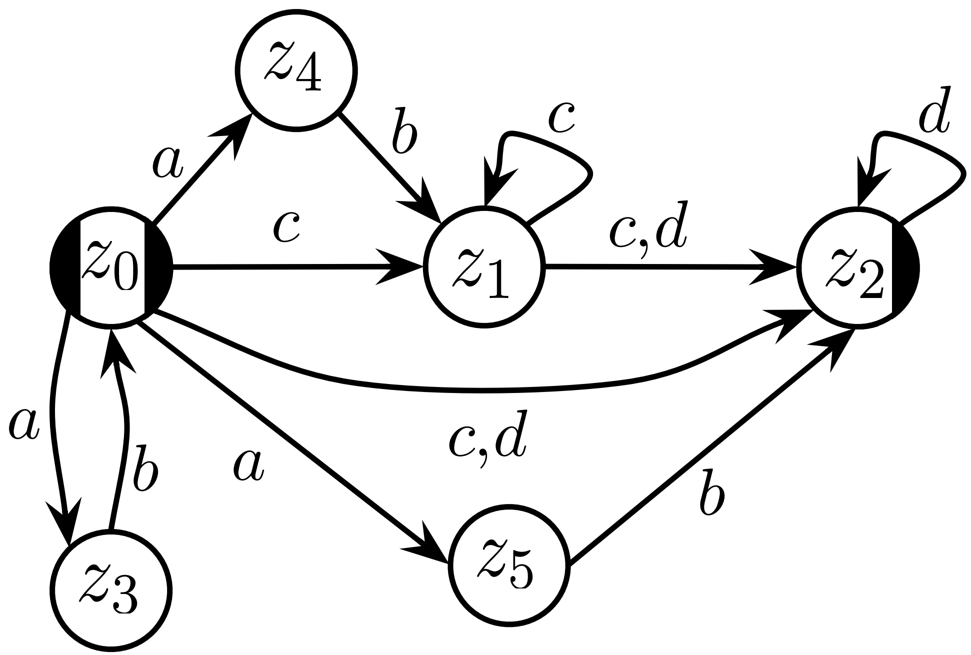
Konstr.: buchstabierender NFA

3. Betrachten wir die Kante (z_0, ab, z_2):



Konstr.: buchstabierender NFA

Insgesamt ergibt sich folgendes Bild:



Idee: aus NFA wird DFA

- Gegeben sei ein λ-**freier, buchstabierender** NFA $A = (Z, \Sigma, Z_{\mathsf{start}}, Z_{\mathsf{end}})$.

Idee: aus NFA wird DFA

- Gegeben sei ein λ-**freier, buchstabierender** NFA $A = (Z, \Sigma, Z_{\text{start}}, Z_{\text{end}})$.
- Konstruiere einen DFA folgendermaßen:

Idee: aus NFA wird DFA

- Gegeben sei ein λ-**freier, buchstabierender** NFA $A = (Z, \Sigma, Z_{\mathsf{start}}, Z_{\mathsf{end}})$.
- Konstruiere einen DFA folgendermaßen:
  - Bezeichne Zustände mit den Teilmengen von Z, d.h. neue Zustandsmenge ist $Z' = 2^Z$.

Idee: aus NFA wird DFA

- Gegeben sei ein λ-**freier, buchstabierender** NFA $A = (Z, \Sigma, Z_{\text{start}}, Z_{\text{end}})$.
- Konstruiere einen DFA folgendermaßen:
  - Bezeichne Zustände mit den Teilmengen von Z, d.h. neue Zustandsmenge ist $Z' = 2^Z$.
  - Einziger Startzustand sei $\{z \mid z \in Z_{\text{start}}\}$.

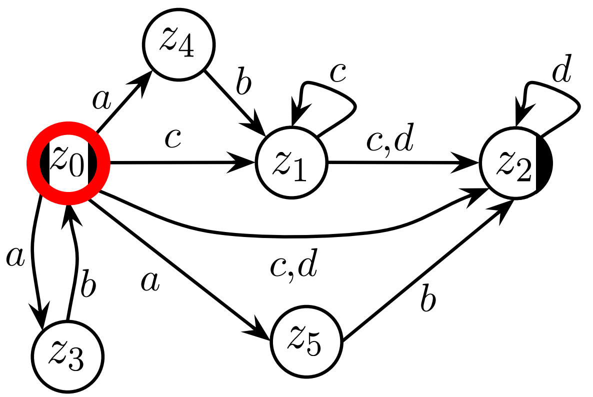
Idee: aus NFA wird DFA

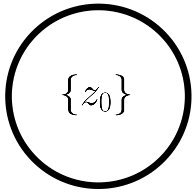
- Gegeben sei ein λ-**freier, buchstabierender** NFA $A = (Z, \Sigma, Z_{\text{start}}, Z_{\text{end}})$.
- Konstruiere einen DFA folgendermaßen:
  - Bezeichne Zustände mit den Teilmengen von Z, d.h. neue Zustandsmenge ist $Z' = 2^Z$.
  - Einziger Startzustand sei $\{z \mid z \in Z_{\text{start}}\}$.
  - Zeichne Kante mit Inschrift $a \in \Sigma$ von $\hat{Z} \subseteq Z$ nach $\bigcup\limits_{\substack{(\hat{z},a,z)\in K \\ \hat{z}\in\hat{Z}}} \{z\}$

Idee: aus NFA wird DFA

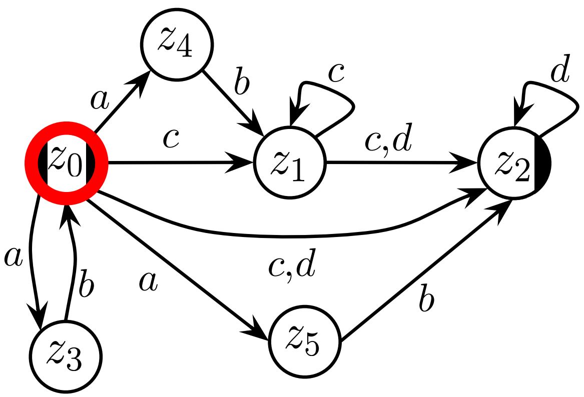
- Gegeben sei ein λ-**freier, buchstabierender** NFA $A = (Z, \Sigma, Z_{\text{start}}, Z_{\text{end}})$.
- Konstruiere einen DFA folgendermaßen:
  - Bezeichne Zustände mit den Teilmengen von Z, d.h. neue Zustandsmenge ist $Z' = 2^Z$.
  - Einziger Startzustand sei $\{z \mid z \in Z_{\text{start}}\}$.
  - Zeichne Kante mit Inschrift $a \in \Sigma$ von $\hat{Z} \subseteq Z$ nach $\bigcup\limits_{\substack{(\hat{z},a,z)\in K \\ \hat{z}\in\hat{Z}}} \{z\}$
  - Endzustand sind alle $\tilde{Z} \in Z'$ mit $\tilde{Z} \cap Z_{\text{end}} \neq \emptyset$

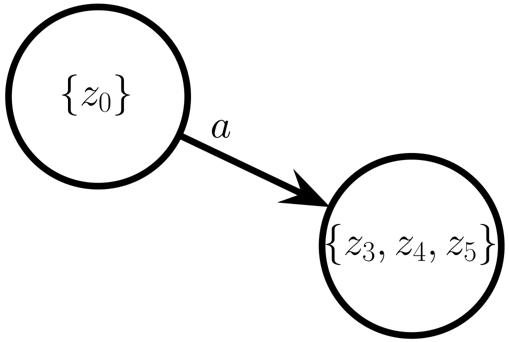
Potenzautomatenkonstruktion



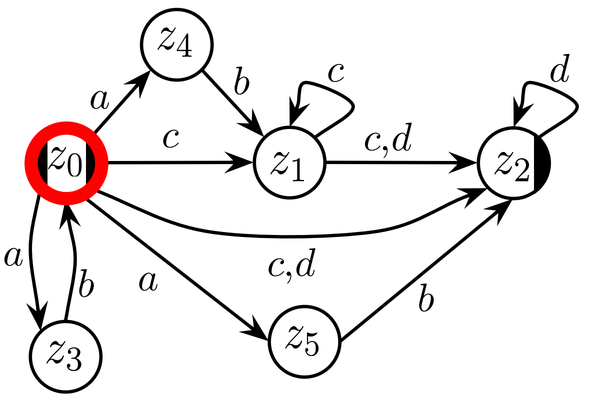


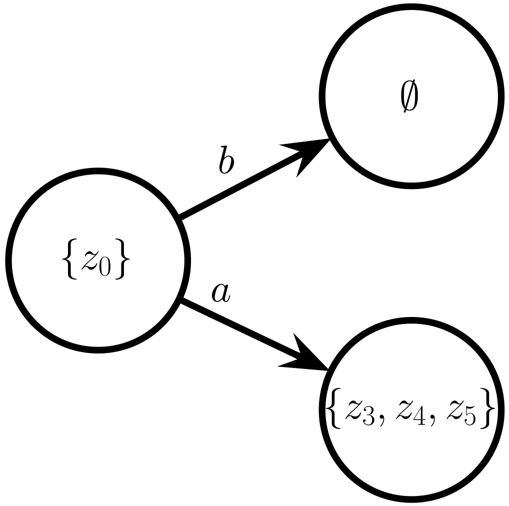
Potenzautomatenkonstruktion



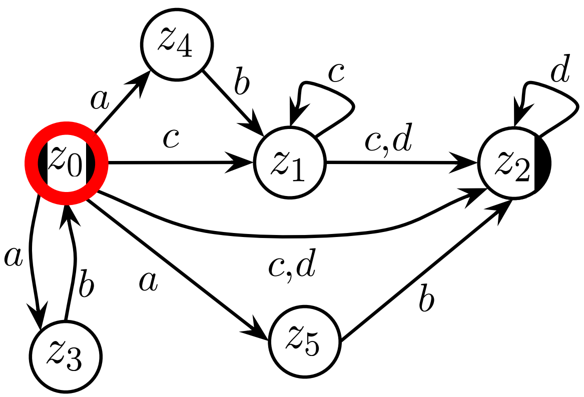


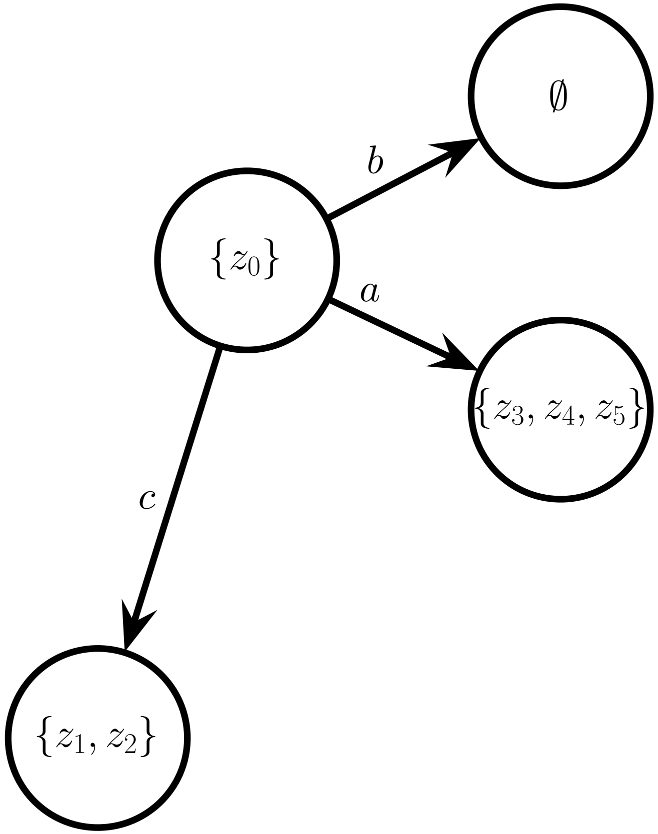
Potenzautomatenkonstruktion



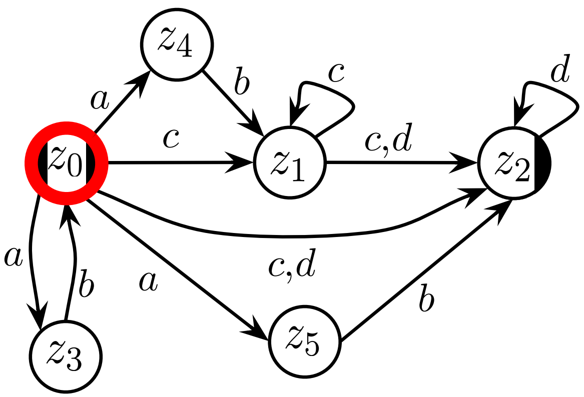


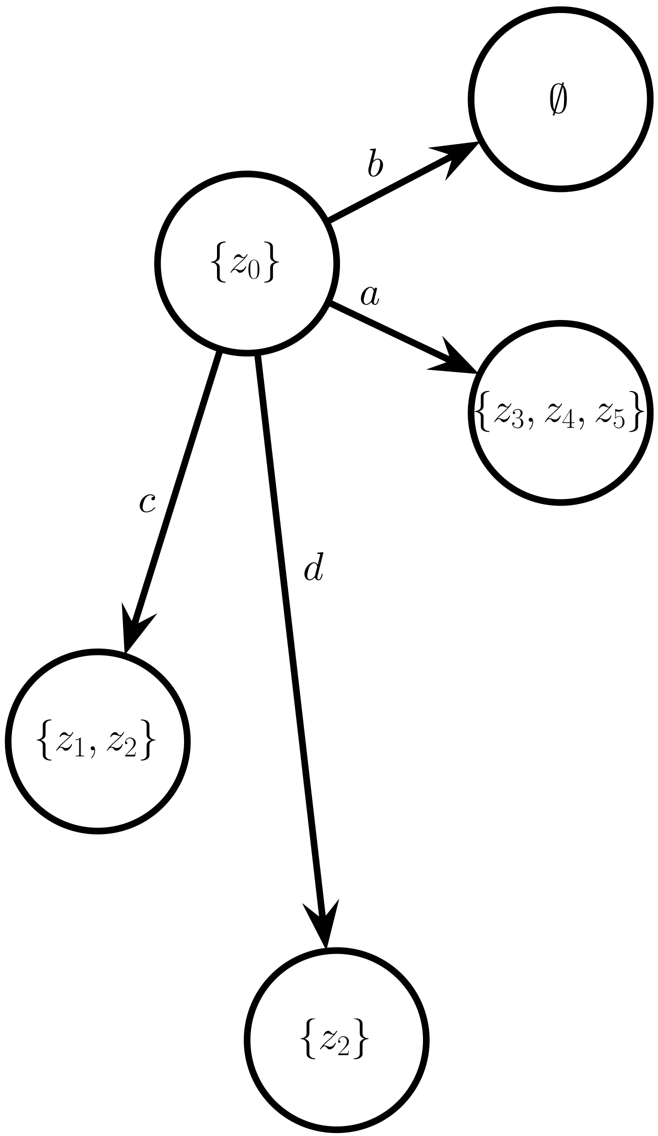
Potenzautomatenkonstruktion



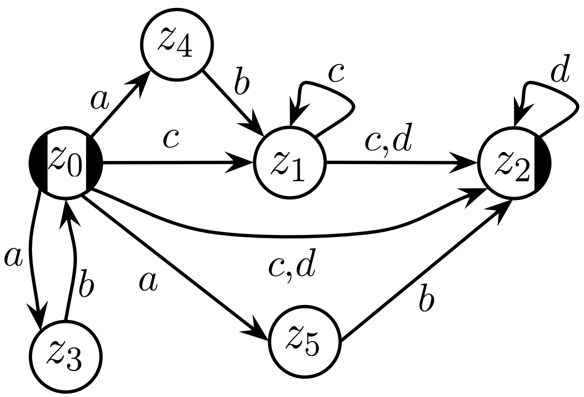


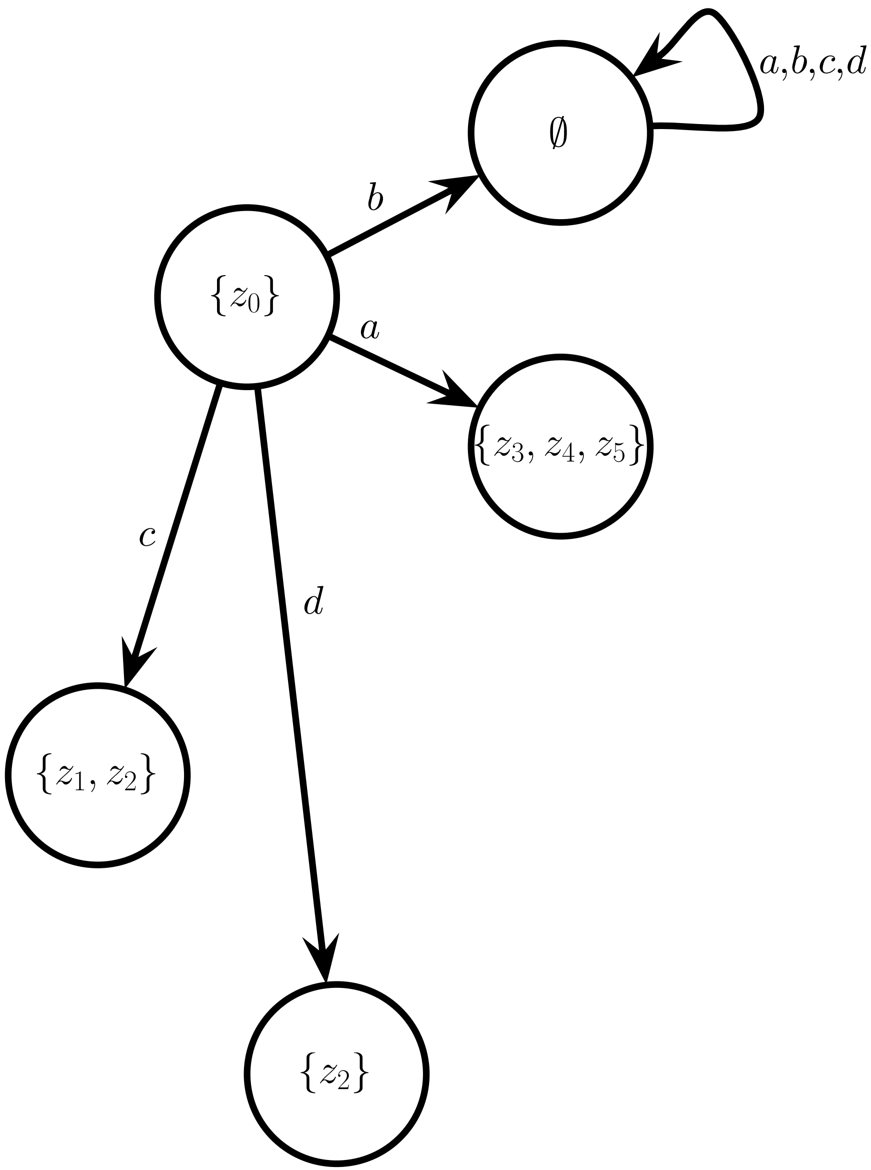
Potenzautomatenkonstruktion



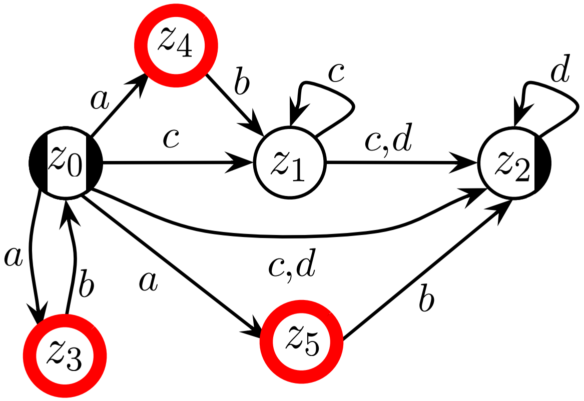


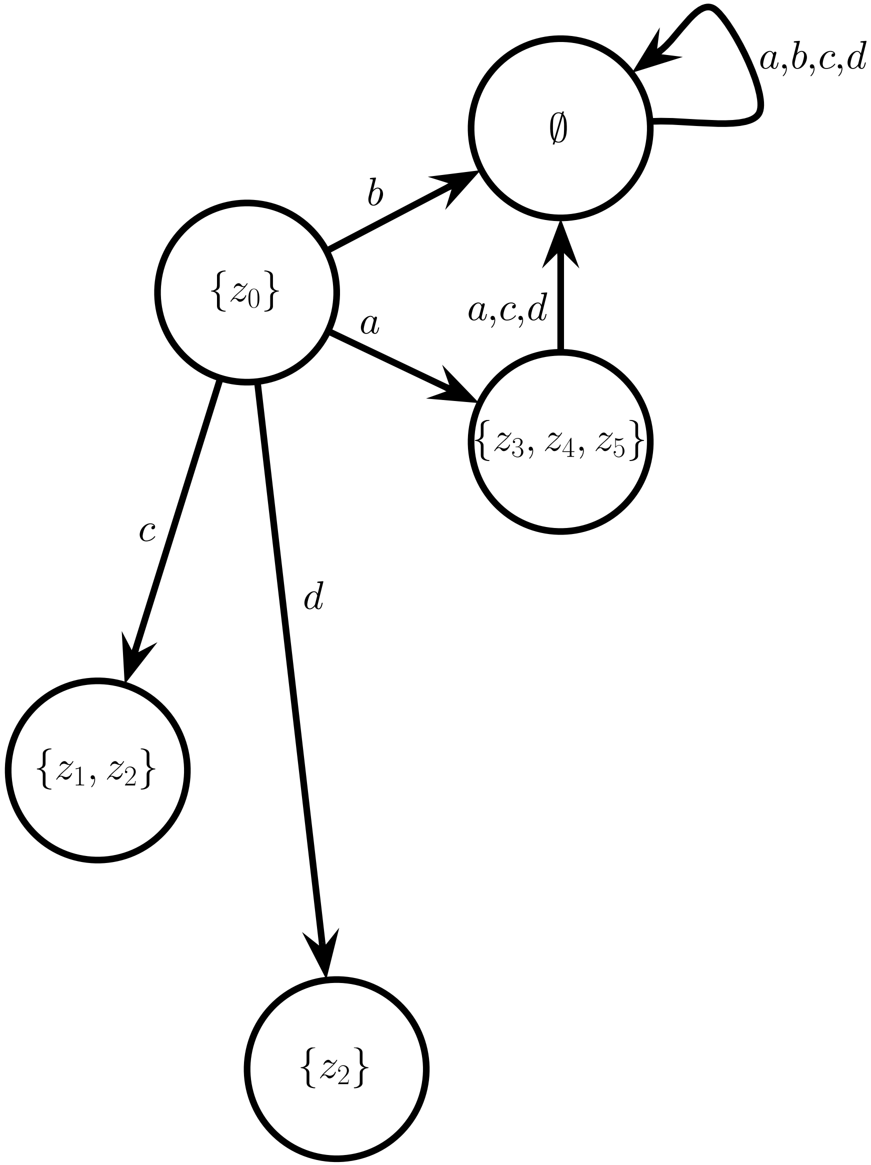
Potenzautomatenkonstruktion



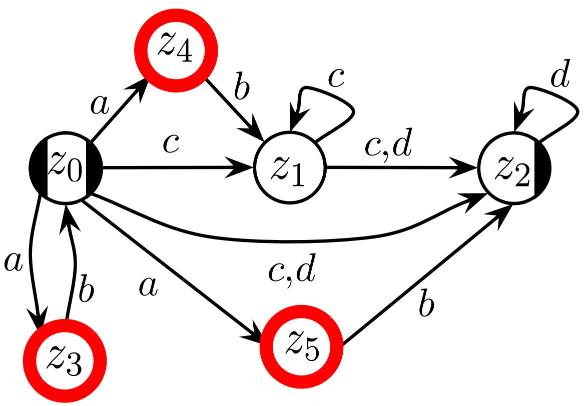


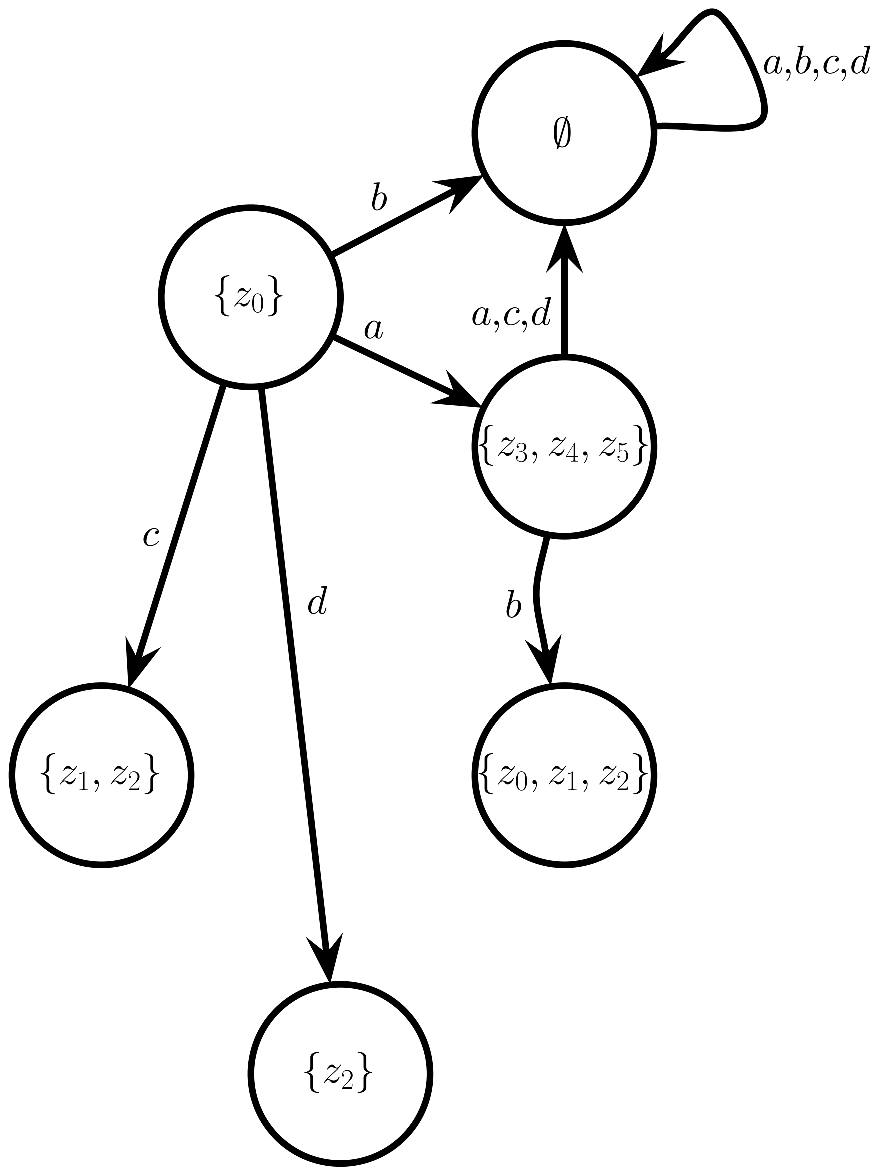
Potenzautomatenkonstruktion



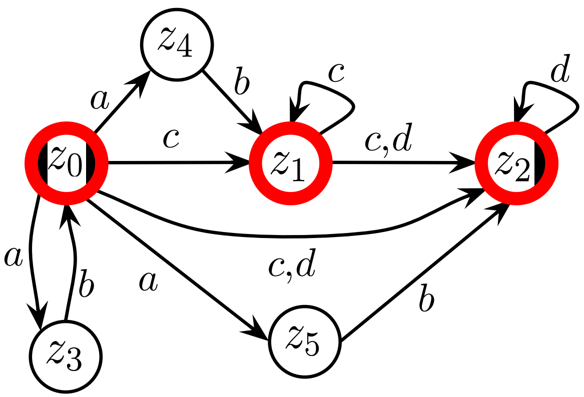


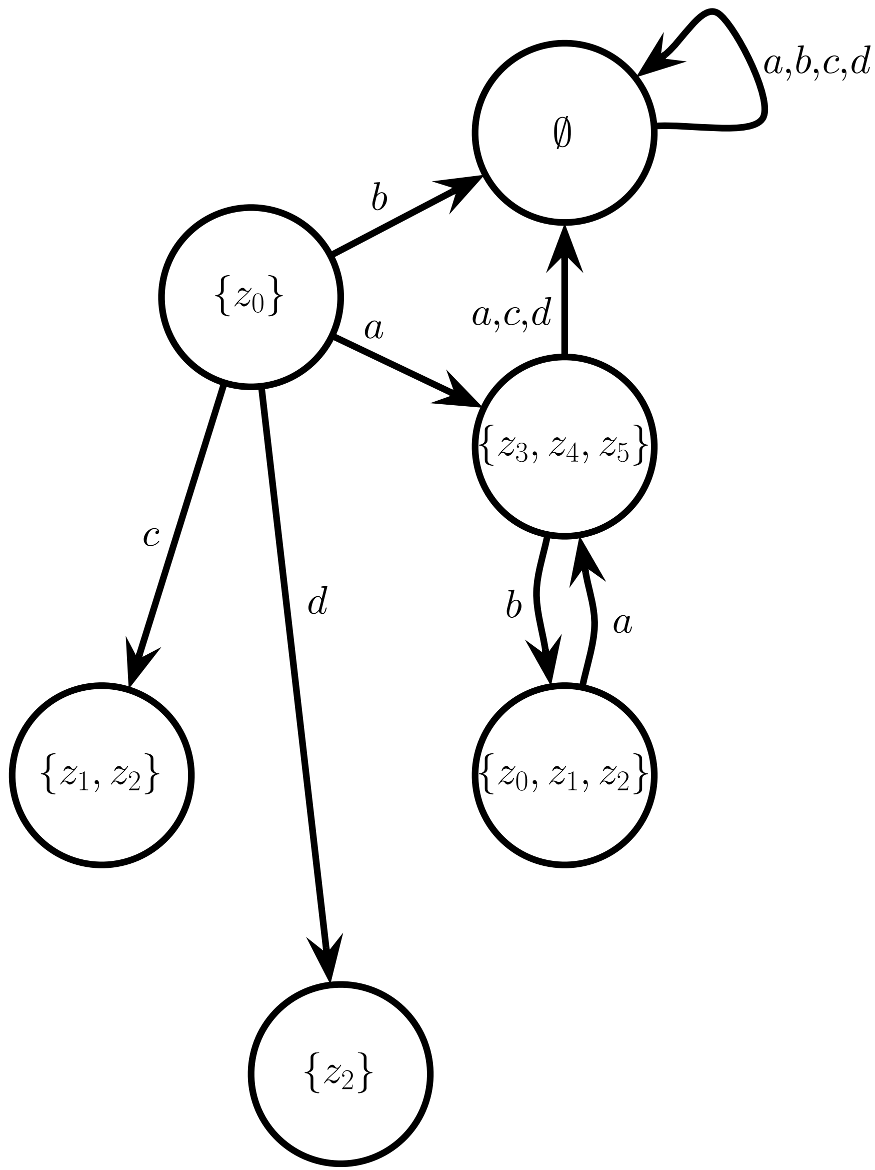
Potenzautomatenkonstruktion



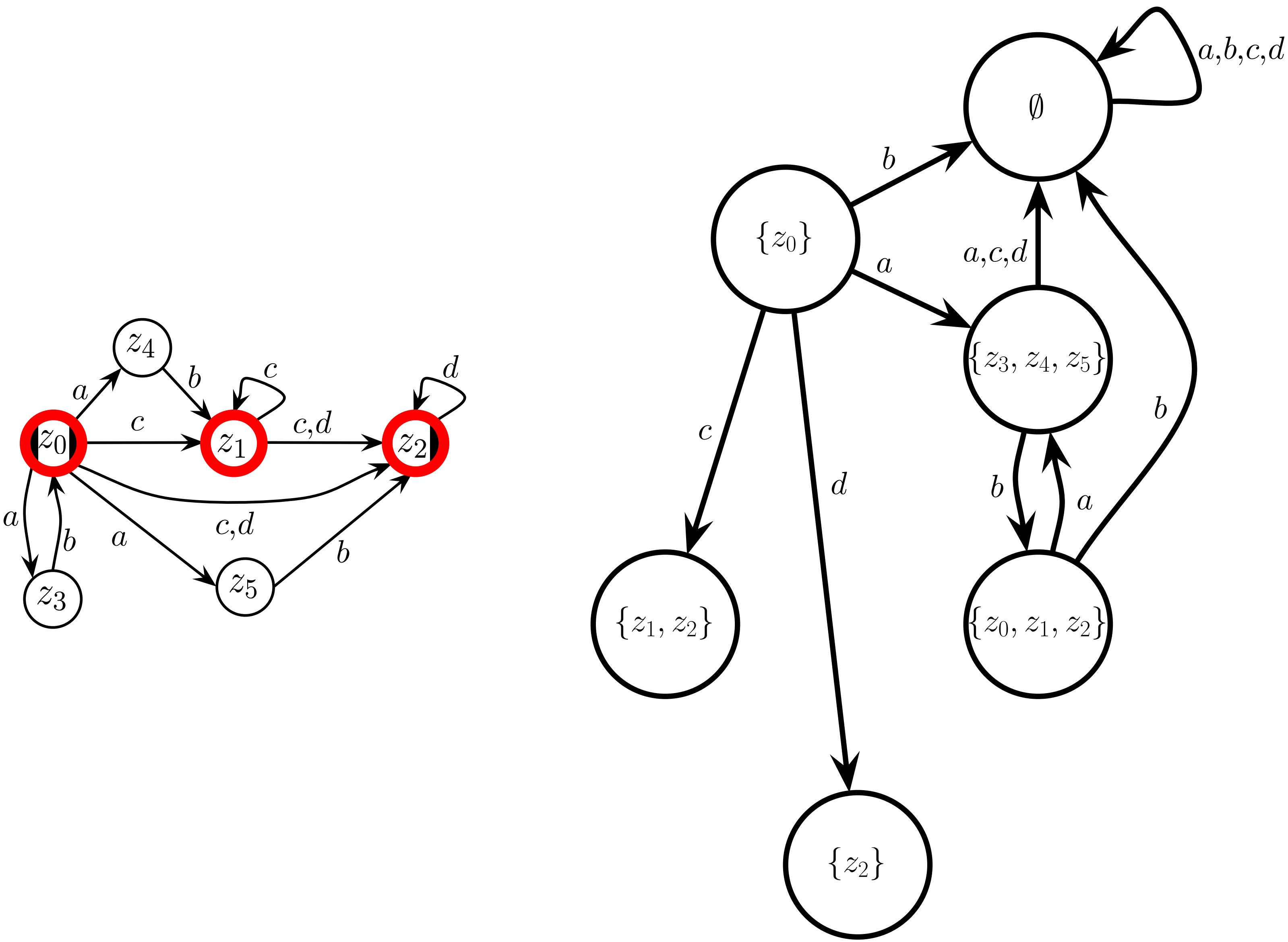


Potenzautomatenkonstruktion

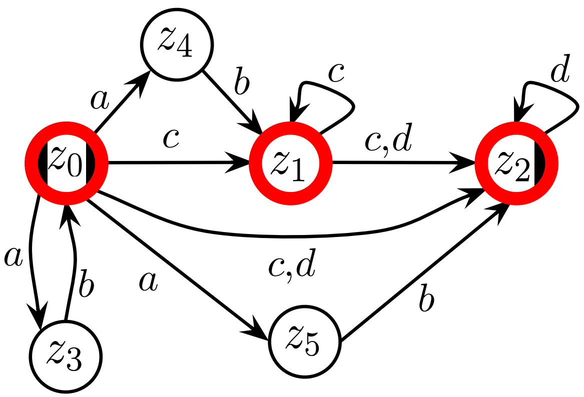


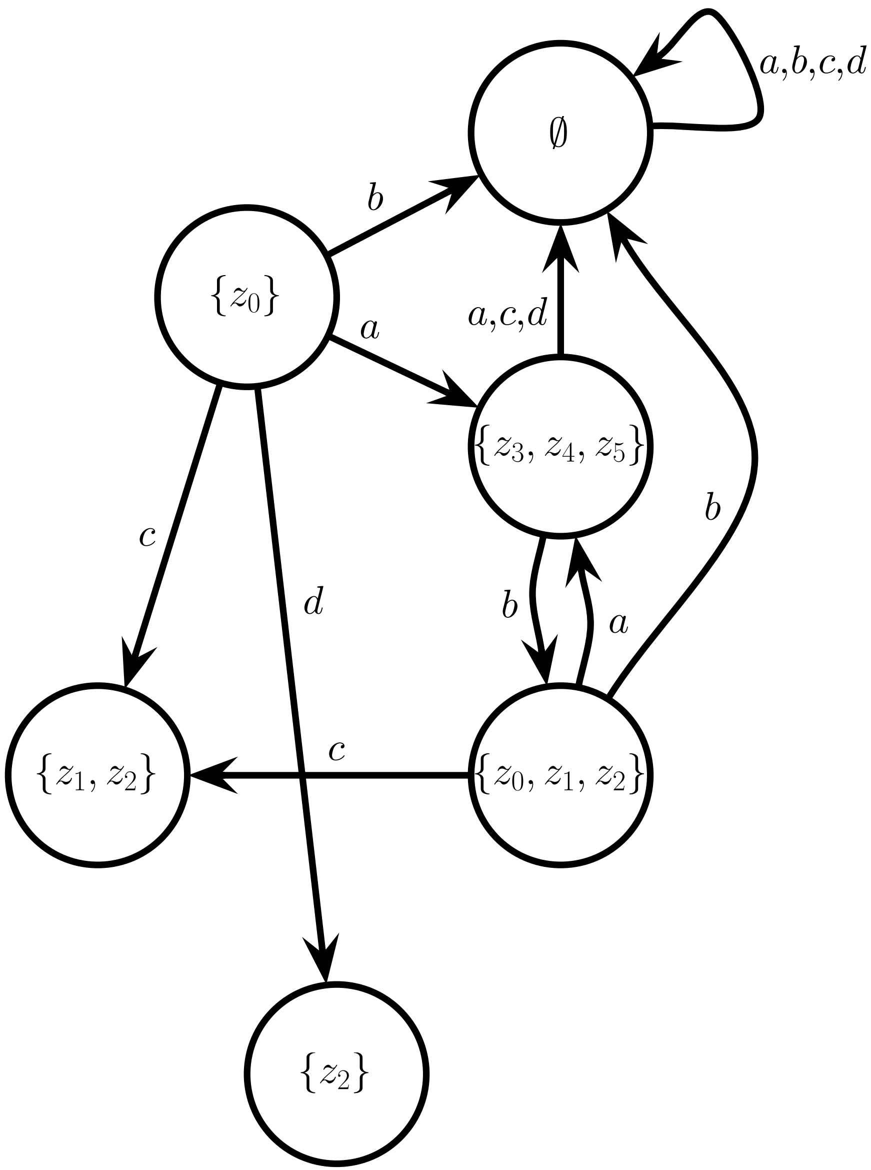


Potenzautomatenkonstruktion

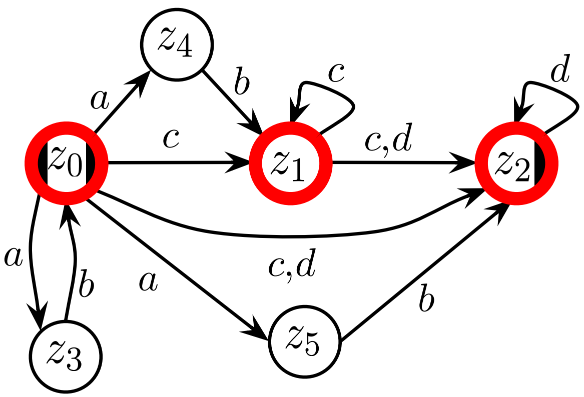


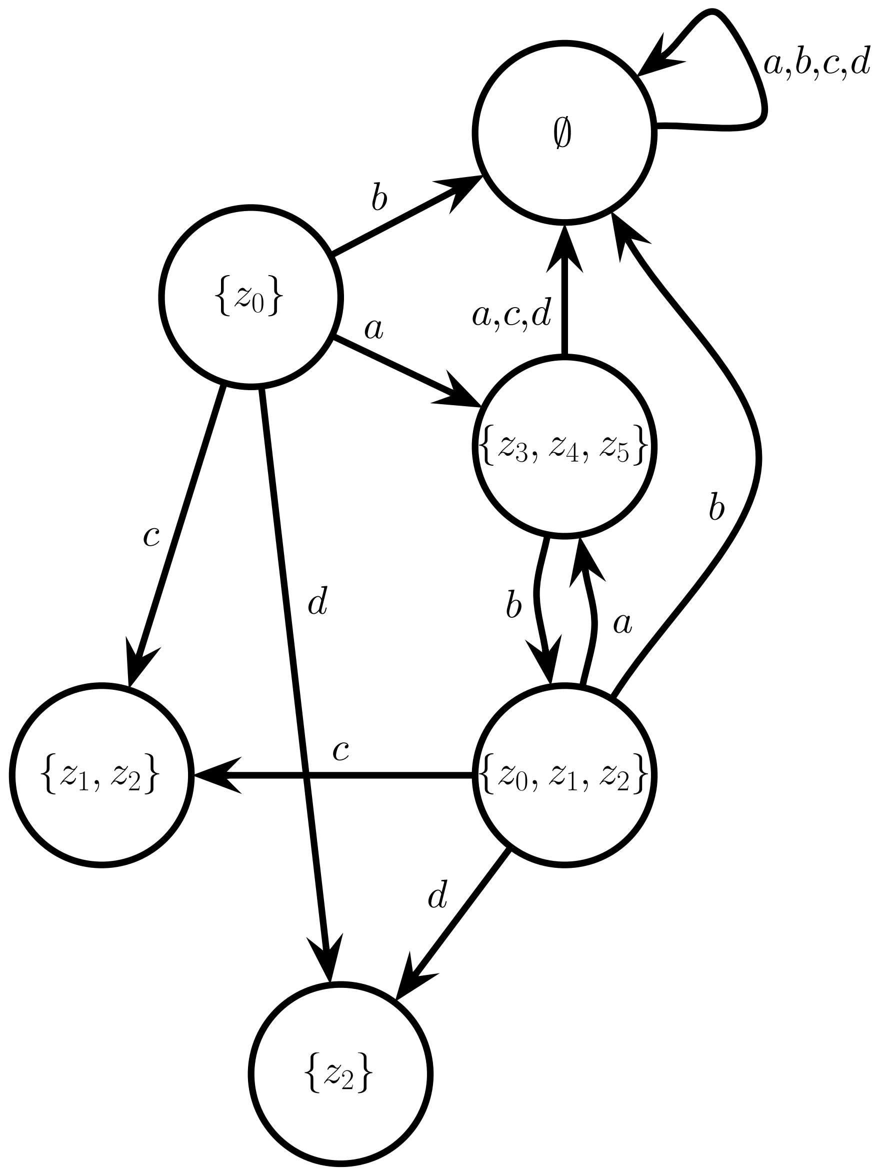
Potenzautomatenkonstruktion



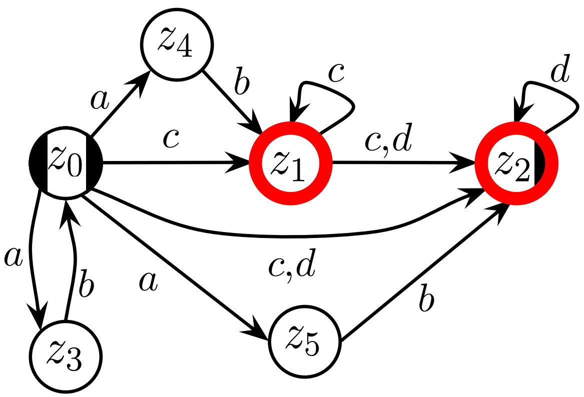


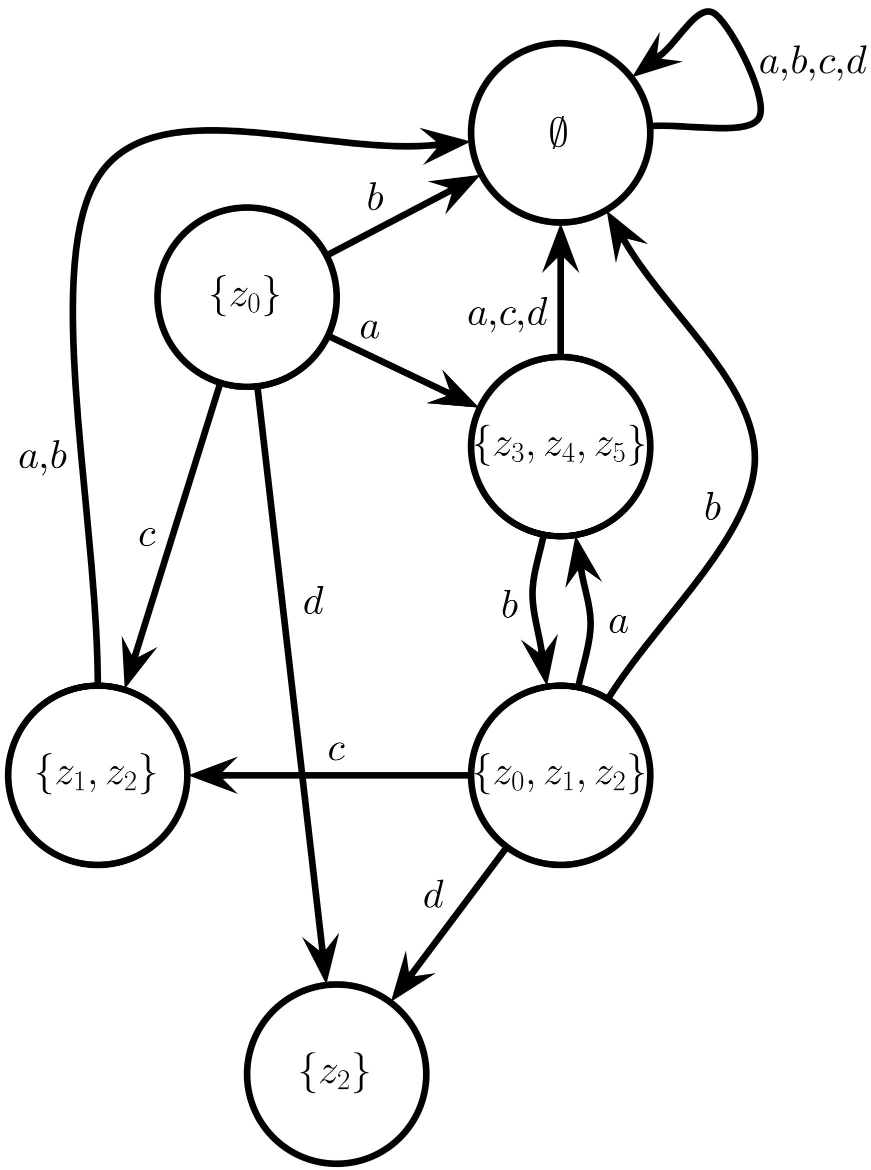
Potenzautomatenkonstruktion



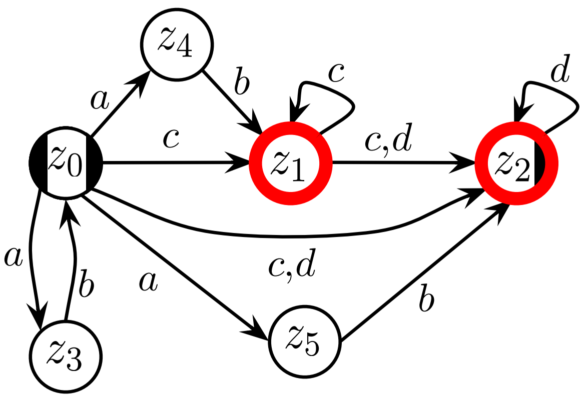


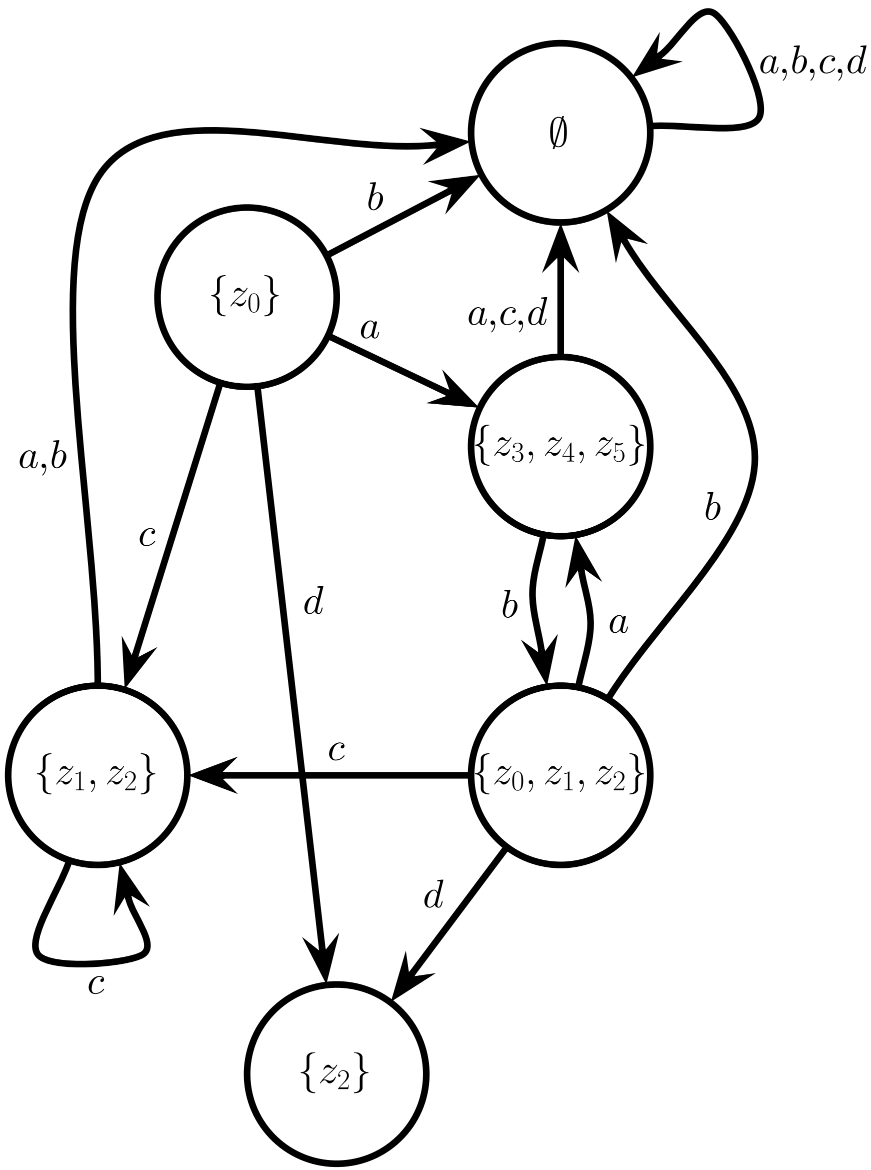
Potenzautomatenkonstruktion



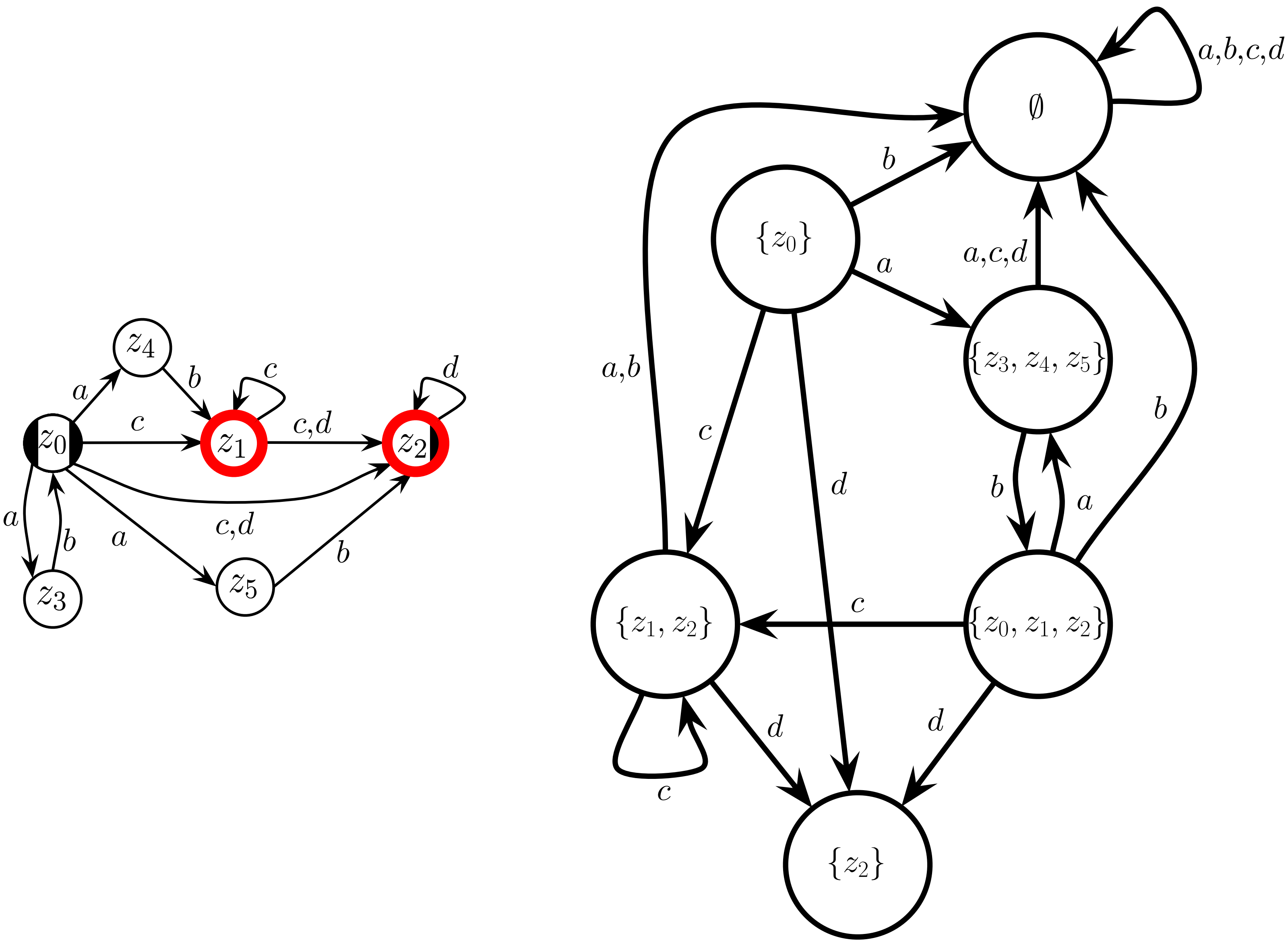


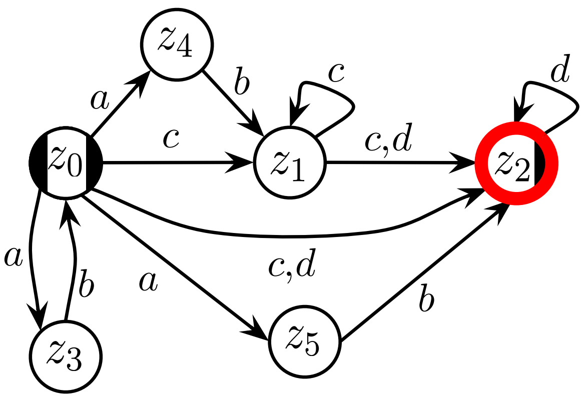
Potenzautomatenkonstruktion

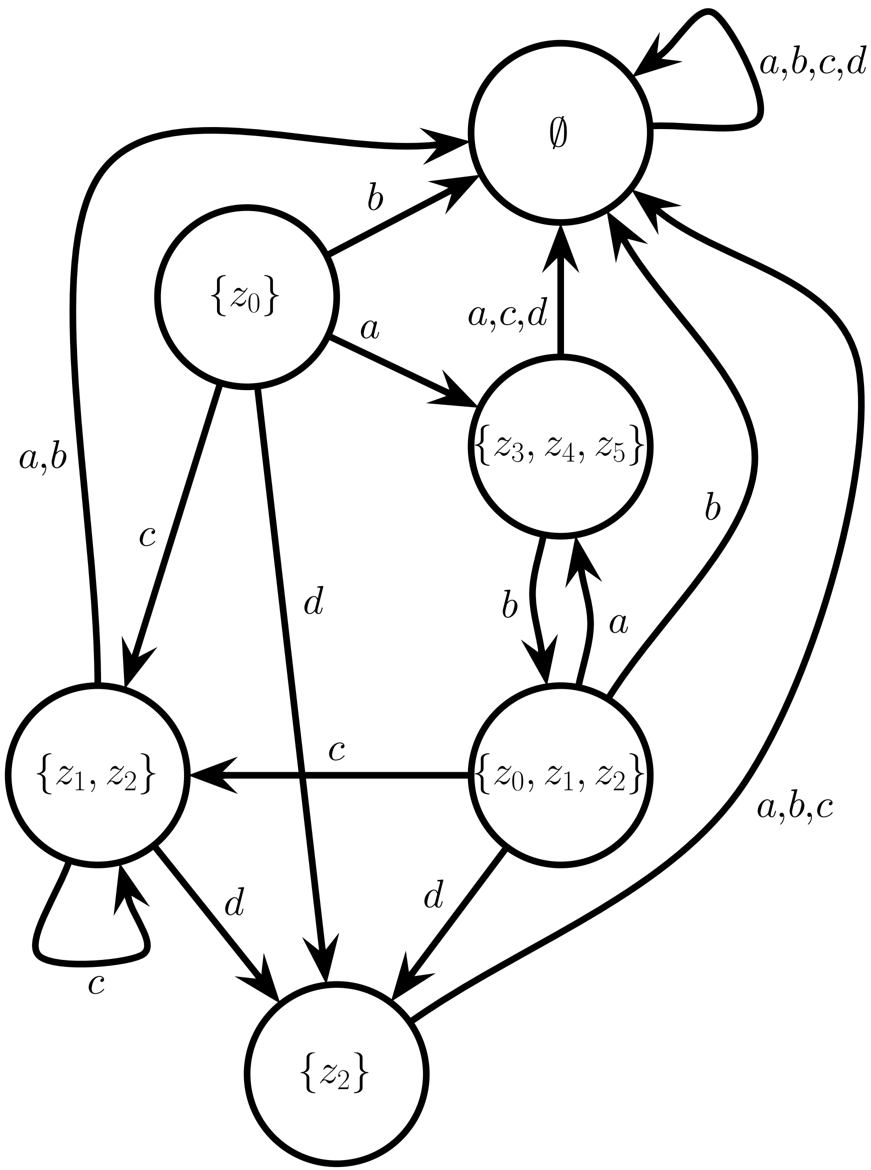




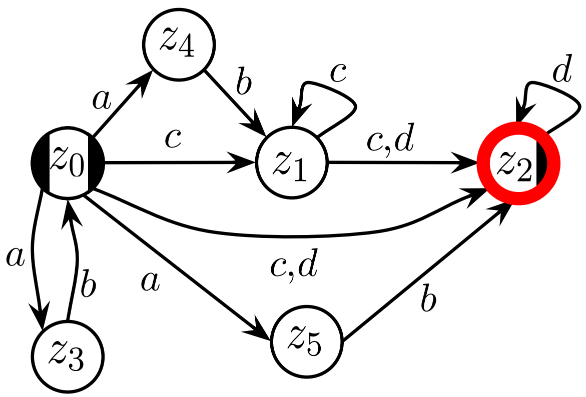
Potenzautomatenkonstruktion

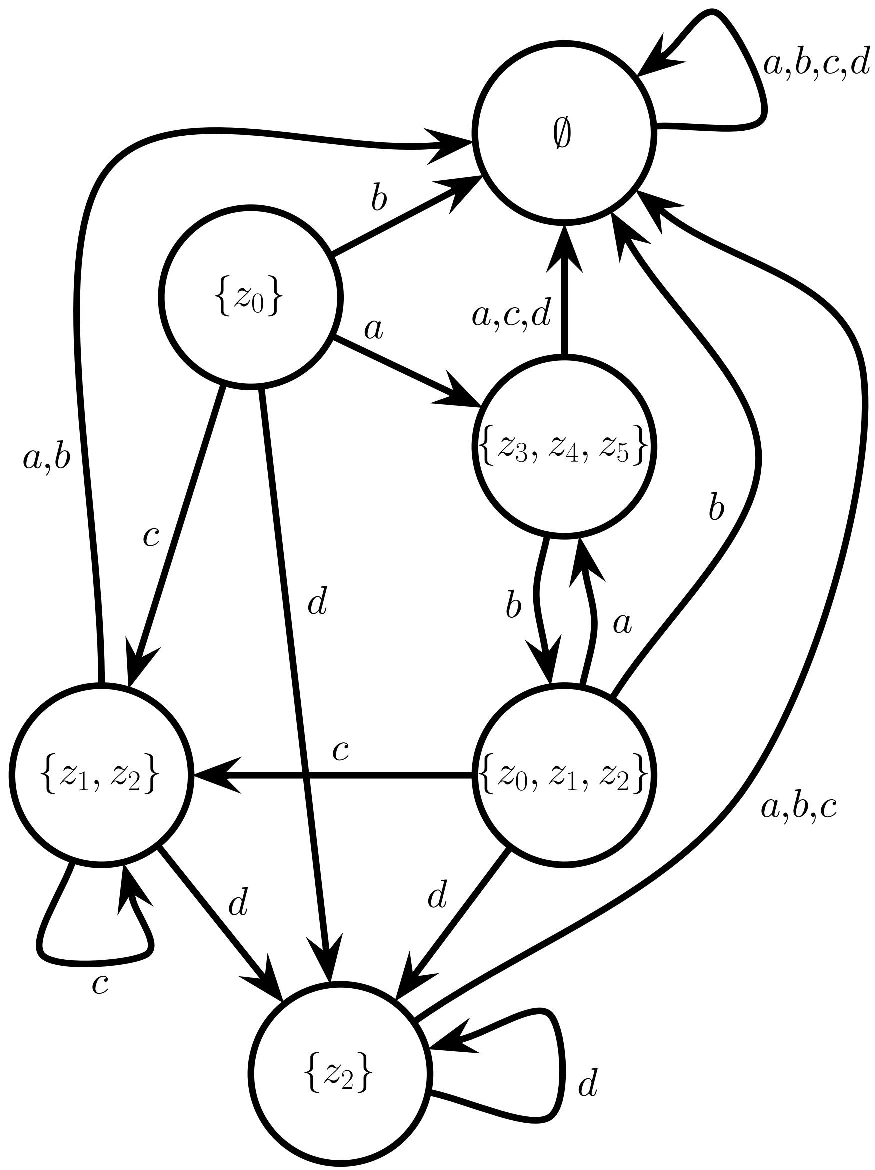



z_4
z_0
z_1
z_2
z_3
z_5
a
b
c
c,d
d
a
b
a
c,d
b

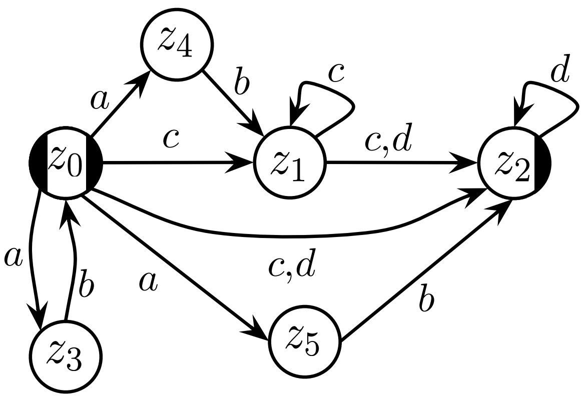

$\emptyset$
a,b,c,d
$\{z_0\}$
b
a
a,c,d
$\{z_3, z_4, z_5\}$
a,b
c
d
b
a
b
$\{z_1, z_2\}$
c
$\{z_0, z_1, z_2\}$
a,b,c
d
d
c
$\{z_2\}$

Potenzautomatenkonstruktion

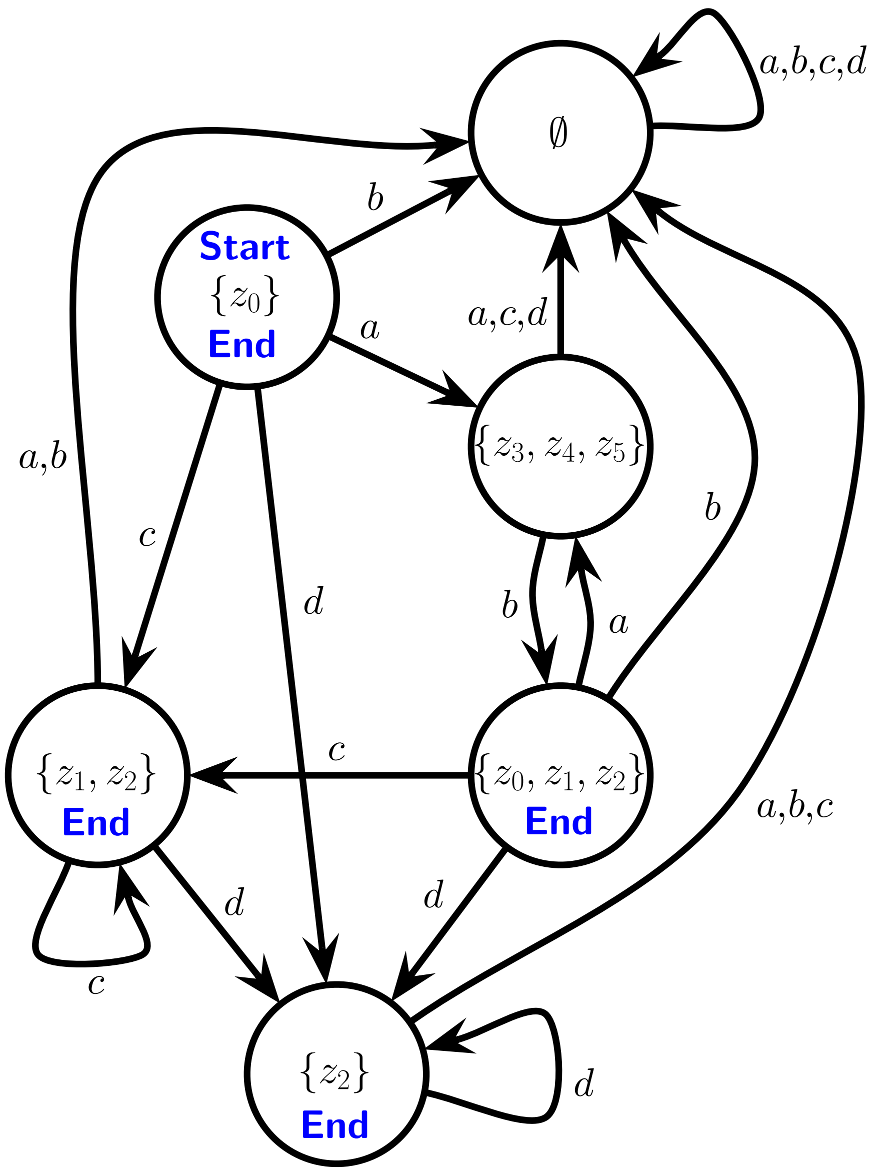




Potenzautomatenkonstruktion



$\equiv$



Potenzautomatenkonstr. (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein buchstabierender NFA.

Potenzautomatenkonstr. (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein buchstabierender NFA.
- Der **Potenzautomat** zu A ist wie folgt definiert:

Potenzautomatenkonstr. (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein buchstabierender NFA.
- Der **Potenzautomat** zu A ist wie folgt definiert:
  - $B := (2^Z, X, \delta, z_0, Z'_{\mathsf{end}})$ mit

Potenzautomatenkonstr. (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein buchstabierender NFA.
- Der **Potenzautomat** zu A ist wie folgt definiert:
  - $B := (2^Z, X, \delta, z_0, Z'_{\mathsf{end}})$ mit
  - $Z'_{\mathsf{end}} := \{M \in 2^Z \mid M \cap Z_{\mathsf{end}} \neq \emptyset\}$ und

Potenzautomatenkonstr. (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein buchstabierender NFA.
- Der **Potenzautomat** zu A ist wie folgt definiert:
  - $B := (2^Z, X, \delta, z_0, Z'_{\mathsf{end}})$ mit
  - $Z'_{\mathsf{end}} := \{M \in 2^Z \mid M \cap Z_{\mathsf{end}} \neq \emptyset\}$ und
  - $z_0 := Z_{\mathsf{start}}$, d.h. die Menge der Startzustände von A bildet nun den einzigen Startzustand z_0 von B.

Potenzautomatenkonstr. (formal)

- Sei $A := (Z, \Sigma, K, Z_{\mathsf{start}}, Z_{\mathsf{end}})$ ein buchstabierender NFA.
- Der **Potenzautomat** zu A ist wie folgt definiert:
  - $B := (2^Z, X, \delta, z_0, Z'_{\mathsf{end}})$ mit
  - $Z'_{\mathsf{end}} := \{M \in 2^Z \mid M \cap Z_{\mathsf{end}} \neq \emptyset\}$ und
  - $z_0 := Z_{\mathsf{start}}$, d.h. die Menge der Startzustände von A bildet nun den einzigen Startzustand z_0 von B.
  - δ ist für alle $M \in 2^Z$ und jedes $x \in \Sigma$ definiert durch:

$$\delta(M, x) := \bigcup_{z \in M} \{z' \in Z \mid (z, x, z') \in K\}.$$

Idee für Korrektheitsbeweis

- Einerseits: Zu jeder **akzeptierenden Rechnung** von B auf einem Wort w gibt es wegen der Definition von δ ebenfalls einen **Erfolgspfad** von A auf w.

Idee für Korrektheitsbeweis

- Einerseits: Zu jeder **akzeptierenden Rechnung** von B auf einem Wort w gibt es wegen der Definition von δ ebenfalls einen **Erfolgspfad** von A auf w.
- Andererseits: Für jeden **Erfolgspfad** von A existiert eine **akzeptierende Rechnung** in B.

Idee für Korrektheitsbeweis

- Einerseits: Zu jeder **akzeptierenden Rechnung** von B auf einem Wort w gibt es wegen der Definition von δ ebenfalls einen **Erfolgspfad** von A auf w.
- Andererseits: Für jeden **Erfolgspfad** von A existiert eine **akzeptierende Rechnung** in B.
  - Die in A besuchten Zustände kommen in den Zustandsnamen des Pfades von B in der gleichen Reihenfolge vor.

Idee für Korrektheitsbeweis

- Einerseits: Zu jeder **akzeptierenden Rechnung** von B auf einem Wort w gibt es wegen der Definition von δ ebenfalls einen **Erfolgspfad** von A auf w.
- Andererseits: Für jeden **Erfolgspfad** von A existiert eine **akzeptierende Rechnung** in B.
  - Die in A besuchten Zustände kommen in den Zustandsnamen des Pfades von B in der gleichen Reihenfolge vor.
- Somit ist der konstruierte vDFA B äquivalent zum NFA A.

Schlussfolgerung und Anwendung

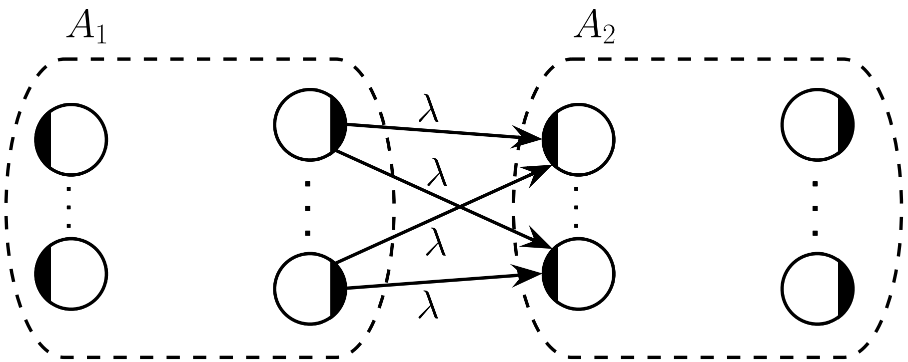
- NFAs akzeptieren dieselbe Sprachfamilie $\mathcal{R}eg$, wie DFAs.

Schlussfolgerung und Anwendung

- NFAs akzeptieren dieselbe Sprachfamilie $\mathcal{R}eg$, wie DFAs.
- Sind L_1 und L_2 reguläre Sprachen, dann ist auch $L_1 \cdot L_2$ regulär.

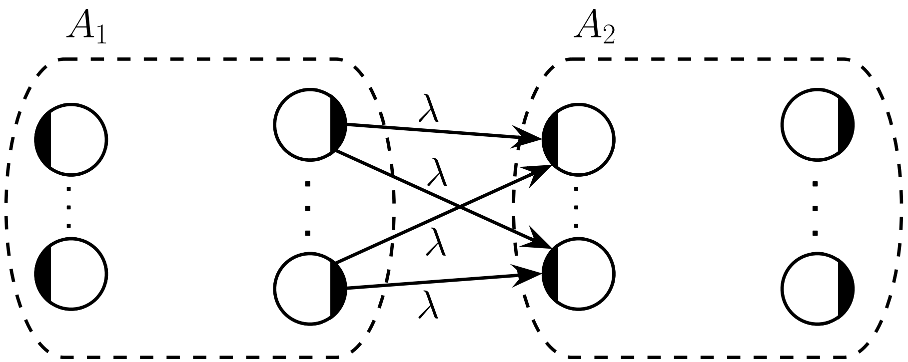
Schlussfolgerung und Anwendung

- NFAs akzeptieren dieselbe Sprachfamilie $\mathcal{R}eg$, wie DFAs.
- Sind L_1 und L_2 reguläre Sprachen, dann ist auch $L_1 \cdot L_2$ regulär.
- graphisch:



Schlussfolgerung und Anwendung

- NFAs akzeptieren dieselbe Sprachfamilie $\mathcal{R}eg$, wie DFAs.
- Sind L_1 und L_2 reguläre Sprachen, dann ist auch $L_1 \cdot L_2$ regulär.
- graphisch:



- formal:
$A = (Z_1 \uplus Z_2, \Sigma_1 \cup \Sigma_2, K_1 \cup K_2 \cup \{(z, \lambda, z') \mid z \in Z^{(1)}_{\text{end}} \wedge z' \in Z^{(2)}_{\text{start}}\}, Z^{(1)}_{\text{start}}, Z^{(2)}_{\text{end}}).$

Weitere Abschlusseigenschaften

- Vereinigung

Weitere Abschlusseigenschaften

- Vereinigung
- Durchnittsbildung

Weitere Abschlusseigenschaften

- Vereinigung
- Durchnittsbildung
- Komplement

Weitere Abschlusseigenschaften

- Vereinigung
- Durchnittsbildung
- Komplement
- Homomorphismen

Weitere Abschlusseigenschaften

- Vereinigung
- Durchnittsbildung
- Komplement
- Homomorphismen
  - Dass die regulären Sprachen gegen diese Operationen abgeschlossen sind ist nicht immer ganz so leicht zu zeigen ...

Weitere Abschlusseigenschaften

- Vereinigung
- Durchnittsbildung
- Komplement
- Homomorphismen
  - Dass die regulären Sprachen gegen diese Operationen abgeschlossen sind ist nicht immer ganz so leicht zu zeigen . . .
  - . . . deshalb verschieben wir die Beweis auf die folgenden Termine!

Weitere Abschlusseigenschaften

- Vereinigung
- Durchnittsbildung
- Komplement
- Homomorphismen
  - Dass die regulären Sprachen gegen diese Operationen abgeschlossen sind ist nicht immer ganz so leicht zu zeigen ...
  - ... deshalb verschieben wir die Beweis auf die folgenden Termine!