FGI 1

Organisatorisches

Michael Köhler-Bußmeier

Die Vortragenden

Prof. Dr. Christopher Habel
AB *Wissens- und Sprachverarbeitung (WSV)*
Haus F, 3. Stock, F-427

Vorlesungsteil: **Logik**

Vertr.-Prof. Dr. Michael Köhler-Bußmeier
AB *Theoretische Grundlagen der Informatik (TGI)*
Haus C, 1. Stock, Raum C-216

Vorlesungsteil: **Automaten und formale Sprachen**
sowie **Berechenbarkeit und Komplexität**

“Der Film zur Vorlesung”

“Achtung, Aufnahme!”

Schon bemerkt?

Die Vorlesung wurde aufgezeichnet.

Zu finden als Podcast im Portal:

http://www.lecture2go.uni-hamburg.de/

Gliederung der Vorlesung

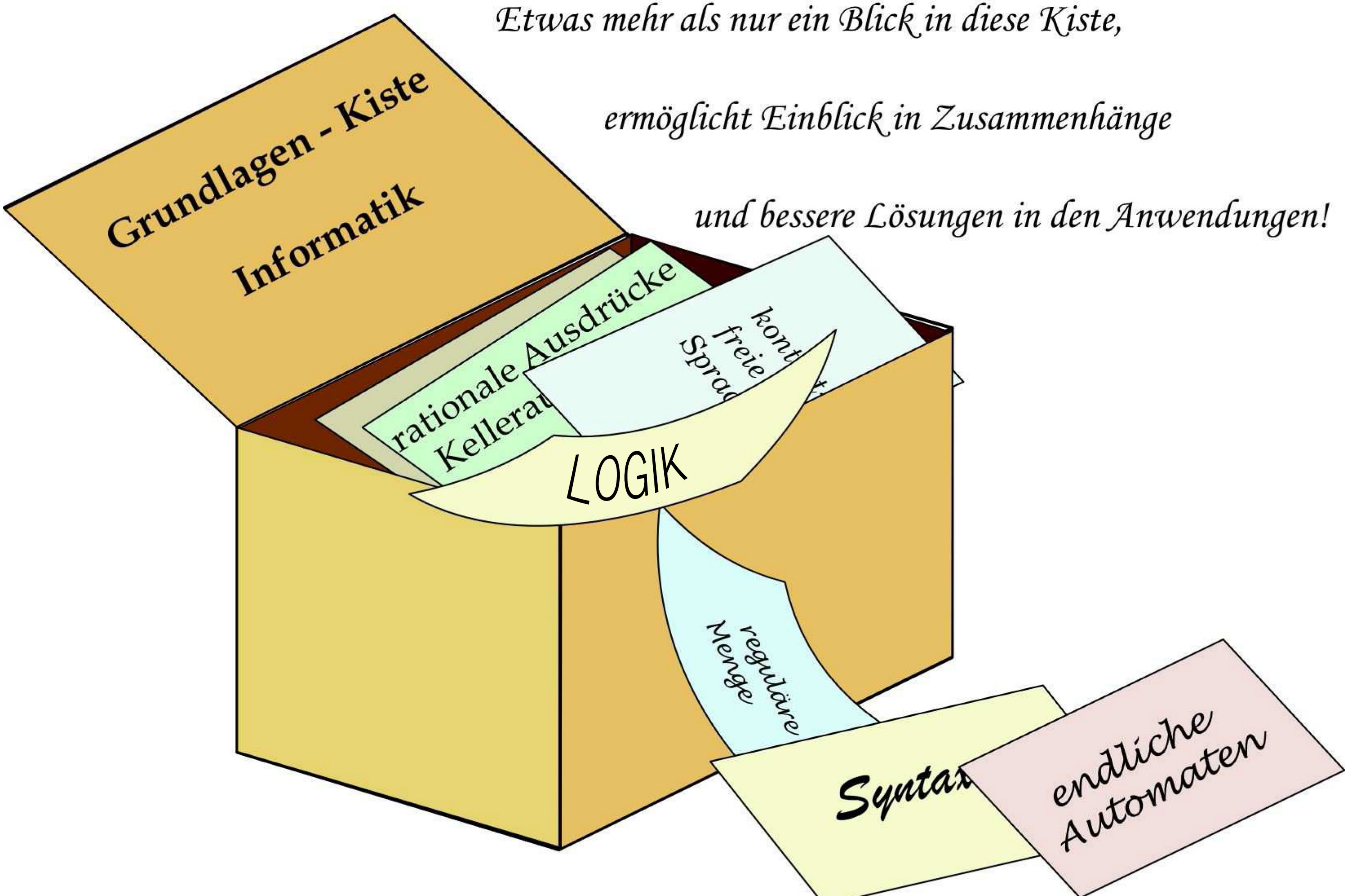
- Endliche Automaten
- Kontextfreie Grammatiken
- Aussagenlogik: Syntax und Semantik
- Prädikatenlogik
- Turingmaschinen und Berechenbarkeit
- Komplexitätstheorie

FGI 1

Kap. 12
Einführung & Motivation

Michael Köhler-Bußmeier

Grundlagenkiste



Warnung: Das “Kleingedruckte”

- Das Folien-Skript und die Vorlesung bilden eine Einheit.

- Das Skript erhebt **nicht** den Anspruch, alleinstehend verwendbar zu sein.

- Alleinstehend ist dagegen die Literatur:

-Hopcroft, John, Ulmann, Jeffrey, and Motwani, Rajeev (2002). Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. Addison-Wesley.

-Vossen, Gottfried & Witt, Kurt-Ulrich (2006). Grundkurs Theoretische Informatik. Vieweg Verlag.

-Sipser, Michael (2006) Introduction to the Theory of Computation, Thomson Course Technology.

-Schöning, Uwe (2000). Logik für Informatiker. Spektrum, Akademischer Verlag.

-Spies, Marcus (2003). Einführung in die Logik. Spektrum, Akademischer Verlag.

Motivation: Formale Sprachen

In der Informatik betrachten wir ständig Zeichenketten w, z.B. Binärworte über dem Zeichenvorrat $\Sigma = \{0, 1\}$. Beispielwort:

$$w = 101001001$$

Mit Σ^* bezeichnen wir die Menge aller Zeichenketten (beliebiger Länge).

Eine (endlich oder unendliche) Menge von Zeichenketten nennen wir *formale Sprache*.

Motivation: Formale Sprachen

In der Informatik betrachten wir ständig Zeichenketten w, z.B. Binärworte über dem Zeichenvorrat $\Sigma = \{0, 1\}$. Beispielwort:

$$w = 101001001$$

Mit Σ^* bezeichnen wir die Menge aller Zeichenketten (beliebiger Länge).

Eine (endlich oder unendliche) Menge von Zeichenketten nennen wir *formale Sprache*.

$$L = \{w \in \Sigma^* \mid \ w \text{ beginnt mit einer 1 und endet mit einer 0 } \}$$

Motivation: Formale Sprachen

In der Informatik betrachten wir ständig Zeichenketten w, z.B. Binärworte über dem Zeichenvorrat $\Sigma = \{0, 1\}$. Beispielwort:

$$w = 101001001$$

Mit Σ^* bezeichnen wir die Menge aller Zeichenketten (beliebiger Länge).

Eine (endlich oder unendliche) Menge von Zeichenketten nennen wir *formale Sprache*.

$$L = \{w \in \Sigma^* \mid \text{ } w \text{ beginnt mit einer 1 und endet mit einer 0 } \}$$

$$L_2 = \{w \in \Sigma^* \mid \text{ } w \text{ kodiert eine Primzahl binär } \}$$

Motivation: Formale Sprachen

In der Informatik betrachten wir ständig Zeichenketten w, z.B. Binärworte über dem Zeichenvorrat $\Sigma = \{0, 1\}$. Beispielwort:

$$w = 101001001$$

Mit Σ^* bezeichnen wir die Menge aller Zeichenketten (beliebiger Länge).

Eine (endlich oder unendliche) Menge von Zeichenketten nennen wir *formale Sprache*.

$$L = \{w \in \Sigma^* \mid \ w \text{ beginnt mit einer 1 und endet mit einer 0 } \}$$

$$L_2 = \{w \in \Sigma^* \mid \ w \text{ kodiert eine Primzahl binär } \}$$

$$L_3 = \{w \in \Sigma^* \mid \ w \text{ hat 0 und 1 in gleicher Anzahl } \}$$

Motivation: Formale Sprachen

In der Informatik betrachten wir ständig Zeichenketten w, z.B. Binärworte über dem Zeichenvorrat $\Sigma = \{0, 1\}$. Beispielwort:

$$w = 101001001$$

Mit Σ^* bezeichnen wir die Menge aller Zeichenketten (beliebiger Länge).

Eine (endlich oder unendliche) Menge von Zeichenketten nennen wir *formale Sprache*.

$$L = \{w \in \Sigma^* \mid \ w \text{ beginnt mit einer 1 und endet mit einer 0 } \}$$

$$L_2 = \{w \in \Sigma^* \mid \ w \text{ kodiert eine Primzahl binär } \}$$

$$L_3 = \{w \in \Sigma^* \mid \ w \text{ hat 0 und 1 in gleicher Anzahl } \}$$

$$L_4 = L_3 \cap L$$

Motivation: Formale Sprachen

In der Informatik betrachten wir ständig Zeichenketten w, z.B. Binärworte über dem Zeichenvorrat $\Sigma = \{0, 1\}$. Beispielwort:

$$w = 101001001$$

Mit Σ^* bezeichnen wir die Menge aller Zeichenketten (beliebiger Länge).

Eine (endlich oder unendliche) Menge von Zeichenketten nennen wir *formale Sprache*.

$$L = \{w \in \Sigma^* \mid \ w \text{ beginnt mit einer 1 und endet mit einer 0 } \}$$

$$L_2 = \{w \in \Sigma^* \mid \ w \text{ kodiert eine Primzahl binär } \}$$

$$L_3 = \{w \in \Sigma^* \mid \ w \text{ hat 0 und 1 in gleicher Anzahl } \}$$

$$L_4 = L_3 \cap L$$

Welche der Sprachen ist "einfacher"?

Motivation: Formale Sprachen

Welche Eigenschaften interessieren an einer formalen Sprache $L \subseteq \Sigma^*$?

- Wortproblem: Gilt $w \in L$ für ein gegebenes Wort w und eine gegebene formale Sprache L?
- Inklusion: Gilt $L_1 \subseteq L_2$ für gegebene formale Sprachen L_1, L_2?
- Leerheit: Gilt $L = \emptyset$ für eine gegebene formale Sprache L?
- Universalität: Gilt $L = \Sigma^*$ für eine gegebene formale Sprache L?
- ....

Motivation: Sprachfamilien

Wir wollen nicht für jede Sprache L ein spezifisches Entscheidungsverfahren entwickeln.

Daher fassen wir Sprachen, die sich gleichartig charakterisieren lassen, zu *Sprachfamilien* $\mathcal{F}$ *(= Mengen von Sprachen)* zusammen.

Motivation: Sprachfamilien

Wir wollen nicht für jede Sprache L ein spezifisches Entscheidungsverfahren entwickeln.

Daher fassen wir Sprachen, die sich gleichartig charakterisieren lassen, zu *Sprachfamilien* $\mathcal{F}$ *(= Mengen von Sprachen)* zusammen.

Sprachfamilie	Automaten	Grammatik
endl. Mengen		
$\mathcal{R}eg$		
$det\mathcal{C}f$		
$\mathcal{L}_2 = \mathcal{C}f$		
$\mathcal{L}_1 = \mathcal{C}s$		
$\mathcal{R}ec$		
$\mathcal{L}_0 = \mathcal{R}e$		
abzählbare Mengen		

Motivation: Sprachfamilien

Wir wollen nicht für jede Sprache L ein spezifisches Entscheidungsverfahren entwickeln.

Daher fassen wir Sprachen, die sich gleichartig charakterisieren lassen, zu *Sprachfamilien* $\mathcal{F}$ *(= Mengen von Sprachen)* zusammen.

Sprachfamilie	Automaten	Grammatik
endl. Mengen	–	–
$\mathcal{R}eg$	DFA=NFA	Typ-3 = rechts-lineare G.
$det\mathcal{C}f$	DPDA	$LR(k)$, $k \geq 1$
$\mathcal{L}_2 = \mathcal{C}f$	PDA	Typ-2 = kontextfreie G.
$\mathcal{L}_1 = \mathcal{C}s$	NLBA	Typ-1 = monotone G.
$\mathcal{R}ec$	terminierende TM	–
$\mathcal{L}_0 = \mathcal{R}e$	DTM = NTM	Typ-0
abzählbare Mengen	-	-

Motivation: Sprachfamilien

Wir wollen nicht für jede Sprache L ein spezifisches Entscheidungsverfahren entwickeln.

Daher fassen wir Sprachen, die sich gleichartig charakterisieren lassen, zu *Sprachfamilien* $\mathcal{F}$ *(= Mengen von Sprachen)* zusammen.

Sprachfamilie	Automaten	Grammatik	Beis
endl. Mengen	–	–	$\{a, ab$
$\mathcal{R}eg$	DFA=NFA	Typ-3 = rechts-lineare G.	$\{a\}^*$
$det\mathcal{C}f$	DPDA	$LR(k)$, $k \geq 1$	$\{a^n b^n \mid$ $\{wcw^{rev} \mid w$
$\mathcal{L}_2 = \mathcal{C}f$	PDA	Typ-2 = kontextfreie G.	$\{ww^{rev} \mid w$
$\mathcal{L}_1 = \mathcal{C}s$	NLBA	Typ-1 = monotone G.	$\{a^n b^n c^n$
$\mathcal{R}ec$	terminierende TM	–	L
$\mathcal{L}_0 = \mathcal{R}e$	DTM = NTM	Typ-0	H, $(G^*$
abzählbare Mengen	-	-	L

Motivation: Sprachfamilien

Sprachfamilie	Automaten	Grammatik	Beispiel
endl. Mengen	–	–	$\{a, ab, abb\}$
$\mathcal{R}eg$	DFA=NFA	Typ-3 = rechts-lineare G.	$\{a\}^*\{b\}^*$
$det\mathcal{C}f$	DPDA	$LR(k)$, $k \geq 1$	$\{a^n b^n \mid n \in \mathbb{N}\}$ $\{wcw^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_2 = \mathcal{C}f$	PDA	Typ-2 = kontextfreie G.	$\{ww^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_1 = \mathcal{C}s$	NLBA	Typ-1 = monotone G.	$\{a^n b^n c^n \mid n \in \mathbb{N}\}$
$\mathcal{R}ec$	terminierende TM	–	L_e
$\mathcal{L}_0 = \mathcal{R}e$	DTM = NTM	Typ-0	H, $(G^* \setminus L_d)$
abzählbare Mengen	-	-	L_d

Ziel ist es, zu jeder Familie $\mathcal{F}$ jeweils ein allgemeines Verfahren zu entwickeln, das auf jede beliebige Sprachen L aus $\mathcal{F}$ angewendet werden kann.

Der Berechnungsaufwand wird steigen, wenn die Charakterisierung „mächtiger“ wird.

Für einige Sprachfamilien, sind manche Probleme schon nicht mehr mit *vertretbarem* Aufwand lösbar.

Es gibt sogar Sprachfamilien, die sind so mächtig, dass es für manche der Probleme *kein* algorithmisches Verfahren geben kann!

Motivation: Sprachfamilien

Sprachfamilie	Automaten	Grammatik	Beispiel
endl. Mengen	–	–	$\{a, ab, abb\}$
$\mathcal{R}eg$	DFA=NFA	Typ-3 = rechts-lineare G.	$\{a\}^*\{b\}^*$
$det\mathcal{C}f$	DPDA	$LR(k)$, $k \geq 1$	$\{a^n b^n \mid n \in \mathbb{N}\}$ $\{wcw^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_2 = \mathcal{C}f$	PDA	Typ-2 = kontextfreie G.	$\{ww^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_1 = \mathcal{C}s$	NLBA	Typ-1 = monotone G.	$\{a^n b^n c^n \mid n \in \mathbb{N}\}$
$\mathcal{R}ec$	terminierende TM	–	L_e
$\mathcal{L}_0 = \mathcal{R}e$	DTM = NTM	Typ-0	H, $(G^* \setminus L_d)$
abzählbare Mengen	-	-	L_d

Ziel ist es, zu jeder Familie $\mathcal{F}$ jeweils ein allgemeines Verfahren zu entwickeln, das auf jede beliebige Sprachen L aus $\mathcal{F}$ angewendet werden kann.

Der Berechnungsaufwand wird steigen, wenn die Charakterisierung „mächtiger“ wird.

Für einige Sprachfamilien, sind manche Probleme schon nicht mehr mit *vertretbarem* Aufwand lösbar.

Es gibt sogar Sprachfamilien, die sind so mächtig, dass es für manche der Probleme *kein* algorithmisches Verfahren geben kann!

Motivation: Sprachfamilien

Sprachfamilie	Automaten	Grammatik	Beispiel
endl. Mengen	–	–	$\{a, ab, abb\}$
$\mathcal{R}eg$	DFA=NFA	Typ-3 = rechts-lineare G.	$\{a\}^*\{b\}^*$
$det\mathcal{C}f$	DPDA	$LR(k)$, $k \geq 1$	$\{a^n b^n \mid n \in \mathbb{N}\}$ $\{wcw^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_2 = \mathcal{C}f$	PDA	Typ-2 = kontextfreie G.	$\{ww^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_1 = \mathcal{C}s$	NLBA	Typ-1 = monotone G.	$\{a^n b^n c^n \mid n \in \mathbb{N}\}$
$\mathcal{R}ec$	terminierende TM	–	L_e
$\mathcal{L}_0 = \mathcal{R}e$	DTM = NTM	Typ-0	H, $(G^* \setminus L_d)$
abzählbare Mengen	-	-	L_d

Ziel ist es, zu jeder Familie $\mathcal{F}$ jeweils ein allgemeines Verfahren zu entwickeln, das auf jede beliebige Sprachen L aus $\mathcal{F}$ angewendet werden kann.

Der Berechnungsaufwand wird steigen, wenn die Charakterisierung „mächtiger“ wird.

Für einige Sprachfamilien, sind manche Probleme schon nicht mehr mit *vertretbarem* Aufwand lösbar.

Es gibt sogar Sprachfamilien, die sind so mächtig, dass es für manche der Probleme *kein* algorithmisches Verfahren geben kann!

Motivation: Sprachfamilien

Sprachfamilie	Automaten	Grammatik	Beispiel
endl. Mengen	–	–	$\{a, ab, abb\}$
$\mathcal{R}eg$	DFA=NFA	Typ-3 = rechts-lineare G.	$\{a\}^*\{b\}^*$
$det\mathcal{C}f$	DPDA	$LR(k)$, $k \geq 1$	$\{a^n b^n \mid n \in \mathbb{N}\}$ $\{wcw^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_2 = \mathcal{C}f$	PDA	Typ-2 = kontextfreie G.	$\{ww^{rev} \mid w \in \{a,b\}^*\}$
$\mathcal{L}_1 = \mathcal{C}s$	NLBA	Typ-1 = monotone G.	$\{a^n b^n c^n \mid n \in \mathbb{N}\}$
$\mathcal{R}ec$	terminierende TM	–	L_e
$\mathcal{L}_0 = \mathcal{R}e$	DTM = NTM	Typ-0	H, $(G^* \setminus L_d)$
abzählbare Mengen	-	-	L_d

Ziel ist es, zu jeder Familie $\mathcal{F}$ jeweils ein allgemeines Verfahren zu entwickeln, das auf jede beliebige Sprachen L aus $\mathcal{F}$ angewendet werden kann.

Der Berechnungsaufwand wird steigen, wenn die Charakterisierung „mächtiger" wird.

Für einige Sprachfamilien, sind manche Probleme schon nicht mehr mit *vertretbarem* Aufwand lösbar.

Es gibt sogar Sprachfamilien, die sind so mächtig, dass es für manche der Probleme *kein* algorithmisches Verfahren geben kann!

Ziele des Automatenteils:

- Definitionen und Umgang mit **formalen Sprachen.**
- einfache **Automatenmodelle**
- **Grammatiken**
- Der Begriff der „**Berechenbarkeit**“:
- Gibt es etwas, das nicht berechenbar ist?
- Probleme, deren Lösung beweisbar **komplex** ist.

FGI 1

Notationen

Michael Köhler-Bußmeier

Aussagenlogik (informale Einführung)

Seien A und B Aussagen (die entweder wahr oder falsch sein können).

- $A \wedge B$: Sowohl A als auch B sind wahr
- $A \vee B$: Entweder A oder B ist wahr – oder beide.
- $A \Rightarrow B$: Wenn A wahr ist, dann auch B
- $A \iff B$: Wenn A wahr ist, dann auch B – und umgekehrt.
- $\forall x.A$: Für alle x gilt A.
  (Wobei der Wahrheitswert von A von x abhängen kann).
- $\exists x.A$: Für mindestens ein x gilt A
  (Wobei der Wahrheitswert von A von x abhängen kann).

Mengen

Eine Menge ist eine *ungeordnete* Ansammlung.

- Aufzählung endlicher Mengen: $\{1, 2, 3, 4\}$
- Unendliche Mengen: $\{1, 2, 3, \ldots\}$
- Charakterisierung: $\{a \in A \mid \phi(a)\}$

  Bsp.: $\{a \mid a \text{ ist gerade }\}$
- $a \in A$: a ist Element von A

Vergleich von Mengen:

- Inklusion $A \subseteq B$ gilt, wenn für alle $a \in A$ auch $a \in B$ folgt.

  Sprechweise: A ist Teilmenge von B.
- Mengengleichheit $A = B$ gilt, wenn sowohl $A \subseteq B$ als auch $B \subseteq A$ gilt.
- Strenge Inklusion $A \subset B$ gilt, wenn für alle $a \in A$ auch $a \in B$ gilt, aber nicht $A = B$ gilt.

  Sprechweise: A ist echte Teilmenge von B.

Mengenoperationen

Operationen auf Mengen:

- Vereinigung $A \cup B := \{x \mid x \in A \text{ oder } x \in B\}$
- Schnitt $A \cap B := \{x \mid x \in A \text{ und } x \in B\}$
- Komplement $A \setminus B := \{x \in A \mid x \in A \text{ und } x \notin B\}$
- Potenzmenge $\mathcal{P}(A) = \{B \mid B \subseteq A\}$
- Kartesisches Produkt $A \times B := \{(a, b) \mid a \in A, b \in B\}$
- Potenzen: Menge aller Paare: $A^2 := A \times A$,
  Menge aller Tripel $A^3 := A \times A \times A$ usw.
- Kardinalität $|A|$ ist die Anzahl der Elemente der Menge A.

Potenzmenge

Satz

Sei A eine beliebige Menge. Die Relation $\subseteq$ ist eine partielle Ordnung auf der Potenzmenge $\mathcal{P}(A)$.
Sie ist i.a. keine totale Ordnung.

(Beweis?)

Satz

Sei A eine beliebige endliche Menge. Die Kardinalität der Potenzmenge $\mathcal{P}(A)$ ergibt sich als:

$$|\mathcal{P}(A)| = 2^{|A|}$$

(Beweis?)

Satz

Sei A eine beliebige endliche Menge der Kardinalität $n = |A|$.
Die Anzahl aller Teilmengen der Kardinalität $k \leq n$ ist $\binom{n}{k}$.

Cartesisches Produkt

Definition 12.1:

- Seien $A_1, A_2, \ldots, A_n$ Mengen und $x_1 \in A_1, \ldots, x_n \in A_n$, dann heißt $(x_1, \ldots, x_n)$ ein **(geordnetes) n-Tupel** von Elementen über $A_1, \ldots, A_n$.
- Die Menge aller geordneten n-Tupel $(x_1, \ldots, x_n)$ mit $x_1 \in A_1, \ldots, x_n \in A_n$ heißt **cartesisches Produkt** der Mengen A_1 bis A_n und wird geschrieben als $A_1 \times A_2 \ldots \times A_n$.

- Es ist also $A_1 \times A_2 \times \ldots \times A_n :=$
$$\{(x_1, \ldots, x_n) \mid x_1 \in A_1, \ldots, x_n \in A_n\}.$$

Relationen

Definition 12.4:

- Eine Teilmenge $R \subseteq A_1 \times A_2 \times \ldots \times A_n$ heißt n**-stellige Relation** über A_1 bis A_n.
- Ist $n = 2$, so spricht man von einer **binären** statt von einer 2-stelligen **Relation**.

Beispiel: Sei A_1 = Menge aller Dozenten
und A_2 = Menge aller Studenten.

Dann ist

$$R_{FGI} = \{Habel, Köhler\} \times \{ s \in A_2 \mid s \text{ besucht } FGI\text{-}1\}$$

eine Relation auf $A_1 \times A_2$.

Eigenschaften von Relationen

Sei $R \subseteq A \times A$ eine Relation auf A. R heißt

- **reflexiv** gdw. $(a, a) \in R$ für alle $a \in A$ gilt.
- **symmetrisch** gdw. für $(a, b) \in R$ stets auch $(b, a) \in R$ gilt.
- **transitiv** gdw. aus $(a, b) \in R$ und $(b, c) \in R$ stets $(a, c) \in R$ folgt. R ist also transitiv, wenn $R \cdot R \subseteq R$ gilt.

Eine Relation heißt **Äquivalenzrelation** gdw. sie **reflexiv**, **symmetrisch** und **transitiv** ist.

Transitiver Abschluss einer binären Relation

Definition 12.5:

- Sei $R \subseteq A \times A$ eine Relation auf der Menge A.
  Dann seien R^+, der **transitive Abschluss** (auch **transitive Hülle**), und R^*, der **reflexive, transitive Abschluss**, von R wie folgt erklärt:
  $$R^+ := \bigcup_{i \geq 1} R_i \text{ mit } R_1 := R \qquad \text{und} \qquad R_{i+1} := R_i \cdot R.$$
  sowie
  $$R^* := R^+ \cup Id_A$$
- Für Relationen R_1 und R_2 ist $R_1 \cdot R_2 := \{(x,z) \mid \exists y : (x,y) \in R_1 \land (y,z) \in R_2\}$ das **Relationenprodukt** oder die **Komposition**. Wie auch beim Komplexprodukt, wird der Punkt " $\cdot$ " hier oft weggelassen.

Beweistechniken

Die meisten Beweise sind für Aussagen, die in WENN-DANN-Form vorliegen:

Wenn *A* gilt, dann ist *B*.

Elementare Beweismuster sind:

- Direkter Beweis
- Indirekter Beweis
- Beweis durch Widerspruch

Betrachten wir einige einfache Beispiele.

Direkter Beweis einer Wenn-Dann Aussage

$$\frac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{(A \Rightarrow B)}$$

Satz

Wenn $m, n \in \mathbb{Z}$ beide ungerade sind, dann ist $m + n$ gerade.

Beweis: Zu zeigen ist $A \Rightarrow B$ mit:

Aussage A	=	Die Zahlen $m, n \in \mathbb{Z}$ sind beide ungerade.
Aussage B	=	Die Zahl $m + n$ ist gerade.

- Wir nehmen A an.
- Also gibt es Zahlen $k, l \in \mathbb{Z}$, so dass $m = 2k + 1$ und $n = 2l + 1$ gilt.
- Dann folgt: $m + n = (2k + 1) + (2l + 1) = 2(k + l + 1)$.
- Also ist $m + n$ gerade (Aussage B).
- Also konnte unter der Annahme von A auch B gezeigt werden.

Also gilt nach der obigen Schlussregel $A \Rightarrow B$. q.e.d.

Indirekter Beweis einer Wenn-Dann Aussage

$$\frac{A \Rightarrow B}{\neg B \Rightarrow \neg A}$$

Satz

Wenn $n \in \mathbb{Z}$ eine Zahl ist, für die n^2 eine gerade Zahl ist, dann ist auch n eine gerade Zahl.

Beweis: Zu zeigen ist $A \Rightarrow B$ mit:

Aussage A	=	Die Zahl n^2 ist gerade ($n \in \mathbb{Z}$).
Aussage B	=	Die Zahl n ist gerade.

- *Annahme des Gegenteils, d.h. von $\neg B$:* Sei $n \in \mathbb{Z}$ ungerade.
- Also gibt es ein $k \in \mathbb{Z}$, so dass $n = 2k + 1$ gilt.
- Dann folgt: $n^2 = (2k + 1)^2 = 4k^2 + 2k + 1 = 2(2k^2 + k) + 1$.
- Also ist n^2 ungerade.
- Also gilt die Aussage A gerade nicht.
- Damit konnte unter der Annahme von $\neg B$ auch $\neg A$ gezeigt werden.
- Also gilt $\neg B \Rightarrow \neg A$.

Also gilt nach der obigen Schlussregel $A \Rightarrow B$. q.e.d.

Widerspruchsbeweis einer Wenn-Dann Aussage

Satz

Wenn $a \in \mathbb{R}^+$ eine Lösung von $x^2 - 2 = 0$ ist, dann ist a irrational.

Beweis: Zu zeigen ist $A \Rightarrow B$ mit:

Aussage A	=	a ist Lösung von $x^2 - 2 = 0$ und $a \in \mathbb{R}^+$.
Aussage B	=	a ist irrational.

$$\frac{(A \wedge \neg B) \lightning}{A \Rightarrow B}$$

- Annahme $A \wedge \neg B$: Also ist a rational und $a^2 = 2$.
- Also gibt es $m, n \in \mathbb{N}, n \neq 0$ mit $a = \frac{m}{n}$.
  Hierbei können wir voraussetzen, dass $\frac{m}{n}$ gekürzt dargestellt wird.
- Es gilt: $2 = a^2 = (\frac{m}{n})^2 = \frac{m^2}{n^2} \iff m^2 = 2n^2$.
- Also ist m^2 eine gerade Zahl.
- Dies kann aber nur sein, wenn auch m eine gerade Zahl ist: $m = 2k$.
- Eingesetzt ergibt sich: $2n^2 = m^2 = (2k)^2 = 4k^2 \iff n^2 = 2k^2$
- Also ist auch n^2 gerade und damit auch n, d.h. $n = 2l$.
- Damit liegt aber $a = \frac{m}{n} = \frac{2k}{2l}$ nicht gekürzt vor. Widerspruch!
- Also muss die Annahme $\neg B$ (d.h. dass $\sqrt{2}$ rational ist), falsch sein.

Also gilt nach der obigen Schlussregel $A \Rightarrow B$. q.e.d.

FGI 1

Wörter, Sprachen

Michael Köhler-Bußmeier

Wortmengen

Definition 12.7:

- Ein **Alphabet** ist eine (total geordnete) endliche Menge von unterschiedlichen **Zeichen** (oder **Symbolen**).

- Für ein Alphabet Σ ist Σ^*, als Kurzform für $(\Sigma^*, \circ)$, das **freie Monoid** mit der **Konkatenation** $(\circ)$ oder Hintereinanderschreibung der einzelnen Zeichen als Monoid-Operation, und dem leeren Wort ϵ (Epsilon) [andernorts auch λ (Lambda)] als neutralem Element. Für $w \in \Sigma^*$ schreiben wir w^k anstelle von $\underbrace{ww \cdots w}_{k}$. Es sei $w^0 := \epsilon$.

  Jede Menge $L \subseteq \Sigma^*$ heißt **formale Sprache**.

- Σ^* bezeichnet also die **Menge aller endlichen Wörter über dem Alphabet Σ**

Beispiele

Beispiele

- Sei $\Sigma := \{A, B, C, 1, 2, 3\}$, dann gilt bspw.:

Beispiele

- Sei $\Sigma := \{A, B, C, 1, 2, 3\}$, dann gilt bspw.:

$$\varepsilon \in \Sigma^* \quad \text{und} \quad C3 \in \Sigma^*$$

aber: $C4 \notin \Sigma^*$.

Beispiele

- Sei $\Sigma := \{A, B, C, 1, 2, 3\}$, dann gilt bspw.:

$$\varepsilon \in \Sigma^* \quad \text{und} \quad C3 \in \Sigma^*$$

$$\text{aber:} \quad C4 \notin \Sigma^*.$$

- Sei $M := \{HAUS, BOOT, TOR\} \subseteq \Sigma^*$, dann sind z.B.

Wörter aus $M^+ \subset M^*$:

Beispiele

- Sei $\Sigma := \{A, B, C, 1, 2, 3\}$, dann gilt bspw.:

$$\varepsilon \in \Sigma^* \quad \text{und} \quad C3 \in \Sigma^*$$

$$\text{aber:} \quad C4 \notin \Sigma^*.$$

- Sei $M := \{HAUS, BOOT, TOR\} \subseteq \Sigma^*$, dann sind z.B.

Wörter aus $M^+ \subset M^*$:

$$HAUSBOOT, \quad TORHAUS \quad \text{und} \quad HAUSTOR$$

Beispiele

- Sei $\Sigma := \{A, B, C, 1, 2, 3\}$, dann gilt bspw.:

$\varepsilon \in \Sigma^*$ und $C3 \in \Sigma^*$

aber: $C4 \notin \Sigma^*$.

- Sei $M := \{HAUS, BOOT, TOR\} \subseteq \Sigma^*$, dann sind z.B.

Wörter aus $M^+ \subset M^*$:

HAUSBOOT, *TORHAUS* und *HAUSTOR*

- *TOR* ist Präfix von *TORHAUS* und

BOOT ist Suffix von *HAUSBOOT*.

b-näre Zahlendarstellung (Wiederholung)

Gelegentlich interpretieren wir Wörter als

Kodierung, z.B. von Zahlen:

Definition 12.9:

- Die b-**näre Darstellung** benutzt die Symbole der Ziffernmenge $B := \{0, 1, \ldots, b-1\}$. Ist $b = 2$, so spricht man von einer **binären** Zahlendarstellung.

- Eine Zahl $n \in \mathbb{N}$ wird zur Basis $b = |B|$ durch eine Folge $a_k a_{k-1} \cdots a_0$ von Symbolen $a_i \in \{0, 1, \ldots, b-1\}$ dargestellt, wenn $n = \sum_{i=0}^{k} a_i \cdot b^i$ gilt.

  Wir notieren dies durch $[a_k a_{k-1} \cdots a_0]_b = n$.

b-adische Zahlendarstellung

Alternative Interpretation (ohne "0"):

Definition 12.8:

- Die b-**adische Darstellung** benutzt die Symbole der Ziffernmenge $B := \{1, 2, \ldots, b\}$. Ist $b = 2$, so spricht man von einer **dyadischen** Zahlendarstellung.

- Eine natürliche Zahl $n \in \mathbb{N}, n \neq 0$ wird zur Basis $b = |B|$ durch eine Folge $a_k a_{k-1} \cdots a_0$ von Symbolen $a_i \in \{1, \ldots, b\}$ notiert, wenn $n = \sum_{i=0}^{k} a_i \cdot b^i$ gilt.

  Wir notieren dann $[a_k a_{k-1} \cdots a_0]_{b-adic} = n$.

Halbgruppe, Monoid

Definition 12.2:

- Ist H eine Menge und $\odot : H \times H \longrightarrow H$ eine Abbildung für die das *Assoziativgesetz* gilt, d.h.
  $$\forall a, b, c \in H : \ (a \odot b) \odot c = a \odot (b \odot c),$$
  dann heißt $(H, \odot)$ **Halbgruppe**.

- Eine Halbgruppe $(H, \odot)$ heißt **Monoid** genau dann, wenn es ein **neutrales Element** $e_H \in H$ gibt, so dass $e_H \odot m = m \odot e_H = m$ für jedes $m \in H$ gilt.

- **Beispiel:** $(\mathbb{N} \setminus \{0\}, +)$ ist eine Halbgruppe, während $(\mathbb{N}, +)$ ein Monoid mit der Null als neutralem Element ist.

Halbgruppe, Monoid

Definition 12.2:

- Ist H eine Menge und $\odot : H \times H \longrightarrow H$ eine Abbildung für die das *Assoziativgesetz* gilt, d.h.
  $$\forall a, b, c \in H : \; (a \odot b) \odot c = a \odot (b \odot c),$$
  dann heißt $(H, \odot)$ **Halbgruppe**.

- Eine Halbgruppe $(H, \odot)$ heißt **Monoid** genau dann, wenn es ein **neutrales Element** $e_H \in H$ gibt, so dass $e_H \odot m = m \odot e_H = m$ für jedes $m \in H$ gilt.

- **Beispiel:** $(\mathbb{N} \setminus \{0\}, +)$ ist eine Halbgruppe, während $(\mathbb{N}, +)$ ein Monoid mit der Null als neutralem Element ist.

Bsp.: Binäre Relationen mit Komposition als Operation und der Identitätsrelation als neutralem Element bilden auch einen Monoid.

Komplexprodukt

Definition 12.3:

- Für Teilmengen U, V einer Halbgruppe oder eines Monoides $(H, \odot)$ bezeichnet die Menge

$$U \cdot V := \{u \odot v \mid u \in U, v \in V\}$$

das **Komplexprodukt** von U und V.

- Mit dem Komplexprodukt zweier Teilmengen einer Halbgruppe, wird die Halbgruppenoperation auf die elementweise Verknüpfung übernommen!

- **Beispiel:** Für Mengen von Wörtern ist die verwendete Operation die Konkatenation, so dass $U \cdot V$ alle Wörter enthält, die sich durch Hintereinanderschreiben eines Wortes aus U und eines Wortes aus V ergeben!

Transitiver Abschluss bei einer Halbgruppe

Definition 12.6:

- Für eine Teilmenge $M \subseteq H$ einer Halbgruppe $(H, \odot)$ seien M^+ die von M erzeugte **Unterhalbgruppe** sowie M^* das von M erzeugte **Untermonoid**.

$$M^+ := \bigcup_{i \geq 1} M^i \text{ mit } M^1 := M \text{ und } M^{i+1} := M^i \cdot M$$

$$M^* := M^+ \cup \{e_H\}$$

Beispiel:

Sei $A := \{2, 5\} \subseteq (\mathbb{N}, +)$ dann gilt

$$A^+ = \mathbb{N} \setminus \{0, 1, 3\}$$

und $$A^* = \mathbb{N} \setminus \{1, 3\},$$

denn in $(\mathbb{N}, +)$ ist $e_H = 0$.

Transitiver Abschluss bei einer Halbgruppe

Definition 12.6:

- Für eine Teilmenge $M \subseteq H$ einer Halbgruppe $(H, \odot)$ seien M^+ die von M erzeugte **Unterhalbgruppe** sowie M^* das von M erzeugte **Untermonoid**.

$$M^+ := \bigcup_{i \geq 1} M^i \text{ mit } M^1 := M \text{ und } M^{i+1} := M^i \cdot M$$

$$M^* := M^+ \cup \{e_H\}$$

M^i ist die Menge der Produkte mit i Faktoren.

Beispiel:

Sei $A := \{2, 5\} \subseteq (\mathbb{N}, +)$ dann gilt

$$A^+ = \mathbb{N} \setminus \{0, 1, 3\}$$

und $$A^* = \mathbb{N} \setminus \{1, 3\},$$

denn in $(\mathbb{N}, +)$ ist $e_H = 0$.

FGI 1

Automaten, Formale Sprachen, Berechenbarkeit

Kap. 13

Endlicher Automat, Potenzautomat

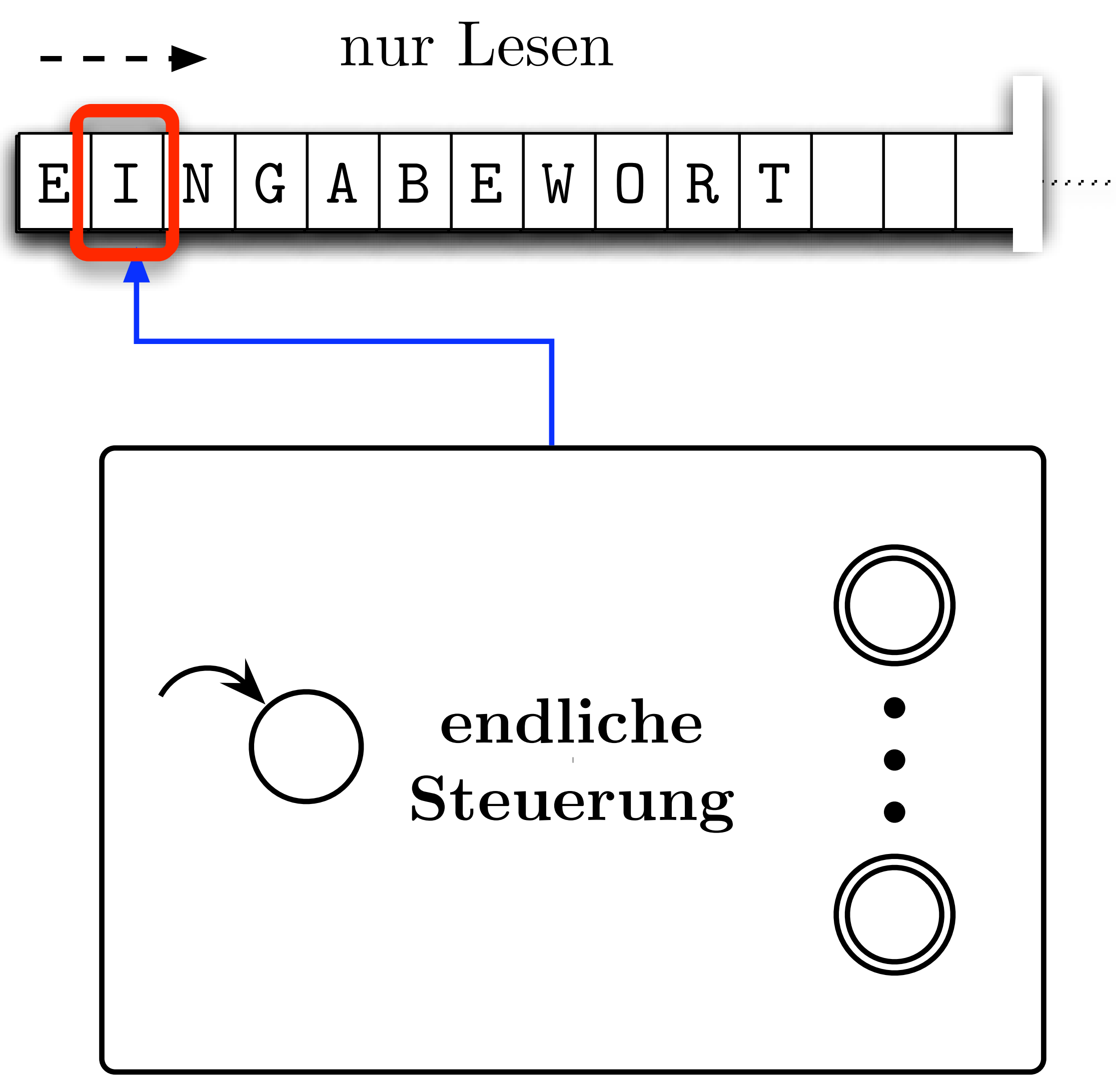
Michael Köhler-Bußmeier

Formale Sprachen

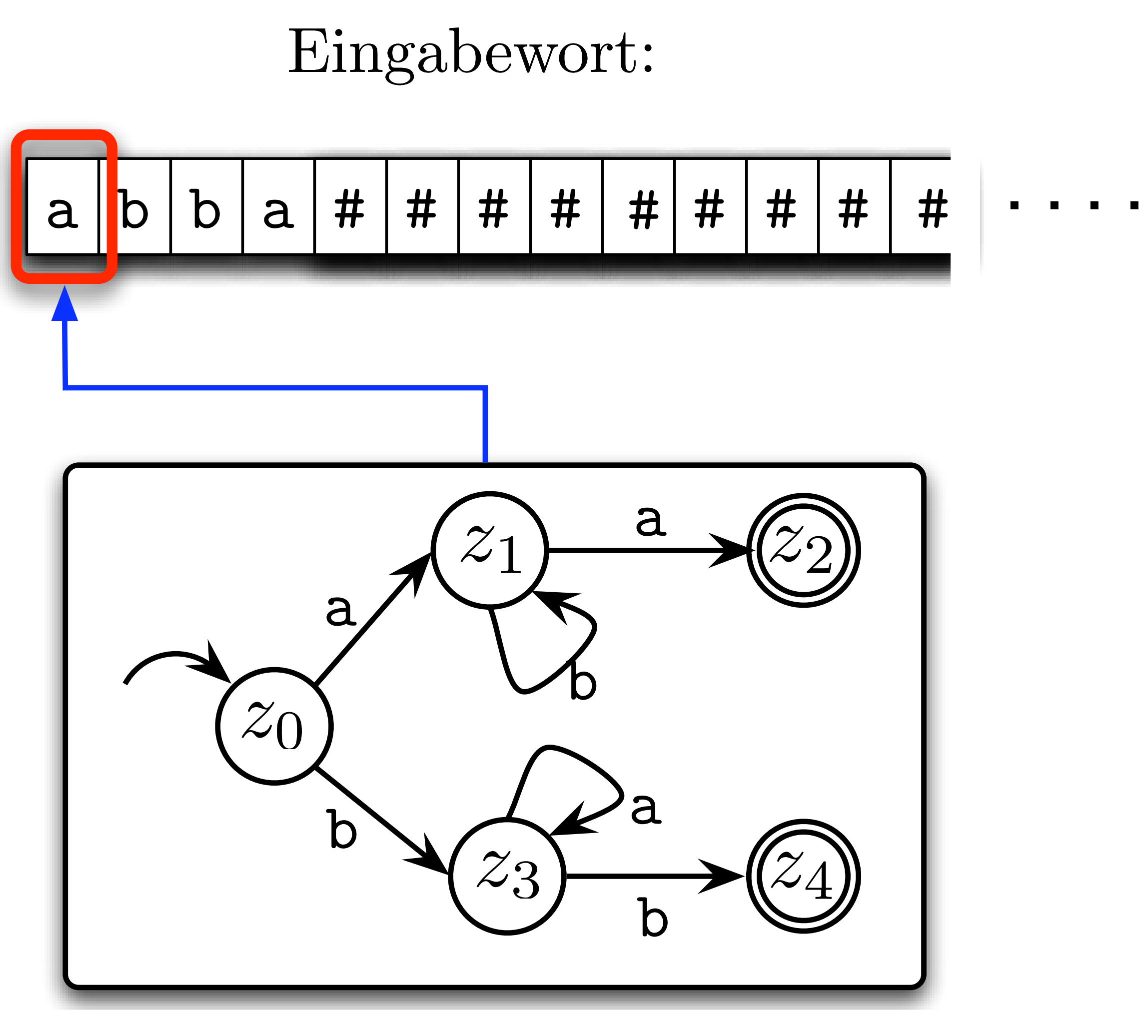
- Eine Sprache muss durch eine **endliche Repräsentation** dargestellt werden.
- Die Beschreibung einer Sprache muss **eindeutig** sein.
- Es kann **unterschiedliche Beschreibungen** für dieselbe Sprache geben.
- Wir werden dazu diesmal einfache Konzepte kennenlernen:
  - endliche Automaten
  - reguläre Ausdrücke
    (eigentlich rationale Ausdrücke)
  - Typ-3 Grammatiken (rechts- und linkslineare)

endlicher Automat

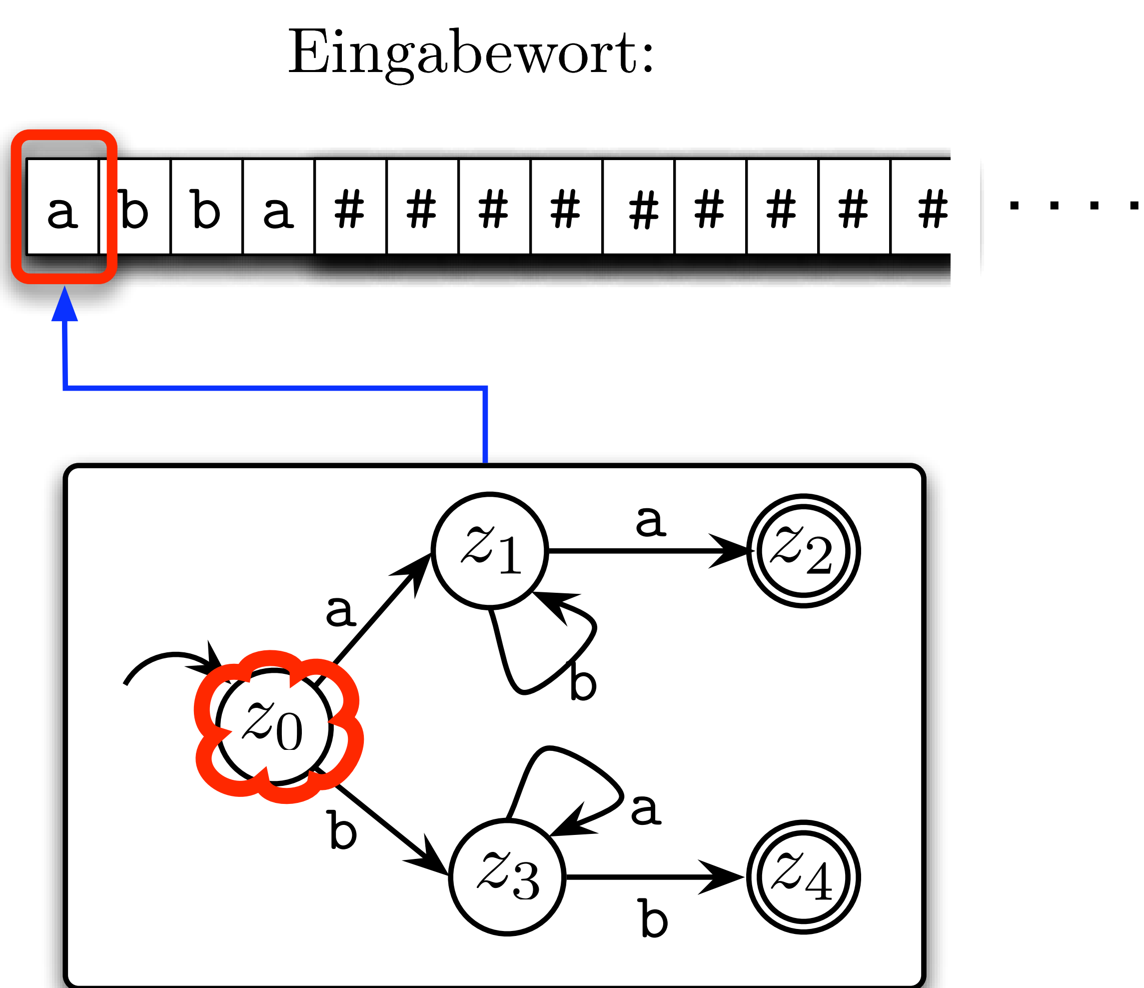
Deterministisches Modell



Beginn der Arbeitsweise (informal)

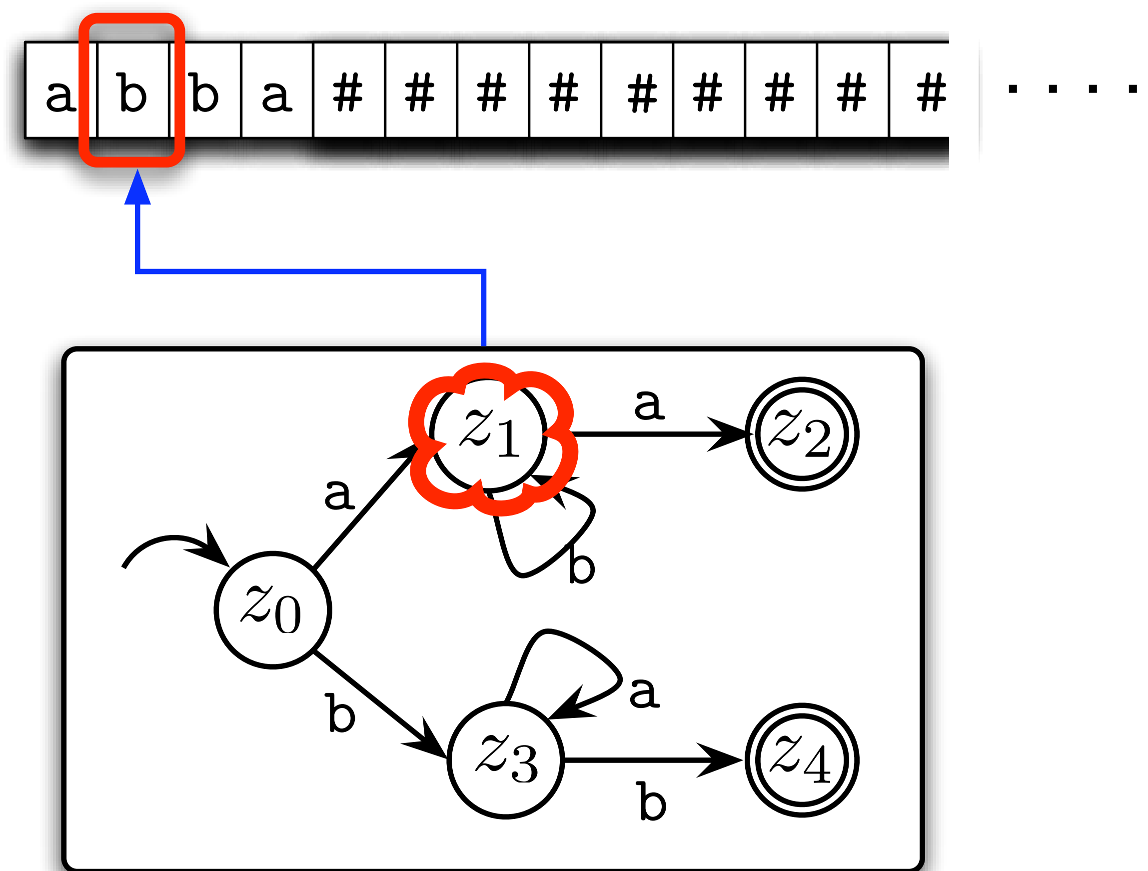


vor dem ersten Schritt

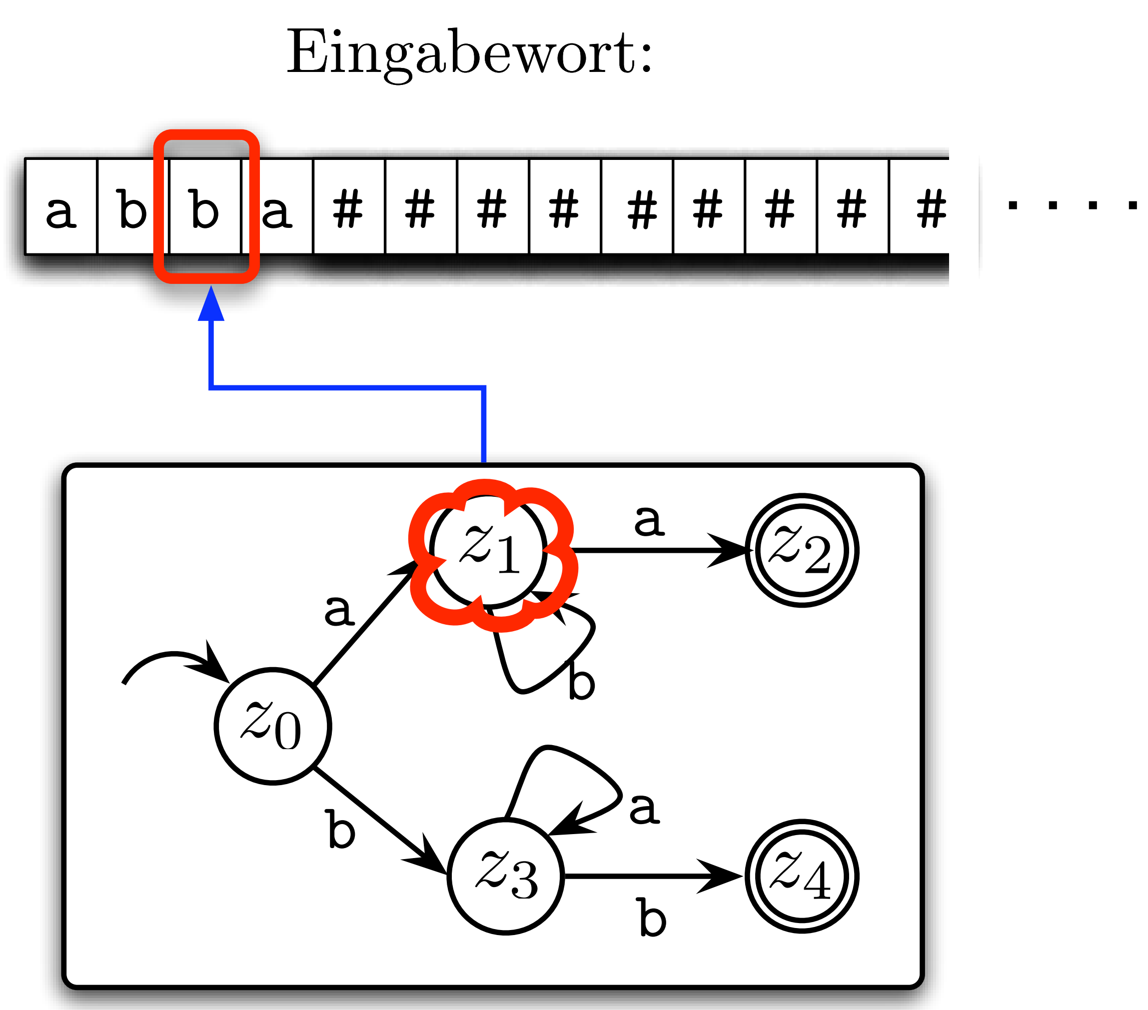


vor dem zweiten Schritt

Eingabewort:

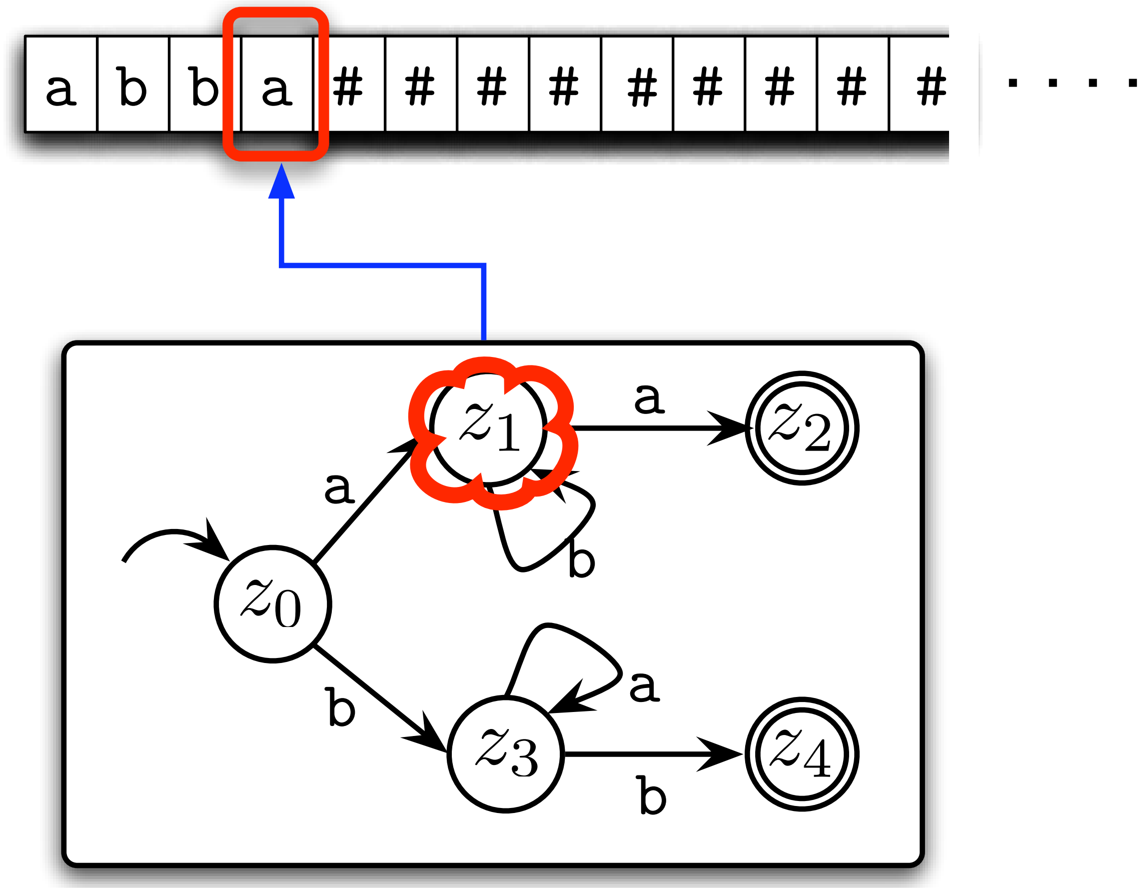


vor dem dritten Schritt

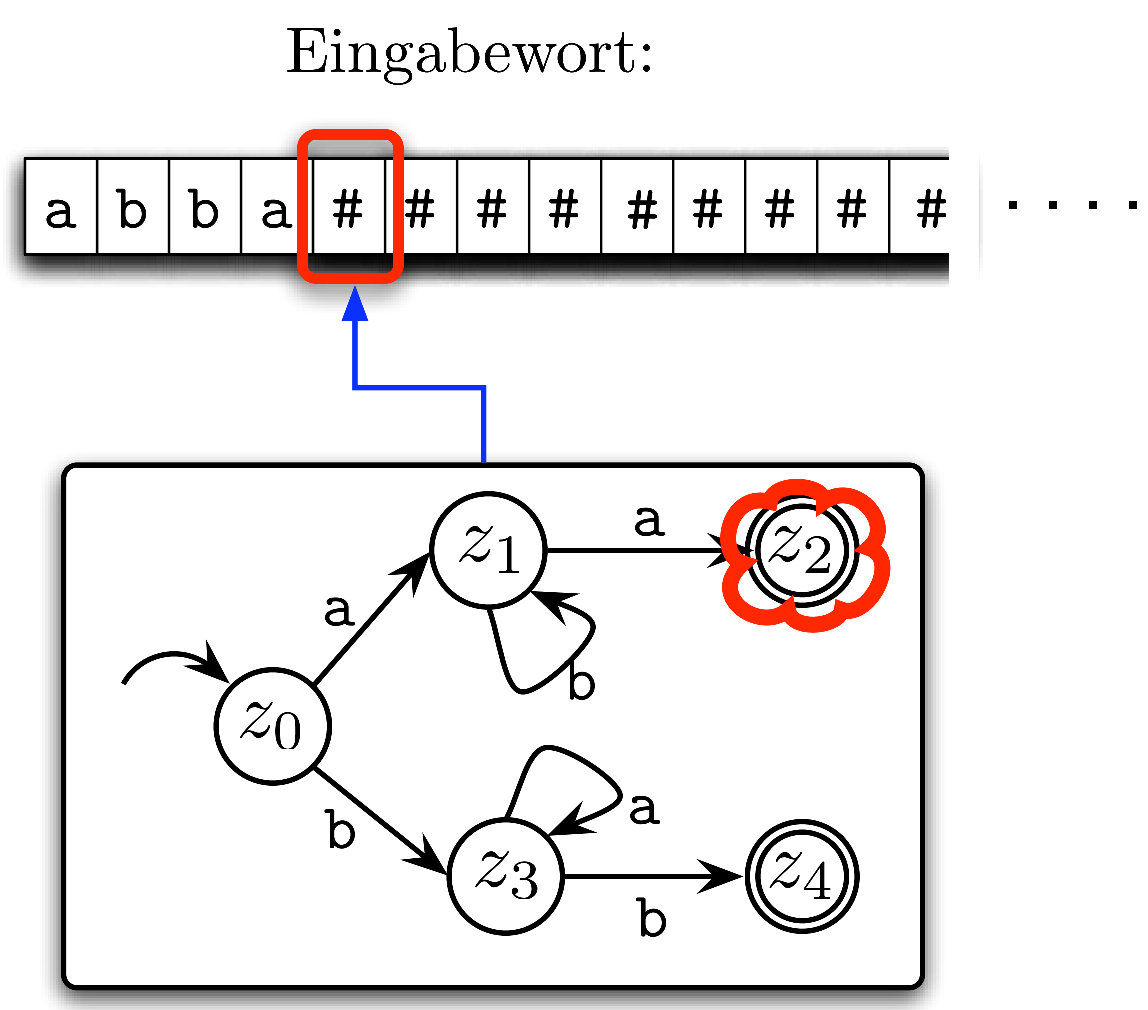


vor dem vierten Schritt

Eingabewort:

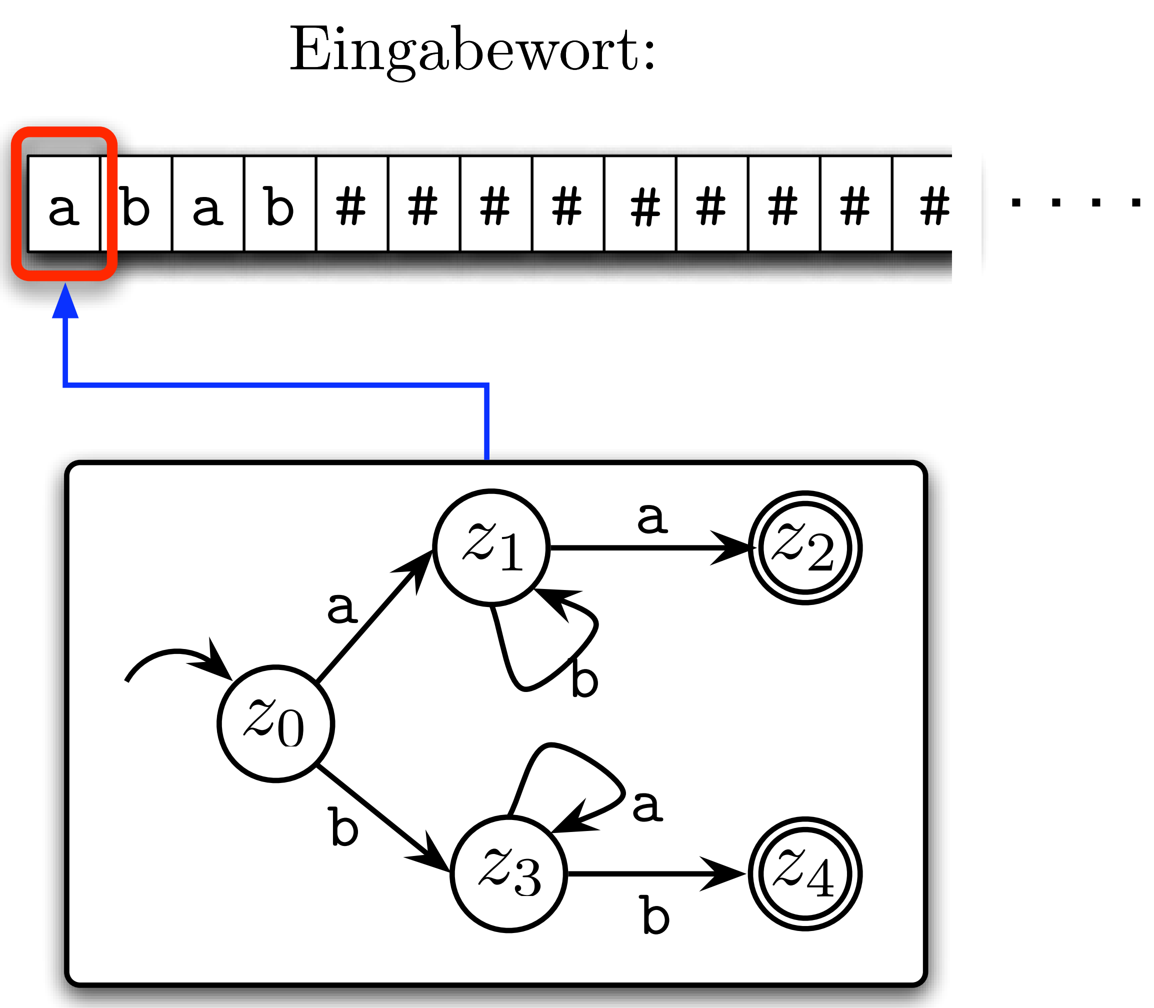


der letzte Schritt

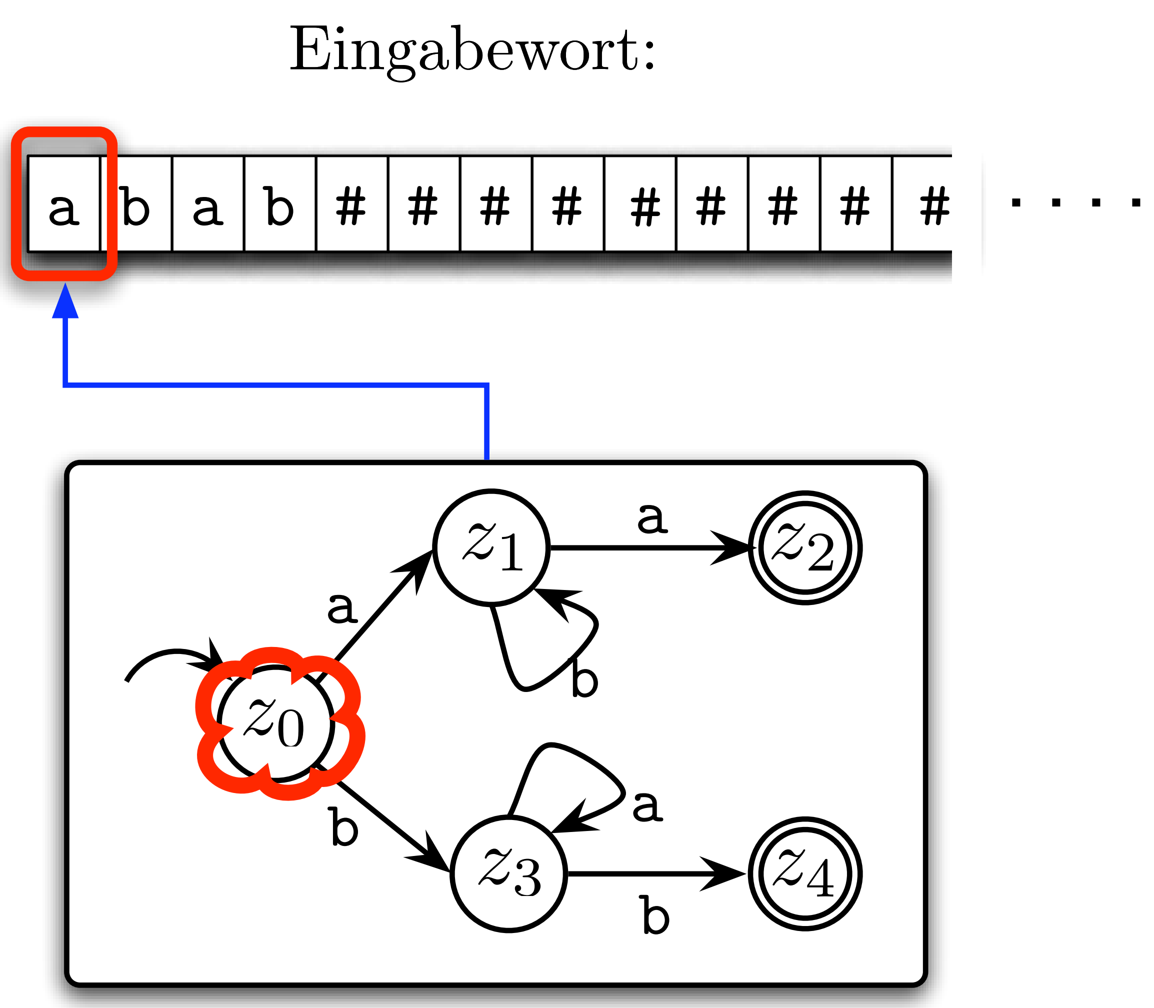


In diesem Zustand wurde das Eingabewort vollständig gelesen und ein Endzustand eingenommen: abba wird daher akzeptiert!

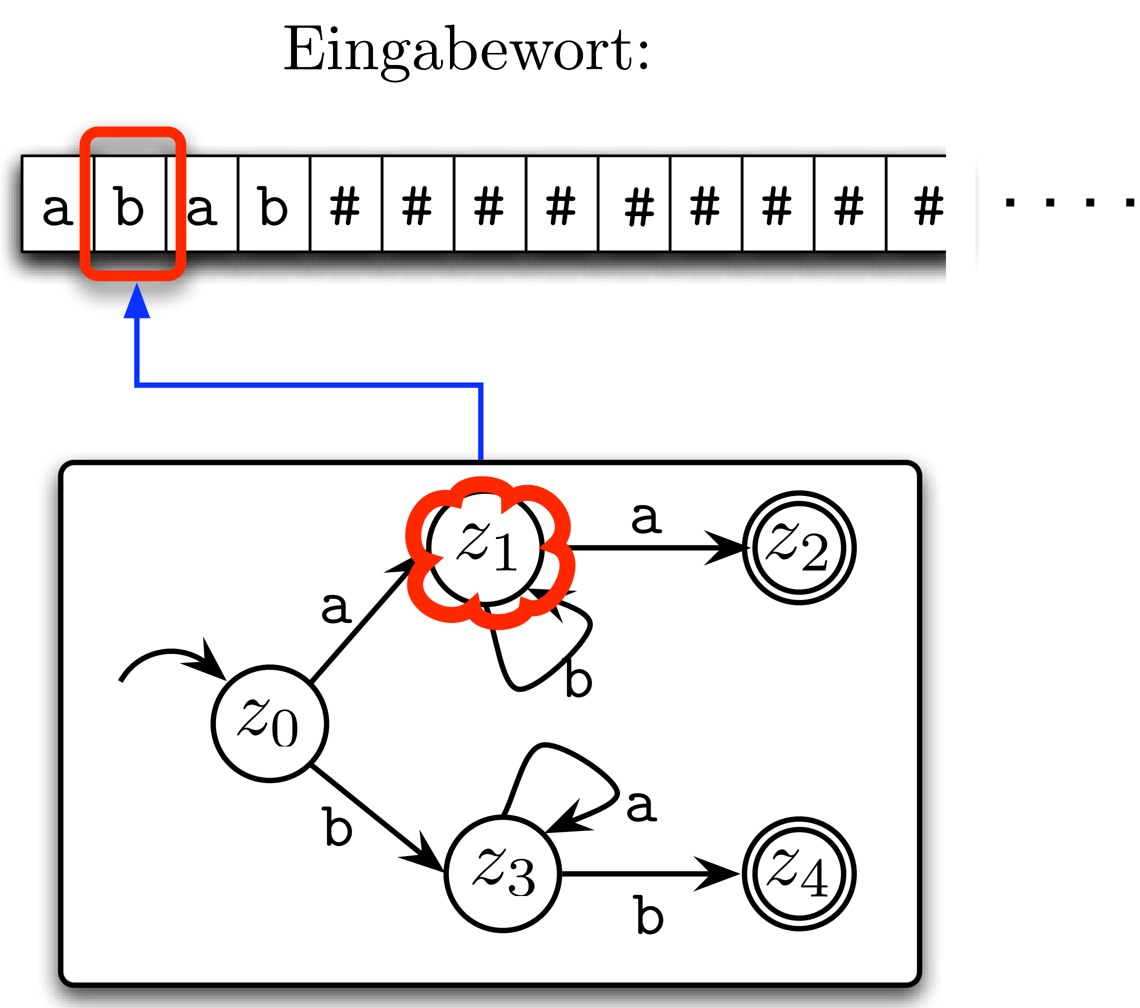
bei anderer Eingabe abab



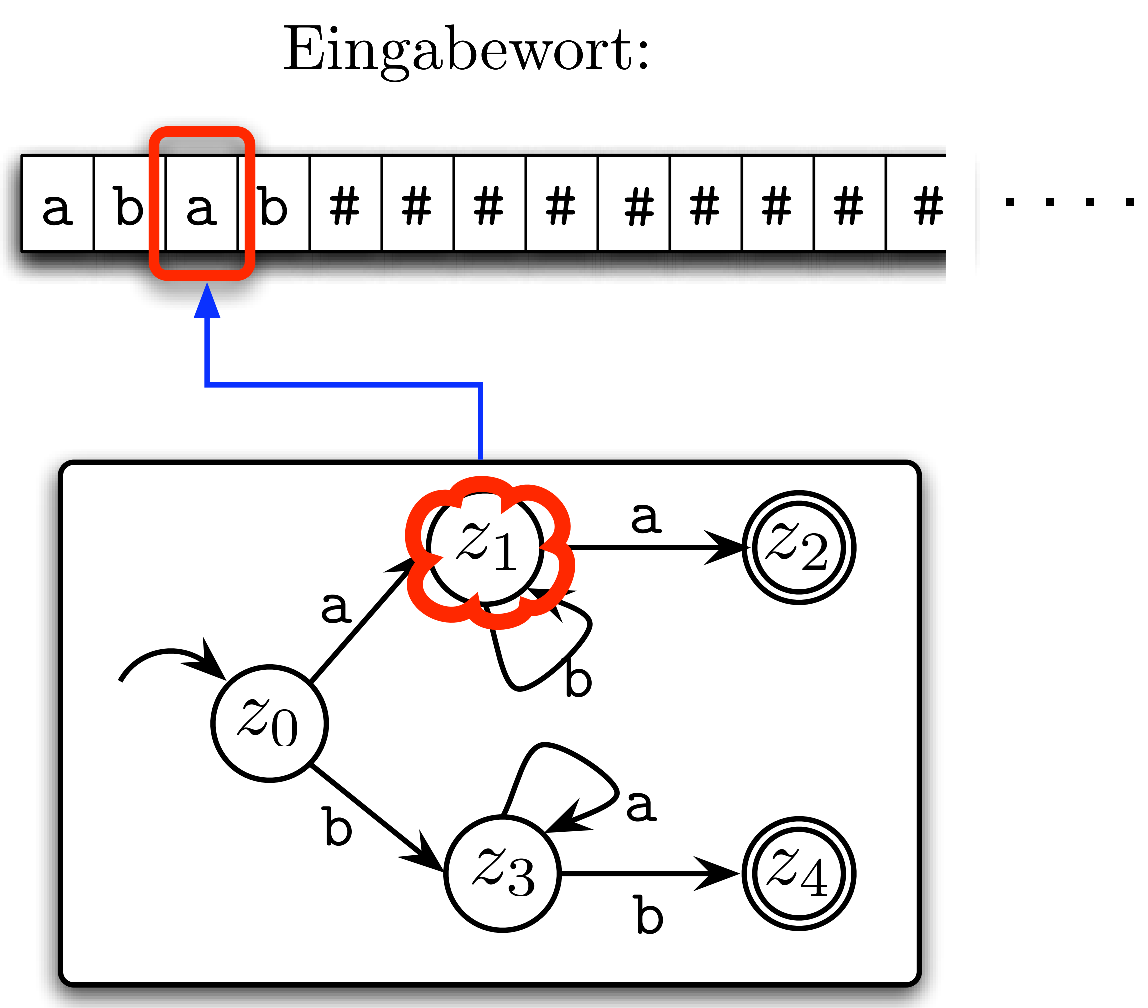
vor dem ersten Schritt



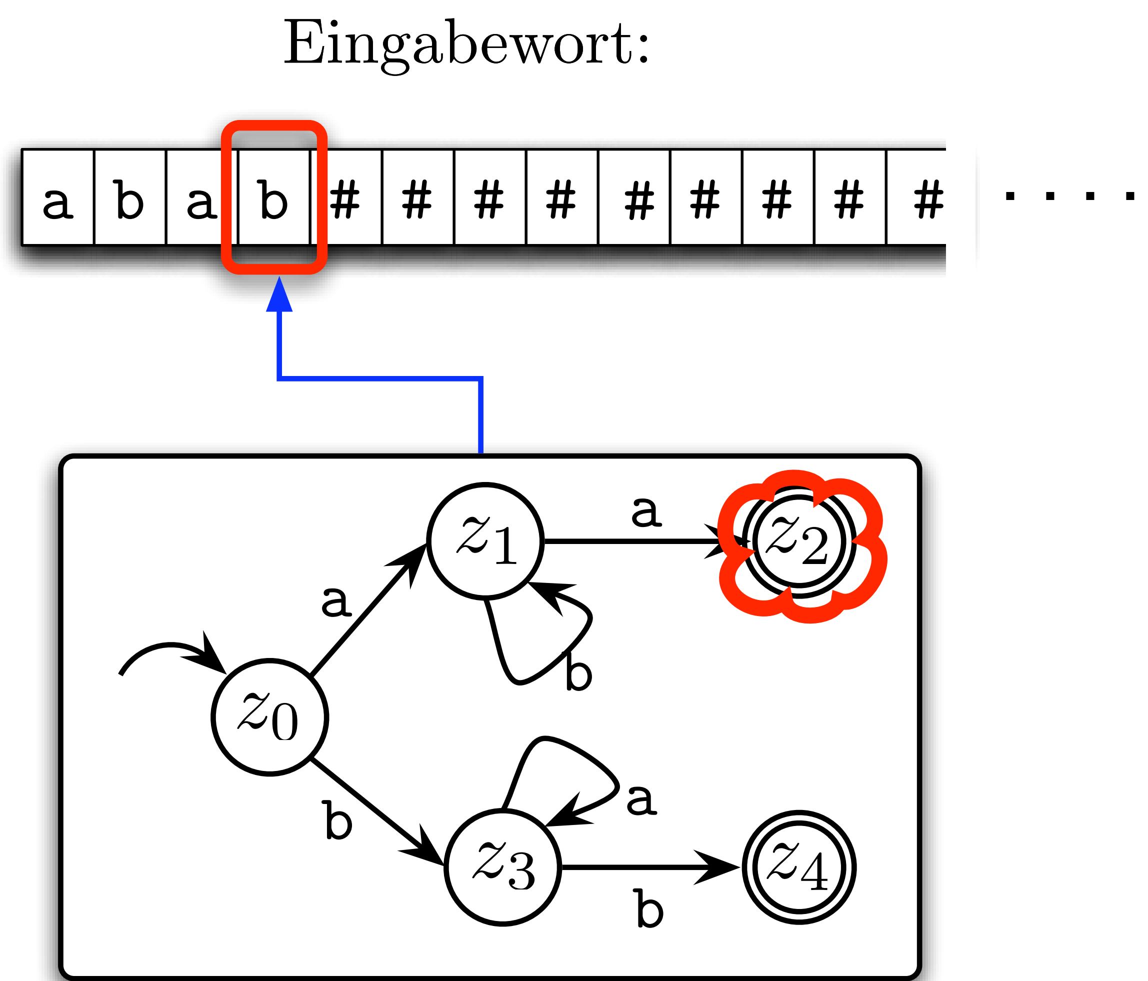
vor dem zweiten Schritt



vor dem dritten Schritt



vor dem vierten Schritt



In diesem Zustand ist für das Eingabesymbol b kein Übergang definiert: abab wird nicht akzeptiert!

deterministischer endlicher Automat (DFA)

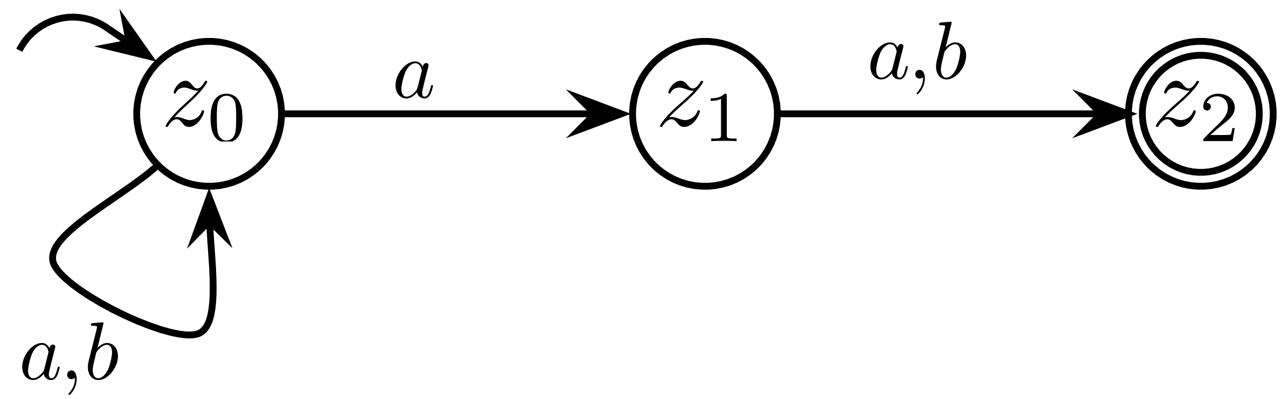
Definition 13.1:

Ein **deterministischer endlicher Automat**, ***DFA*** für ***d**eterministic **f**inite **a**utomaton* (**DEA** in einigen Deutschen Büchern) wird durch ein Quintupel $A := (Q, \Sigma, \delta, q_0, F)$ (auch $A := (\Sigma, S, \delta, s_0, F)$) beschrieben, wobei

- Q (oder S) eine endliche Menge von **Zuständen** ist,
- Σ ein endliches Alphabet von **Eingabesymbolen** ist,
- $\delta : Q \times \Sigma \longrightarrow Q$ die nicht notwendigerweise totale **Überführungsfunktion** ist,
- $q_0 \in Q$ (oder $s_0 \in S$) der **Startzustand** ist und
- $F \subseteq Q$ (oder $F \subseteq S$) die Menge der **Endzustände** ist.

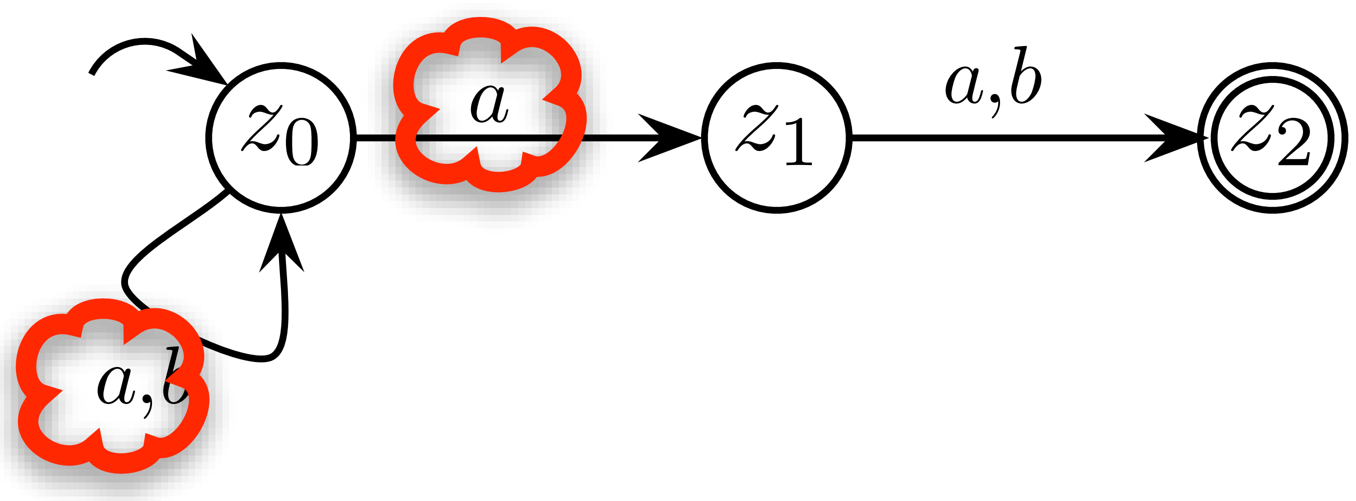
Quizfrage

Ist dies ein DFA?



Quizfrage

nein:



akzeptierte Sprache

Definition 13.2:

- Zu einem DFA $A := (Q, \Sigma, \delta, q_0, F)$ definieren wir die **erweiterte Überführungsfunktion**

$$\delta^* : Q \times \Sigma^* \longrightarrow Q$$

durch:

$$\delta^*(z, xw) := \delta^*(\delta(z, x), w)$$

für alle Zustände $z \in Q$, alle Symbole $x \in \Sigma$ und alle Wörter $w \in \Sigma^*$, sowie

$$\forall\, q \in Q : \delta^*(q, \epsilon) := q.$$

- Die von dem DFA A **akzeptierte Sprache** ist die Menge

$$L(A) := \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}.$$

Die Familie REG der regulären Sprachen

Definition 13.3:

- Die Mengen von Wörtern, die von einem *DFA* akzeptiert werden können, heißen **reguläre Mengen**.
- Die Familie der regulären Mengen wird mit *REG* bezeichnet.
- Mit DFA_Σ wird die Familie aller Sprachen $L \subseteq \Sigma^*$ bezeichnet, die von DFAs mit Eingabe-Alphabet Σ akzeptiert werden können.
- Es gilt:

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

  und

  $$REG_\Sigma := DFA_\Sigma \text{ sowie } REG = \bigcup_{\substack{\Sigma \text{ ist endl.} \\ \text{Alphabet}}} REG_\Sigma.$$

vollständiger DFA und andere DFA

Definition 13.4:

Ein *DFA* $A := (Q, \Sigma, \delta, q_0, F)$ heißt:

- **vollständig** (Abk.: *vDFA*)
  genau dann, wenn für jedes $(p, x) \in Q \times \Sigma$ ein $q \in Q$ existiert, so dass $q = \delta(p, x)$ ist.

  Jede Eingabe kann verarbeitet werden!

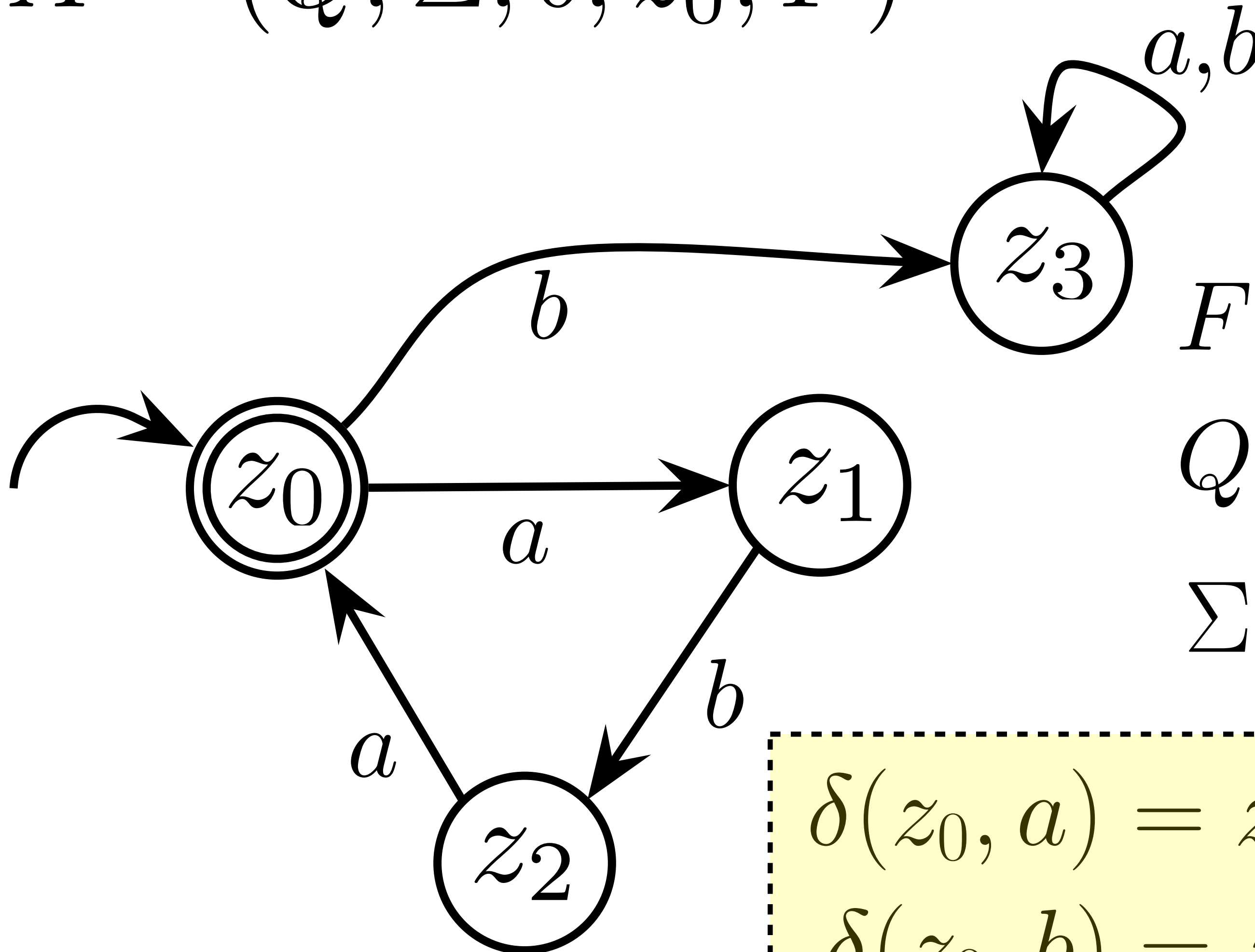
- **initial zusammenhängend** (Abk.: *izDFA*)
  genau dann, wenn zu jedem Zustand $p \in Q$ ein Wort $w \in \Sigma^*$ existiert, mit $p = \delta^*(z_0, w)$.

  Jeder Zustand kann von z_0 erreicht werden!

- Zwei *DFA*, A und B, heißen **äquivalent**
  genau dann, wenn $L(A) = L(B)$ ist, sie also die gleiche Sprache akzeptieren.

ein anderes Beispiel

$$A = (Q, \Sigma, \delta, z_0, F)$$



$$F = \{z_0\}$$

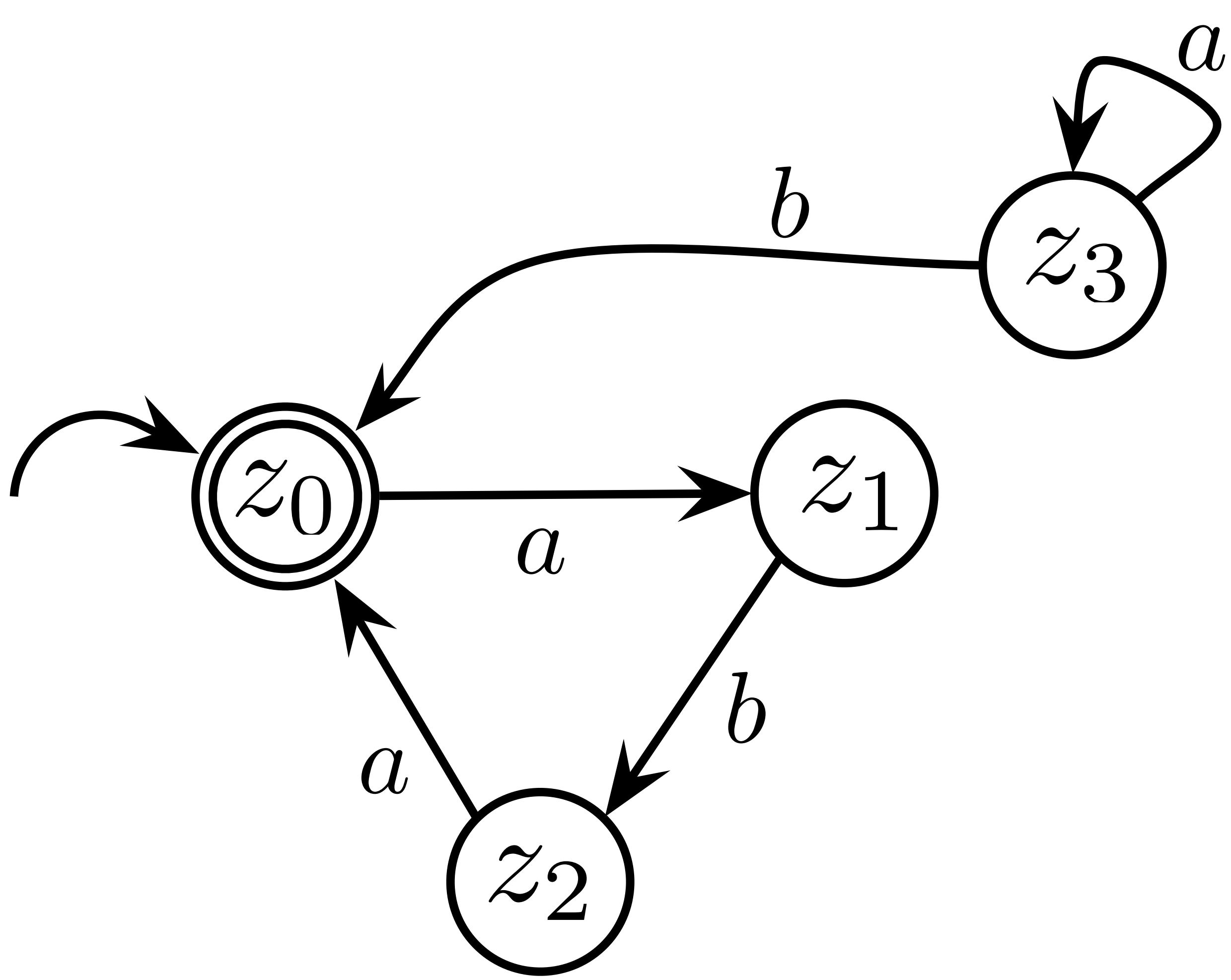
$$Q = \{z_0, z_1, z_2, z_3\}$$

$$\Sigma = \{a, b\}$$

$\delta(z_0, a) = z_1$	$\delta(z_2, a) = z_0$
$\delta(z_0, b) = z_3$	$\delta(z_3, a) = z_3$
$\delta(z_1, b) = z_2$	$\delta(z_3, b) = z_3$

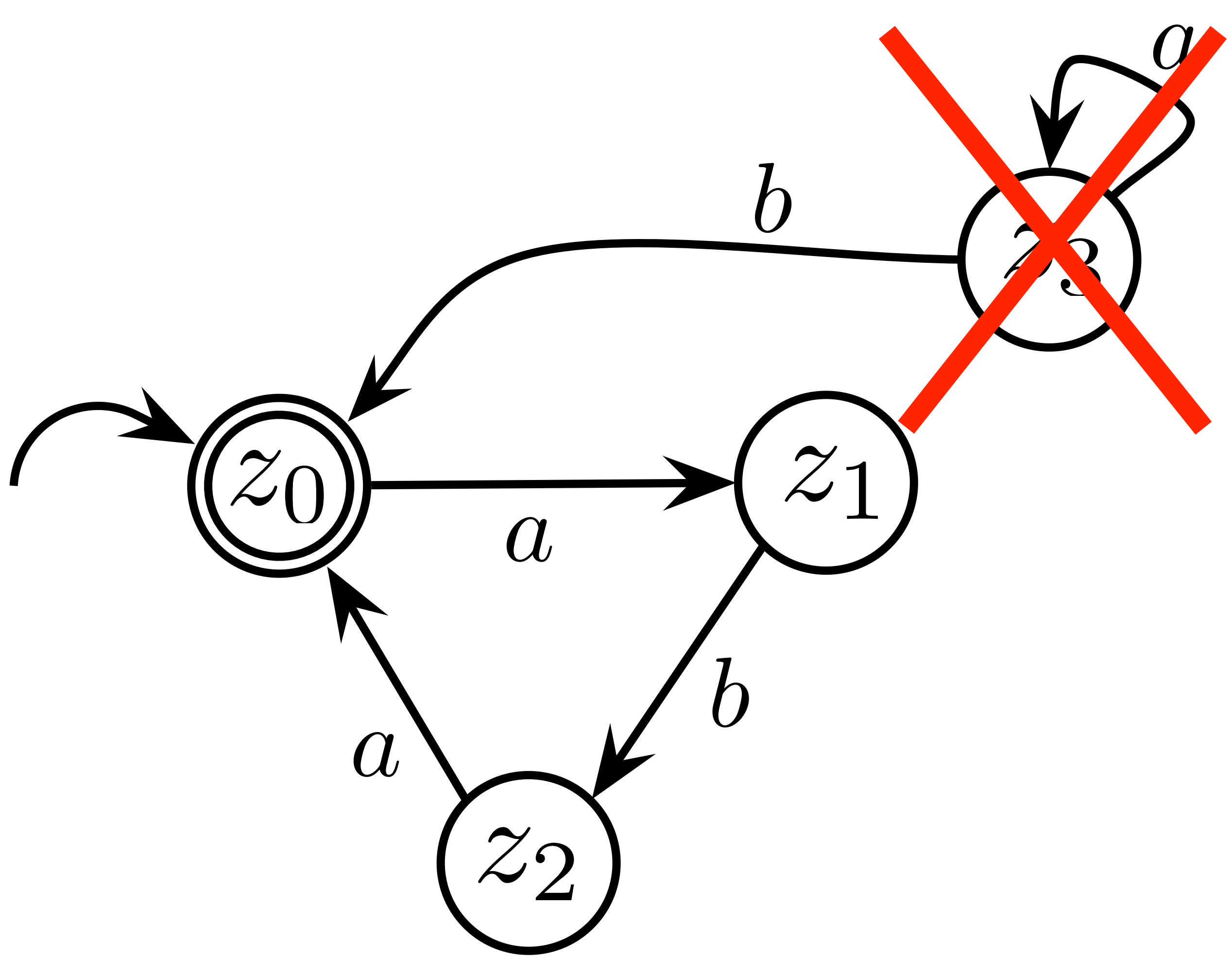
Dieser DFA ist initial zusammenhängend!

ein anderes Beispiel



Dieser DFA ist weder initial zusammenhängend noch vollständig!

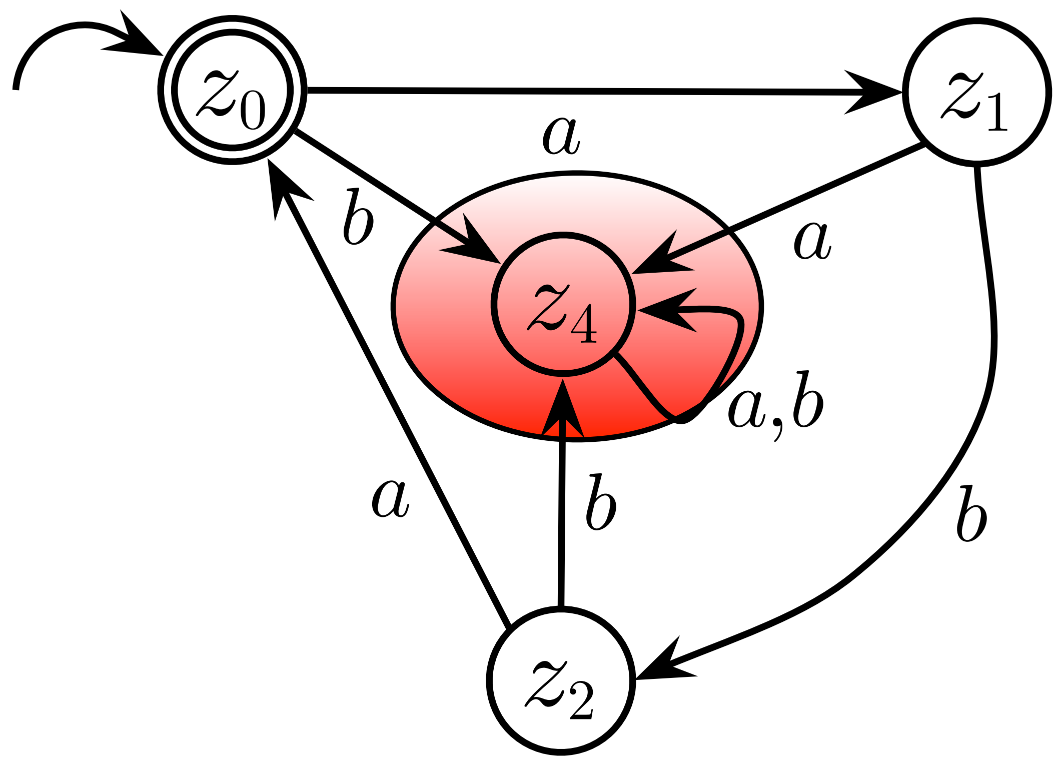
ein anderes Beispiel



Dieser DFA ist weder initial zusammenhängend noch vollständig!

Vervollständigen eines DFA ist leicht:

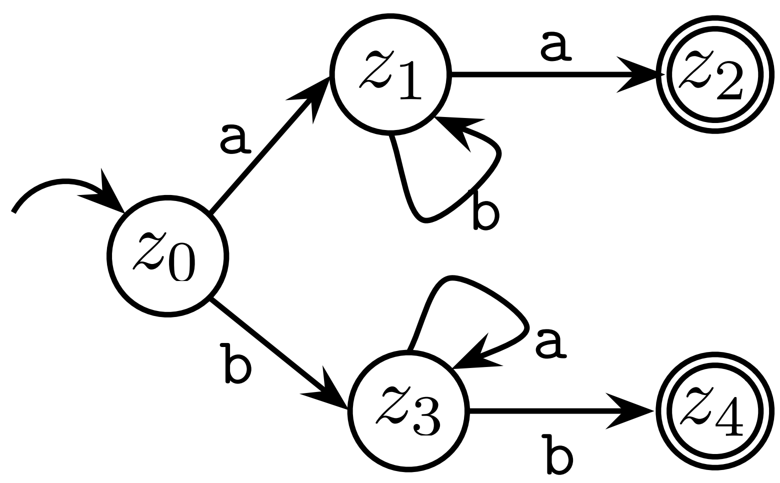
Einen neuen Zustand als „Falle“ ergänzen:
Alle Kanten für nicht erklärte Übergänge führen hinein;
der Zustand kann nicht wieder verlassen werden!



Dieser DFA ist initial zusammenhängend und vollständig, also zugleich izDFA und vDFA!

Der DFA als Akzeptor

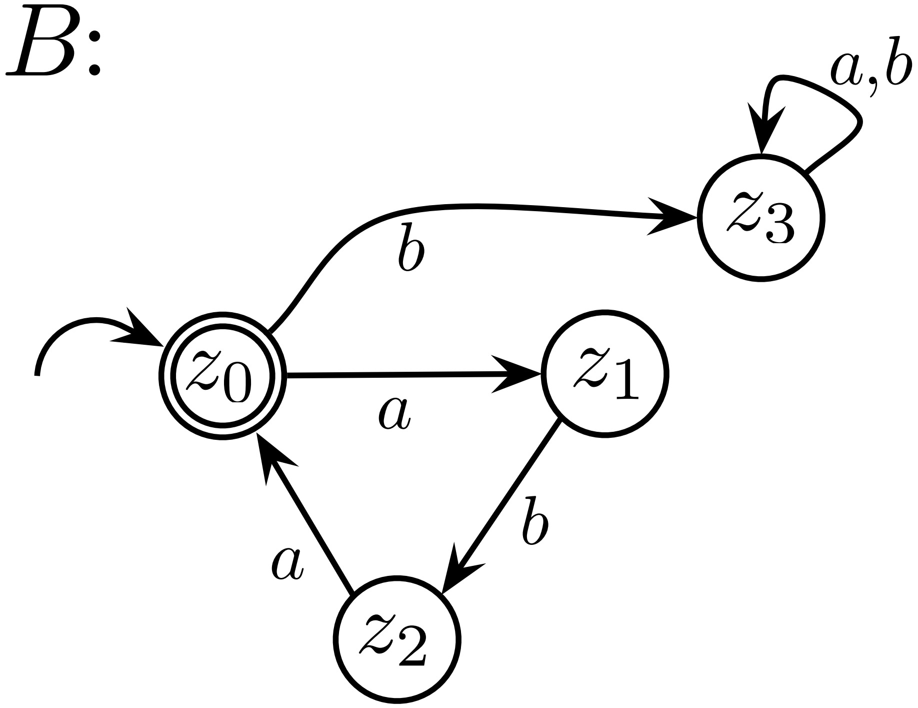
A:



$$L(A) = \{w \in \{a,b\}^* \mid \exists n \in \mathbb{N}(w = ab^n a \vee w = ba^n b)\}$$

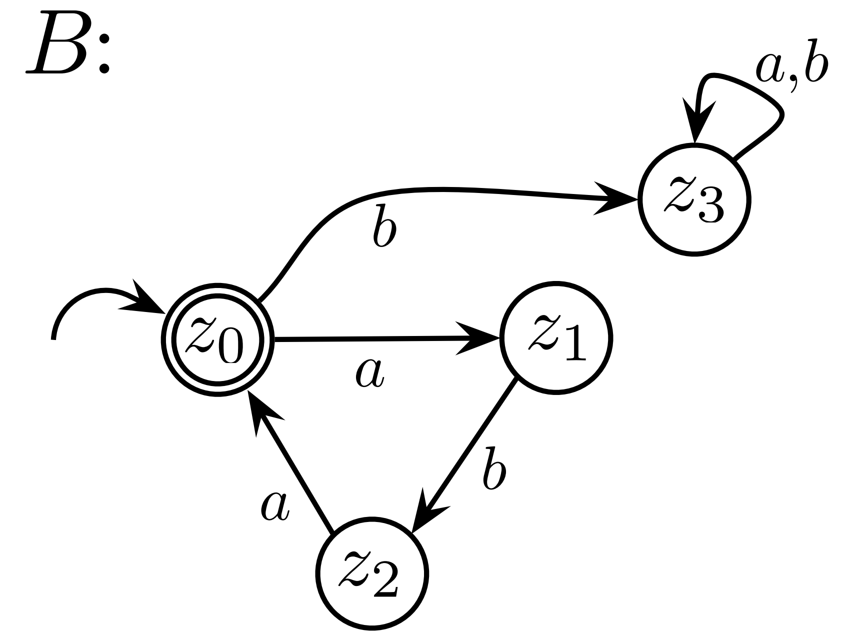
… alle Wörter, die mit a beginnen und enden und dazwischen nur b's haben und umgekehrt.

B:



$L(B) = \{aba\}^* =$
$\{\epsilon, aba, abaaba, abaabaaba, \ldots\}$

Wie beweist man L(B) = L?



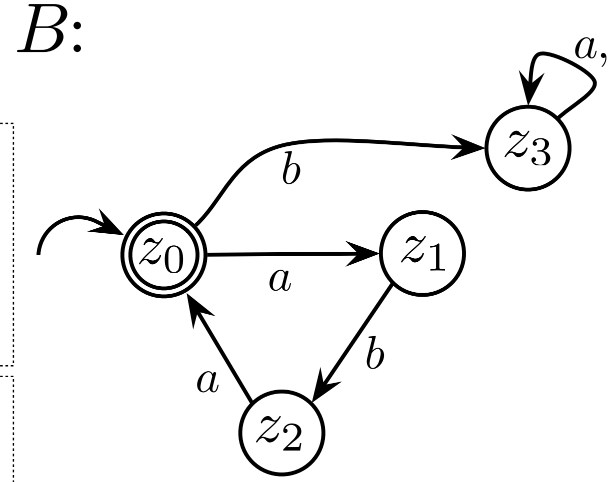
Aussage 1 *Es gilt* $L(B) = L := \{aba\}^*$.

Beweis: Die Mengengleichheit zeigen wir, indem wir zwei Inklusionen zeigen:

1. $L(B) \subseteq L$ (= Jedes von B akzeptierte Wort ist in L.)
2. $L \subseteq L(B)$ (= Jedes Wort in L wird von B akzeptiert.)

Variante 1: L(B) = L

B:



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

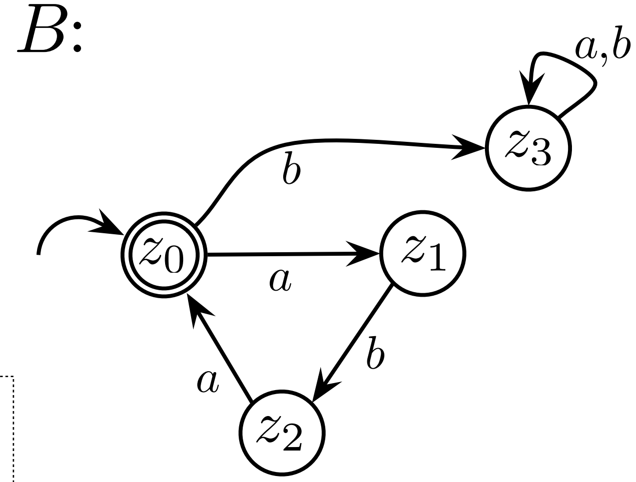
   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \in F$$

   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

Variante 1: L(B) = L

B:



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

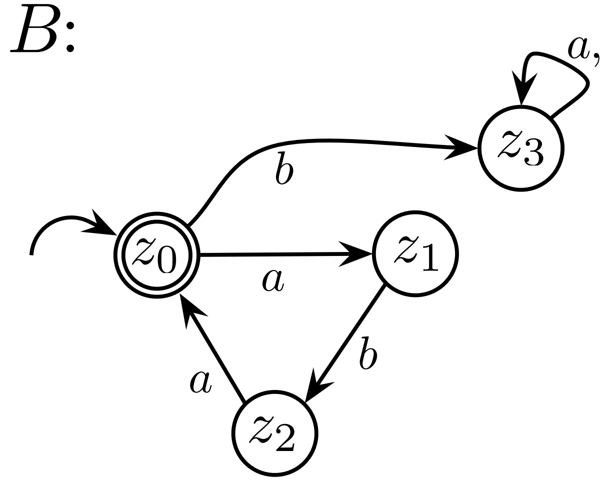
   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \in F$$

   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

Variante 1: L(B) = L

B:



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \in F$$

   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

Variante 1: L(B) = L

1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

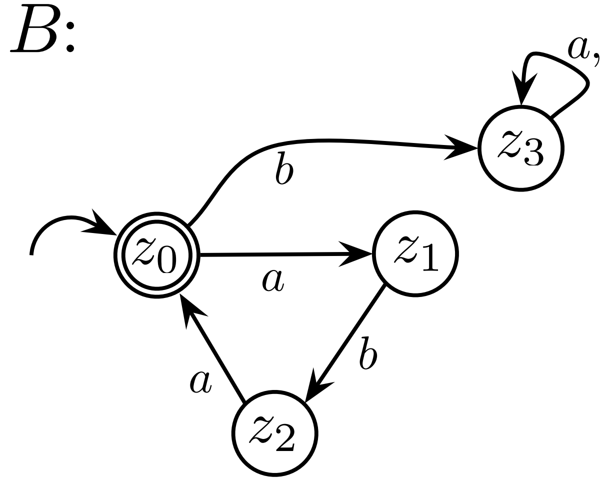
2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \in F$$

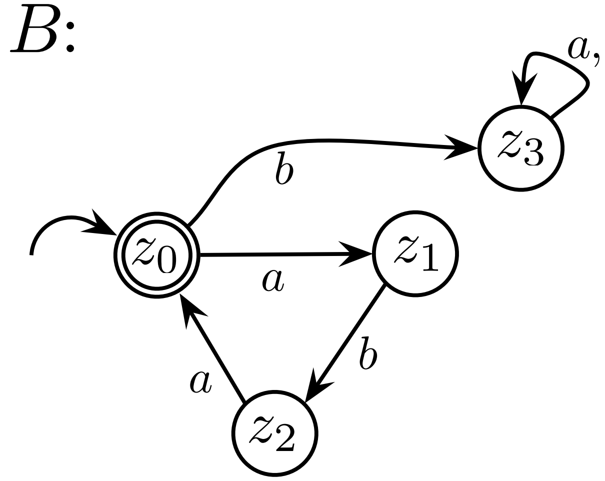
   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

B:



Variante 1: L(B) = L

B:



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \in F$$

   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

Variante 1: L(B) = L

1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

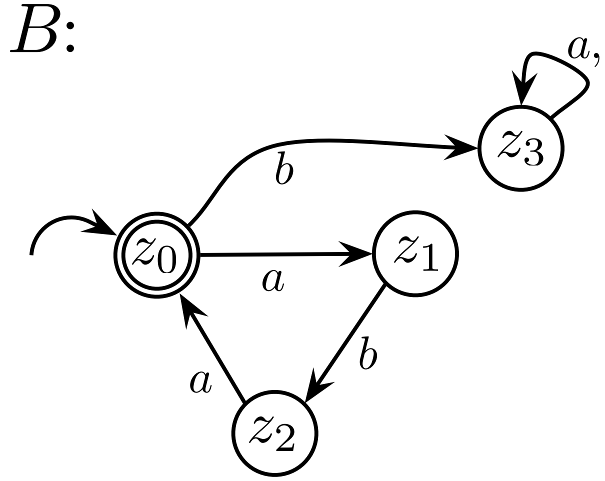
2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \xrightarrow{a} \cdots z_0 \underbrace{\xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \in F$$

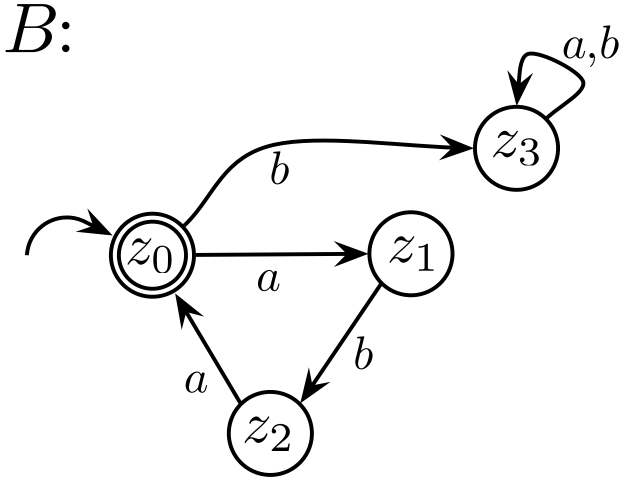
Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

B:



Variante 1: L(B) = L

B:



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort *aba*. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

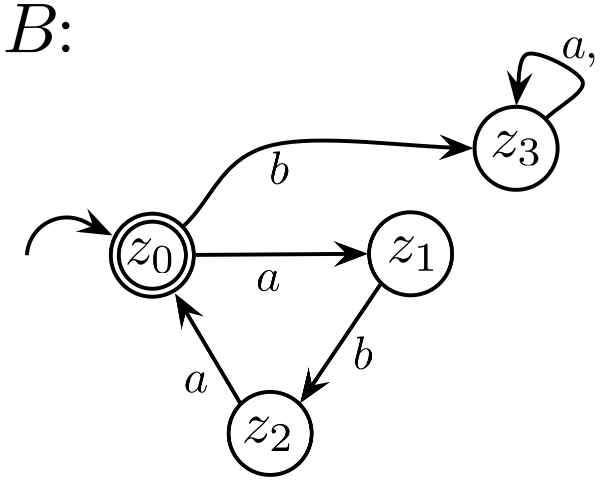
   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \in F$$

   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

Variante 1: L(B) = L

B:



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

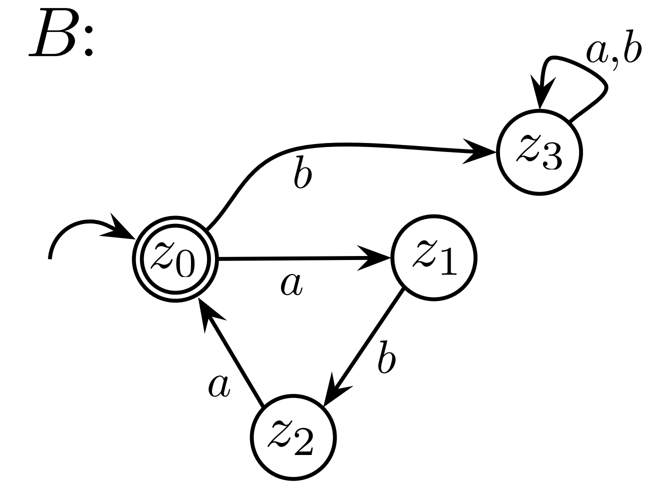
2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$z_0 \underbrace{\stackrel{a}{\to} z_1 \stackrel{b}{\to} z_2 \stackrel{a}{\to} z_0} \stackrel{a}{\to} \cdots z_0 \underbrace{\stackrel{a}{\to} z_1 \stackrel{b}{\to} z_2 \stackrel{a}{\to} z_0} \in F$$

   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

Variante 1: L(B) = L



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

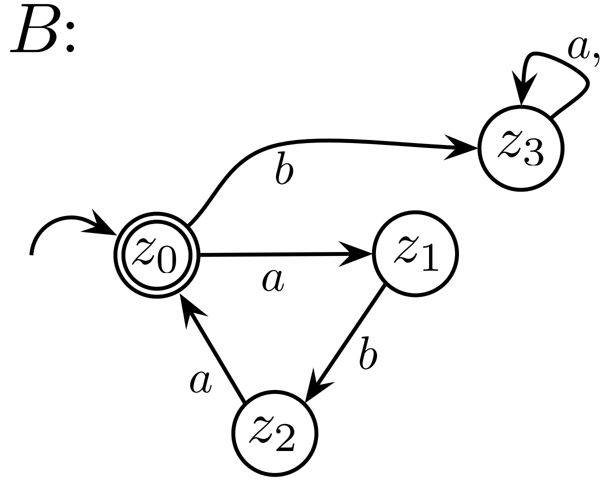
$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0} \in F$$

Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

qed.

Variante 1: L(B) = L

B:



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Nicht formal genug??

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \in F$$

   Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

qed.

Variante 1: L(B) = L

1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

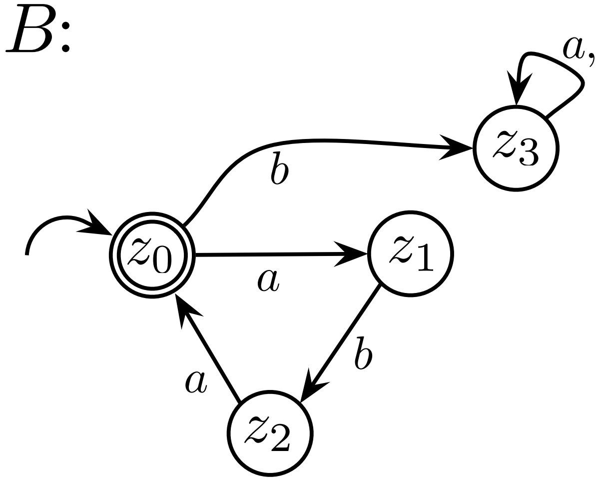
   Nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Die einzige Möglichkeit, von z_0 nach z_0 zu gelangen, besteht darin, beliebig oft (also n-mal für ein $n \in \mathbb{N}$) den Kreis $z_0 z_1 z_2$ zu durchlaufen.

   Nicht formal genug???

   Entlang des Kreises $z_0 z_1 z_2$ lesen wir das Wort aba. Durchlaufen wir den Kreis n-mal, dann haben wir also das Wort $w = (aba)^n$ gelesen. Da $(aba)^n \in \{aba\}^* = L$ für jedes n gilt, haben wir $w \in L$ gezeigt.

B:



2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Kein Problem!

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist. Dieses Wort durchläuft folgende Zustände:

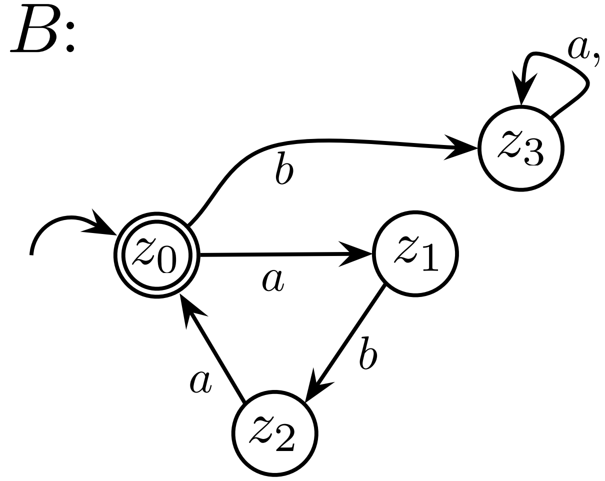
$$\underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \xrightarrow{a} \cdots \underbrace{z_0 \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{a} z_0}_{} \in F$$

Also gilt für jedes n auch $w = (aba)^n \in L(B)$.

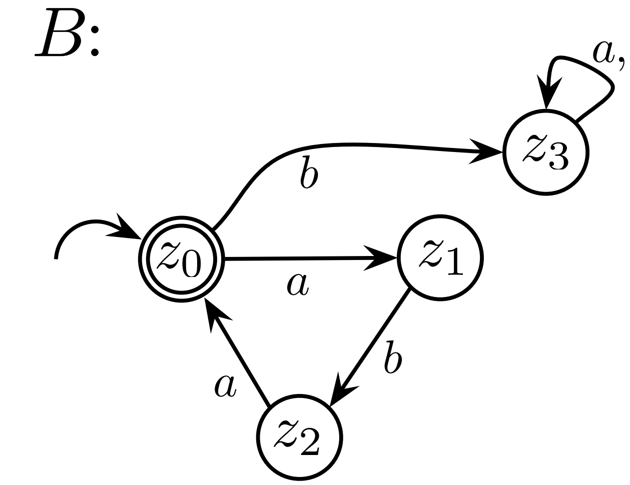
qed.

Variante 2: L(B) = L

B:



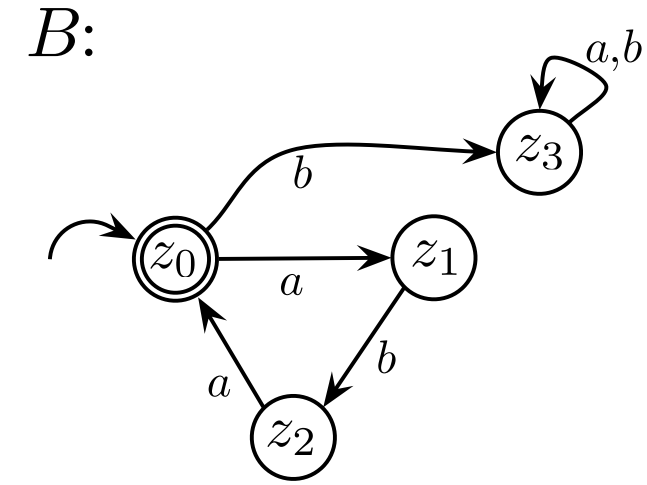
Variante 2: L(B) = L



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Damit das Wort w akzeptiert wird, muss das Wort vom Start- in einem Endzustand führen (hier: jeweils z_0), denn nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

Variante 2: L(B) = L



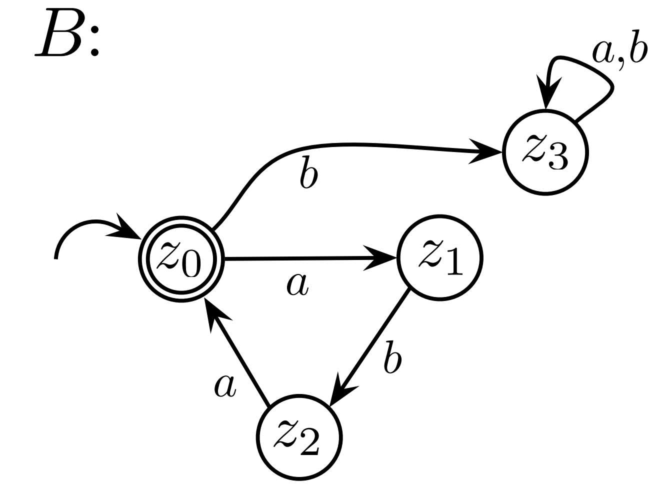
1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Damit das Wort w akzeptiert wird, muss das Wort vom Start- in einem Endzustand führen (hier: jeweils z_0), denn nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

Wir zeigen per Induktion über die Wortlänge $n = |w|$ folgende Behauptung:

Für jedes Wort $w \in \Sigma^*$ gilt: Wenn $w \in L(B)$, dann ist $|w|$ ein Vielfaches von 3, d.h. $n = 3k$ für ein k und $w = (aba)^k$.

Variante 2: L(B) = L



1. $L(B) \subseteq L$: Sei $w \in L(B)$. Wir müssen $w \in L$ zeigen.

   Damit das Wort w akzeptiert wird, muss das Wort vom Start- in einem Endzustand führen (hier: jeweils z_0), denn nach Definition gilt $w \in L(B)$ gdw. $\delta^*(z_0, w) \in F$, d.h. hier: $\delta^*(z_0, w) = z_0$, da $F = \{z_0\}$.

   Wir zeigen per Induktion über die Wortlänge $n = |w|$ folgende Behauptung:

   Für jedes Wort $w \in \Sigma^*$ gilt: Wenn $w \in L(B)$, dann ist $|w|$ ein Vielfaches von 3, d.h. $n = 3k$ für ein k und $w = (aba)^k$.

   (a) Induktionsanfang für $n = 0$.

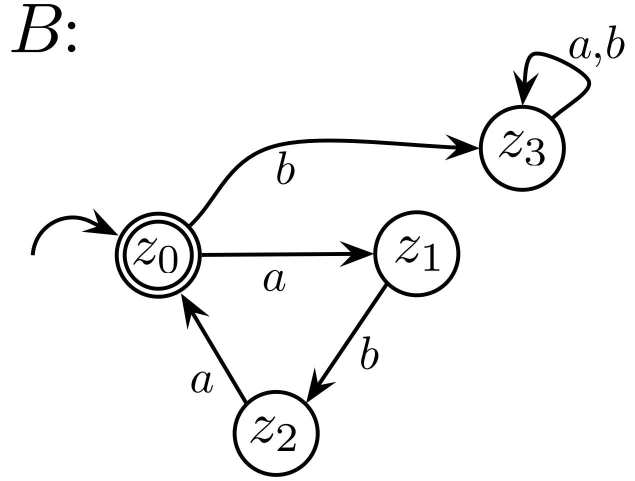
   Dann ist $w = \epsilon$. Da nach Defintion $\delta^*(z_0, \epsilon) = z_0 \in F$ gilt, haben wir $w \in L(B)$.

   Die Wenn-Dann-Schlussfolgerung gilt, denn $|w| = 0$ ein Vielfaches von 3, da $n = 3k$ für $k = 0$ und $w = \epsilon = (aba)^0 = (aba)^k$.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass die Behauptung für alle m, die echt kleiner als ein *festes* n sind, gilt.

B:



(c) Induktionsschritt: Sei $|w| > 0$, d.h. $w = w_1 \cdots w_n$ mit $w_i \in \Sigma$ und $n > 0$.

Wenn das Wort die Länge 1 oder 2 hat, dann wird es nicht akzeptiert, und es ist im Wenn-Dann-Teil nichts zu zeigen. Sei also $|w| \geq 3$.

Aus den Diagramm kann man erkennen, dass die *einzige* Möglichkeit, mit einem Wort w den Endzustand z_0 zu erreichen, darin besteht, dass die letzten drei Zeichen aba sind und dass man zuvor in dem Zustand z_0 gewesen ist:

$$\delta^*(z_0, w_1 \cdots w_{n-3}) = z_0$$

Also gilt für das Teilwort $w_1 \cdots w_{n-3} \in L(B)$.

Da es die Länge $n-3 < n$ besitzt, ist die IA anwendbar, d.h. es gibt ein l, so dass:

$$(w_1 \cdots w_{n-3}) = (aba)^l$$

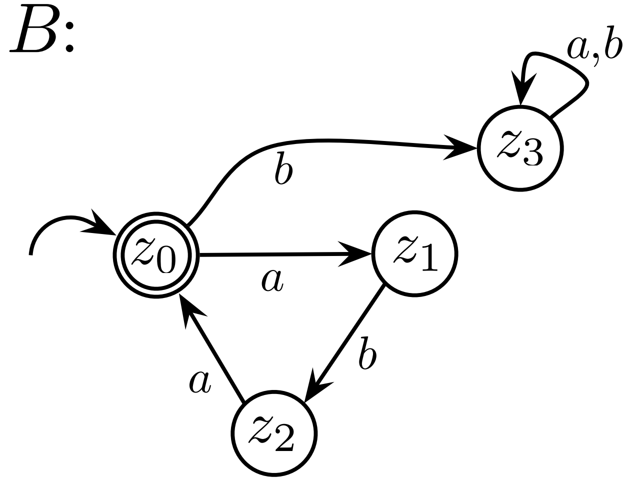
Also ist auch w ein Vielfaches von 3, denn $|w| = |(aba)^l(aba)| = 3(l+1)$ und $w = (aba)^l(aba) = (aba)^{l+1}$.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass die Behauptung für alle m, die echt kleiner als ein *festes* n sind, gilt.

(c) Induktionsschritt: Sei $|w| > 0$, d.h. $w = w_1 \cdots w_n$ mit $w_i \in \Sigma$ und $n > 0$.

B:



Wenn das Wort die Länge 1 oder 2 hat, dann wird es nicht akzeptiert, und es ist im Wenn-Dann-Teil nichts zu zeigen. Sei also $|w| \geq 3$.

Aus den Diagramm kann man erkennen, dass die *einzige* Möglichkeit, mit einem Wort w den Endzustand z_0 zu erreichen, darin besteht, dass die letzten drei Zeichen aba sind und dass man zuvor in dem Zustand z_0 gewesen ist:

$$\delta^*(z_0, w_1 \cdots w_{n-3}) = z_0$$

Also gilt für das Teilwort $w_1 \cdots w_{n-3} \in L(B)$.

Da es die Länge $n-3 < n$ besitzt, ist die IA anwendbar, d.h. es gibt ein l, so dass:

$$(w_1 \cdots w_{n-3}) = (aba)^l$$

Also ist auch w ein Vielfaches von 3, denn $|w| = |(aba)^l(aba)| = 3(l+1)$ und $w = (aba)^l(aba) = (aba)^{l+1}$.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass die Behauptung für alle m, die echt kleiner als ein *festes* n sind, gilt.

(c) Induktionsschritt: Sei $|w| > 0$, d.h. $w = w_1 \cdots w_n$ mit $w_i \in \Sigma$ und $n > 0$.

Wenn das Wort die Länge 1 oder 2 hat, dann wird es nicht akzeptiert, und es ist im Wenn-Dann-Teil nichts zu zeigen. Sei also $|w| \geq 3$.

Aus den Diagramm kann man erkennen, dass die *einzige* Möglichkeit, mit einem Wort w den Endzustand z_0 zu erreichen, darin besteht, dass die letzten drei Zeichen aba sind und dass man zuvor in dem Zustand z_0 gewesen ist:

$$\delta^*(z_0, w_1 \cdots w_{n-3}) = z_0$$

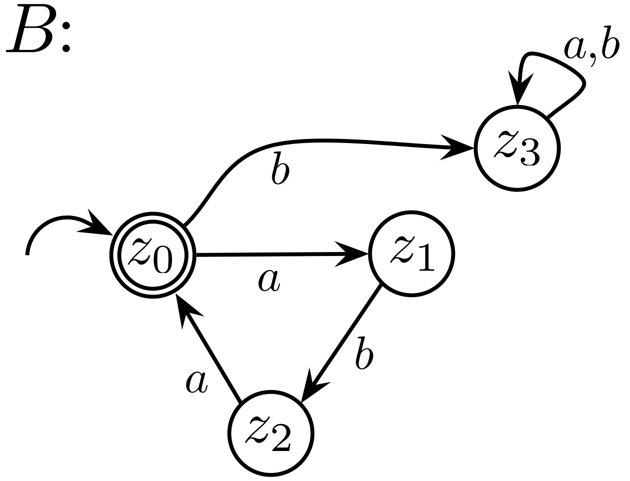
Also gilt für das Teilwort $w_1 \cdots w_{n-3} \in L(B)$.

Da es die Länge $n-3 < n$ besitzt, ist die IA anwendbar, d.h. es gibt ein l, so dass:

$$(w_1 \cdots w_{n-3}) = (aba)^l$$

Also ist auch w ein Vielfaches von 3, denn $|w| = |(aba)^l(aba)| = 3(l+1)$ und $w = (aba)^l(aba) = (aba)^{l+1}$.

B:



Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass die Behauptung für alle m, die echt kleiner als ein *festes* n sind, gilt.

(c) Induktionsschritt: Sei $|w| > 0$, d.h. $w = w_1 \cdots w_n$ mit $w_i \in \Sigma$ und $n > 0$.

Wenn das Wort die Länge 1 oder 2 hat, dann wird es nicht akzeptiert, und es ist im Wenn-Dann-Teil nichts zu zeigen. Sei also $|w| \geq 3$.

Aus den Diagramm kann man erkennen, dass die *einzige* Möglichkeit, mit einem Wort w den Endzustand z_0 zu erreichen, darin besteht, dass die letzten drei Zeichen aba sind und dass man zuvor in dem Zustand z_0 gewesen ist:

$$\delta^*(z_0, w_1 \cdots w_{n-3}) = z_0$$

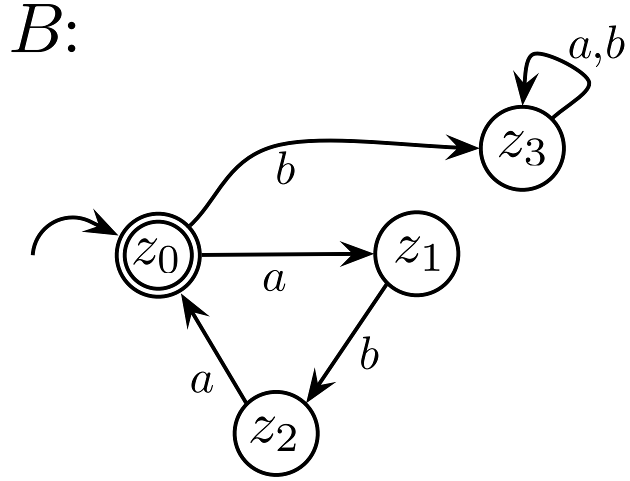
Also gilt für das Teilwort $w_1 \cdots w_{n-3} \in L(B)$.

Da es die Länge $n - 3 < n$ besitzt, ist die IA anwendbar, d.h. es gibt ein l, so dass:

$$(w_1 \cdots w_{n-3}) = (aba)^l$$

Also ist auch w ein Vielfaches von 3, denn $|w| = |(aba)^l(aba)| = 3(l + 1)$ und $w = (aba)^l(aba) = (aba)^{l+1}$.

B:



Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass die Behauptung für alle m, die echt kleiner als ein *festes* n sind, gilt.

(c) Induktionsschritt: Sei $|w| > 0$, d.h. $w = w_1 \cdots w_n$ mit $w_i \in \Sigma$ und $n > 0$.

Wenn das Wort die Länge 1 oder 2 hat, dann wird es nicht akzeptiert, und es ist im Wenn-Dann-Teil nichts zu zeigen. Sei also $|w| \geq 3$.

Aus den Diagramm kann man erkennen, dass die *einzige* Möglichkeit, mit einem Wort w den Endzustand z_0 zu erreichen, darin besteht, dass die letzten drei Zeichen aba sind und dass man zuvor in dem Zustand z_0 gewesen ist:

$$\delta^*(z_0, w_1 \cdots w_{n-3}) = z_0$$

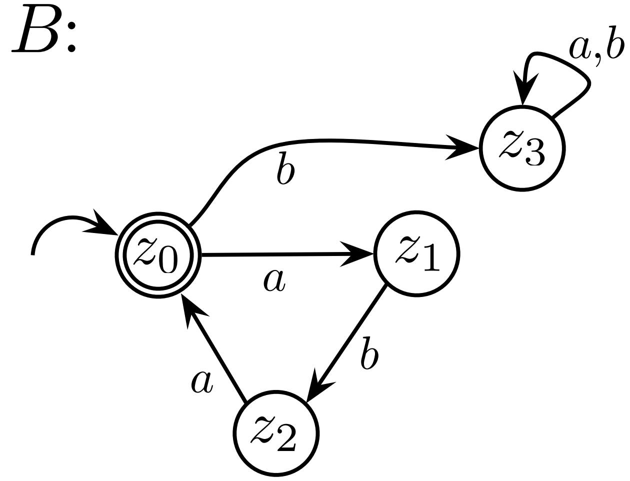
Also gilt für das Teilwort $w_1 \cdots w_{n-3} \in L(B)$.

Da es die Länge $n-3 < n$ besitzt, ist die IA anwendbar, d.h. es gibt ein l, so dass:

$$(w_1 \cdots w_{n-3}) = (aba)^l$$

Also ist auch w ein Vielfaches von 3, denn $|w| = |(aba)^l(aba)| = 3(l+1)$ und $w = (aba)^l(aba) = (aba)^{l+1}$.

B:



Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass die Behauptung für alle m, die echt kleiner als ein *festes* n sind, gilt.

(c) Induktionsschritt: Sei $|w| > 0$, d.h. $w = w_1 \cdots w_n$ mit $w_i \in \Sigma$ und $n > 0$.

Wenn das Wort die Länge 1 oder 2 hat, dann wird es nicht akzeptiert, und es ist im Wenn-Dann-Teil nichts zu zeigen. Sei also $|w| \geq 3$.

Aus den Diagramm kann man erkennen, dass die *einzige* Möglichkeit, mit einem Wort w den Endzustand z_0 zu erreichen, darin besteht, dass die letzten drei Zeichen aba sind und dass man zuvor in dem Zustand z_0 gewesen ist:

$$\delta^*(z_0, w_1 \cdots w_{n-3}) = z_0$$

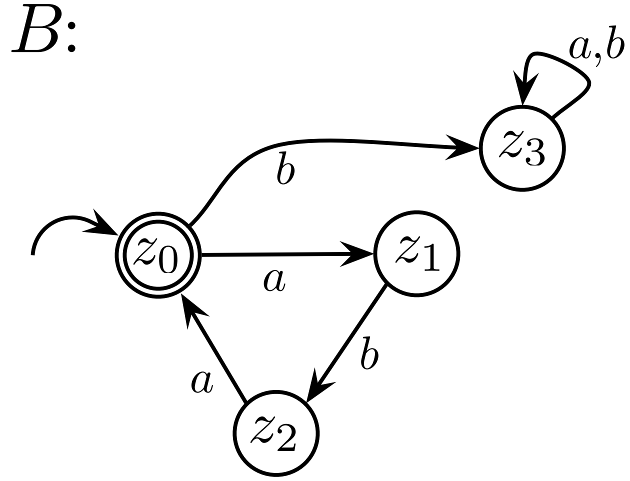
Also gilt für das Teilwort $w_1 \cdots w_{n-3} \in L(B)$.

Da es die Länge $n - 3 < n$ besitzt, ist die IA anwendbar, d.h. es gibt ein l, so dass:

$$(w_1 \cdots w_{n-3}) = (aba)^l$$

Also ist auch w ein Vielfaches von 3, denn $|w| = |(aba)^l(aba)| = 3(l+1)$ und $w = (aba)^l(aba) = (aba)^{l+1}$.

B:



Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass die Behauptung für alle m, die echt kleiner als ein *festes* n sind, gilt.

(c) Induktionsschritt: Sei $|w| > 0$, d.h. $w = w_1 \cdots w_n$ mit $w_i \in \Sigma$ und $n > 0$.

Wenn das Wort die Länge 1 oder 2 hat, dann wird es nicht akzeptiert, und es ist im Wenn-Dann-Teil nichts zu zeigen. Sei also $|w| \geq 3$.

Aus den Diagramm kann man erkennen, dass die *einzige* Möglichkeit, mit einem Wort w den Endzustand z_0 zu erreichen, darin besteht, dass die letzten drei Zeichen aba sind und dass man zuvor in dem Zustand z_0 gewesen ist:

$$\delta^*(z_0, w_1 \cdots w_{n-3}) = z_0$$

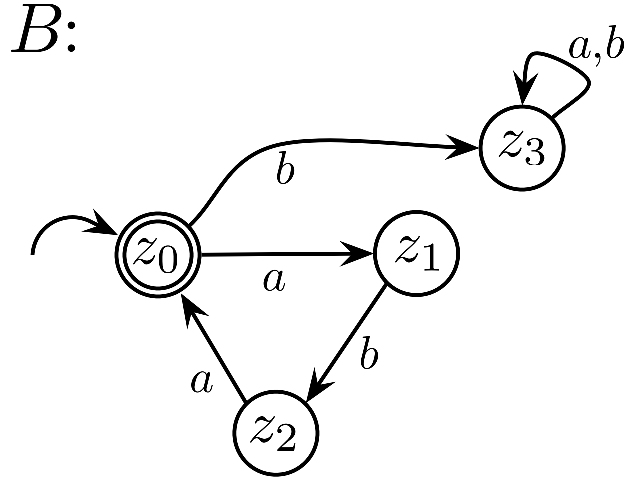
Also gilt für das Teilwort $w_1 \cdots w_{n-3} \in L(B)$.

Da es die Länge $n - 3 < n$ besitzt, ist die IA anwendbar, d.h. es gibt ein l, so dass:

$$(w_1 \cdots w_{n-3}) = (aba)^l$$

Also ist auch w ein Vielfaches von 3, denn $|w| = |(aba)^l(aba)| = 3(l+1)$ und $w = (aba)^l(aba) = (aba)^{l+1}$.

B:

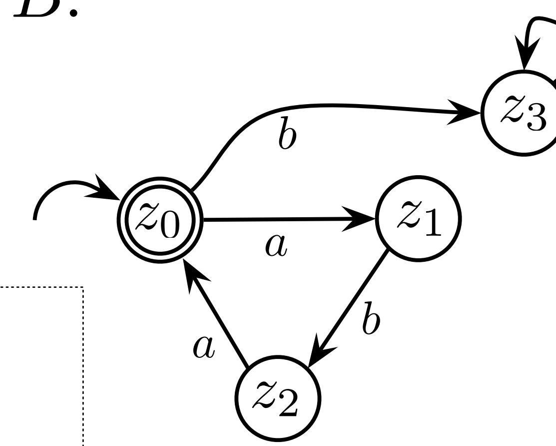


Variante 2: L(B) = L (Fortsetzung)

Damit ist die Induktionsbehauptung bewiesen.

Es folgt direkt: Wenn $w \in L(B)$, dann ist $w = (aba)^k \in \{aba\}^* = L$

B:



2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist.

   Wir zeigen per Induktion über n, dass für jedes n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

   (a) Induktionsanfang: Für $n = 0$ gilt:

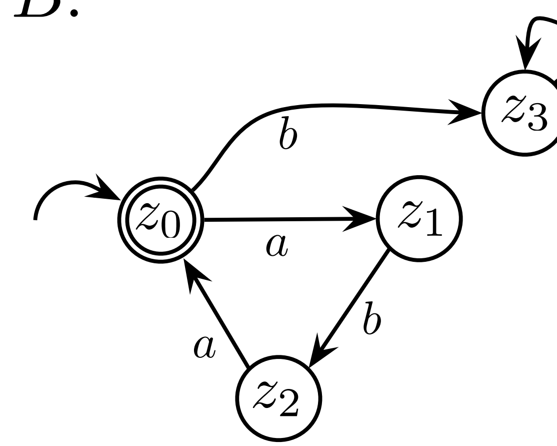
$$\delta^*(z_0, (aba)^0) = \delta^*(z_0, \epsilon) = z_0 \in F$$

Variante 2: L(B) = L (Fortsetzung)

Damit ist die Induktionsbehauptung bewiesen.

Es folgt direkt: Wenn $w \in L(B)$, dann ist $w = (aba)^k \in \{aba\}^* = L$

B:



2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist.

   Wir zeigen per Induktion über n, dass für jedes n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

   (a) Induktionsanfang: Für $n = 0$ gilt:

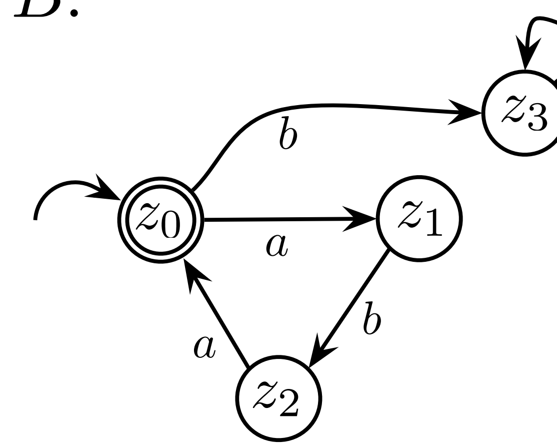
   $$\delta^*(z_0, (aba)^0) = \delta^*(z_0, \epsilon) = z_0 \in F$$

Variante 2: L(B) = L (Fortsetzung)

Damit ist die Induktionsbehauptung bewiesen.

Es folgt direkt: Wenn $w \in L(B)$, dann ist $w = (aba)^k \in \{aba\}^* = L$

B:



2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist.

   Wir zeigen per Induktion über n, dass für jedes n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

   (a) Induktionsanfang: Für $n = 0$ gilt:

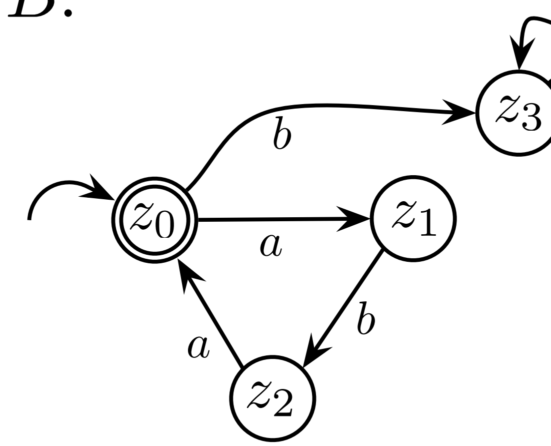
   $$\delta^*(z_0, (aba)^0) = \delta^*(z_0, \epsilon) = z_0 \in F$$

Variante 2: L(B) = L (Fortsetzung)

Damit ist die Induktionsbehauptung bewiesen.

Es folgt direkt: Wenn $w \in L(B)$, dann ist $w = (aba)^k \in \{aba\}^* = L$

B:



2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist.

   Wir zeigen per Induktion über n, dass für jedes n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

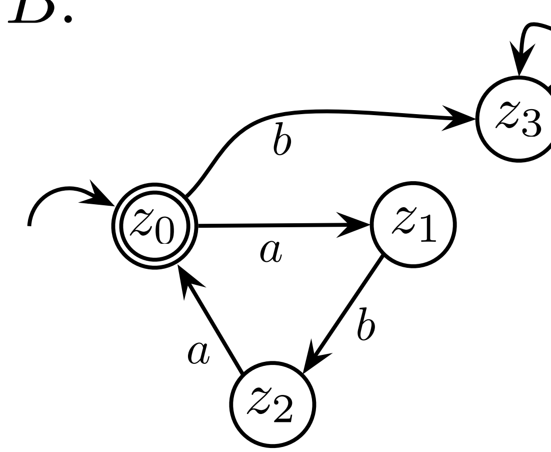
   (a) Induktionsanfang: Für $n = 0$ gilt:

   $$\delta^*(z_0, (aba)^0) = \delta^*(z_0, \epsilon) = z_0 \in F$$

Damit ist die Induktionsbehauptung bewiesen.

Es folgt direkt: Wenn $w \in L(B)$, dann ist $w = (aba)^k \in \{aba\}^* = L$

B:



2. $L \subseteq L(B)$ Sei $w \in L$. Wir müssen $w \in L(B)$ zeigen.

   Wenn $w \in L = \{aba\}^*$ ist, dann muss es ein $n \in \mathbb{N}$ geben, so dass $w = (aba)^n$ ist.

   Wir zeigen per Induktion über n, dass für jedes n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

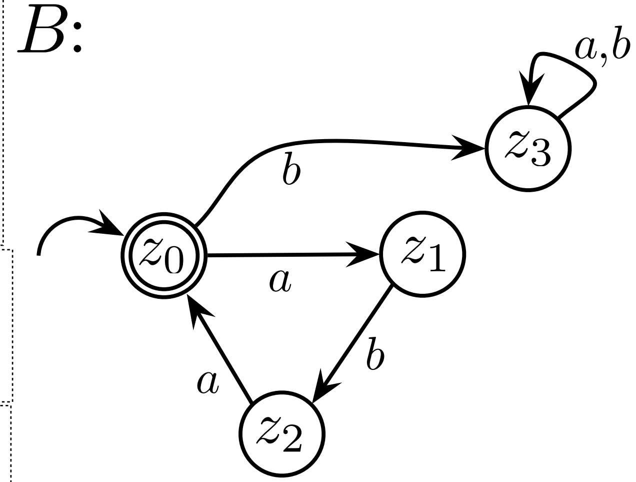
   (a) Induktionsanfang: Für $n = 0$ gilt:

$$\delta^*(z_0, (aba)^0) = \delta^*(z_0, \epsilon) = z_0 \in F$$

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

B:



(c) Induktionsschritt von n auf $n+1$:
Wir betrachten also das Wort $w = (aba)^{n+1}$.

$$\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

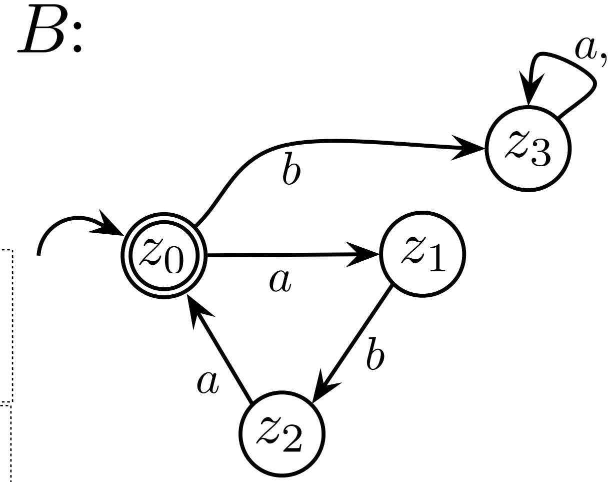
Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

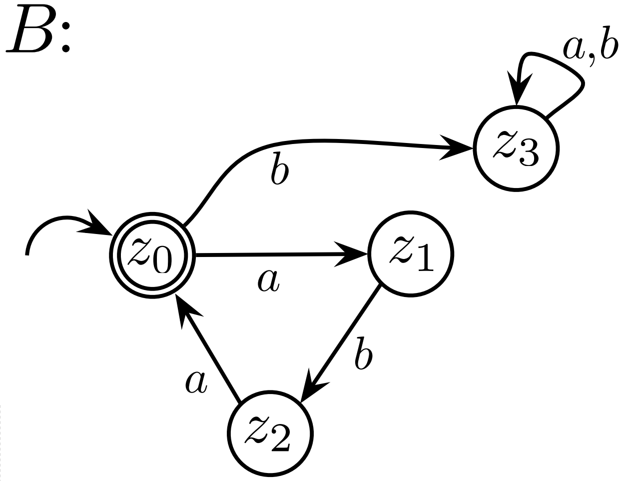
Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:
Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{aligned}
& \delta^*(z_0, (aba)^{n+1}) \\
= \quad & \delta^*(z_0, aba(aba)^n) \\
= \quad & \delta^*(\delta(z_0, a), ba(aba)^n) \quad = \quad \delta^*(z_1, ba(aba)^n) \\
= \quad & \delta^*(\delta(z_1, b), a(aba)^n) \quad = \quad \delta^*(z_2, ba(aba)^n) \\
= \quad & \delta^*(\delta(z_2, a), (aba)^n) \quad = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} = \quad & z_0 \in F
\end{aligned}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

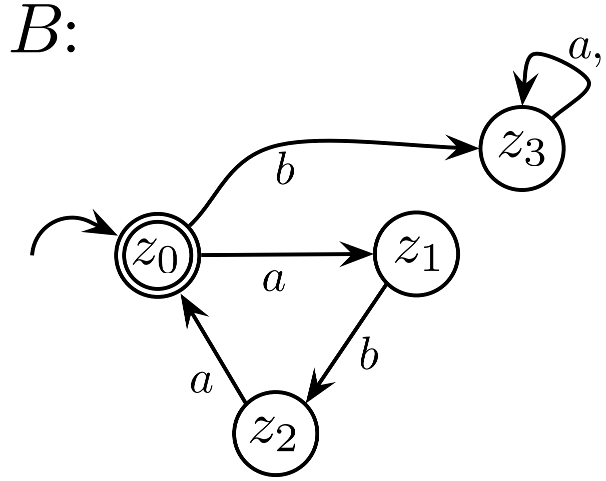
Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

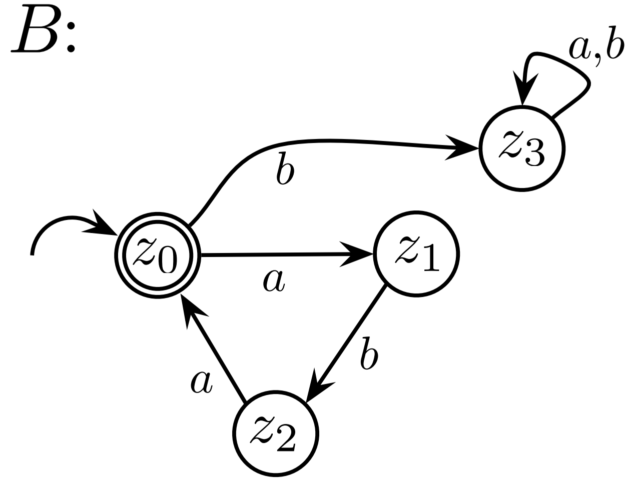
Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

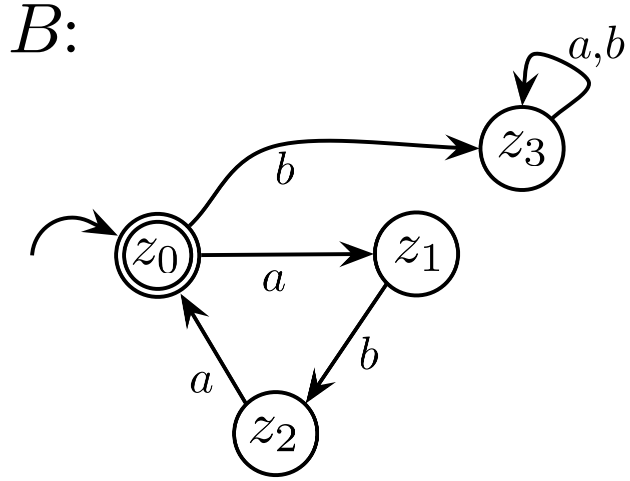
Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0,(aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:
Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{array}{rcll}
 & & \delta^*(z_0,(aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0,a), ba(aba)^n) & = \ \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1,b), a(aba)^n) & = \ \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2,a), (aba)^n) & = \ \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0,(aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

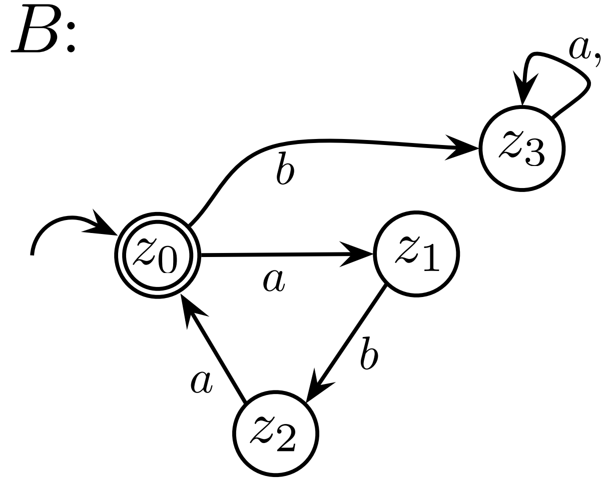
Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

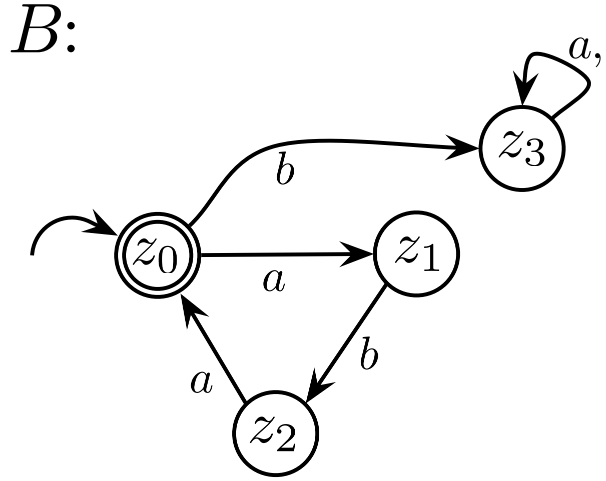
Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:
Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

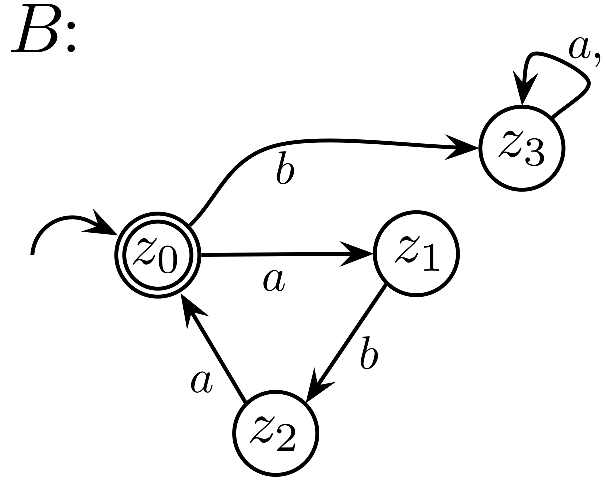
Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$.

B:



$$
\begin{aligned}
& & & \delta^*(z_0, (aba)^{n+1}) & & \\
& & = \; & \delta^*(z_0, aba(aba)^n) & & \\
& & = \; & \delta^*(\delta(z_0, a), ba(aba)^n) & = \; & \delta^*(z_1, ba(aba)^n) \\
& & = \; & \delta^*(\delta(z_1, b), a(aba)^n) & = \; & \delta^*(z_2, ba(aba)^n) \\
& & = \; & \delta^*(\delta(z_2, a), (aba)^n) & = \; & \delta^*(z_0, (aba)^n) \\
& \text{nach IA} & = \; & z_0 \in F & &
\end{aligned}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

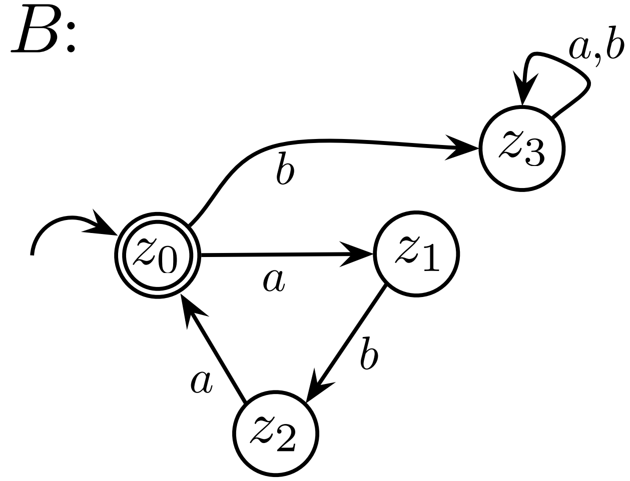
Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

B:



(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$.

$$\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

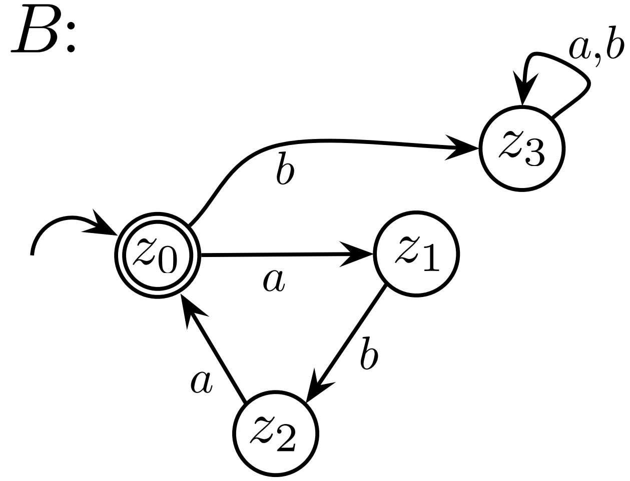
Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

B:



(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$.

$$
\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt. **qed.**

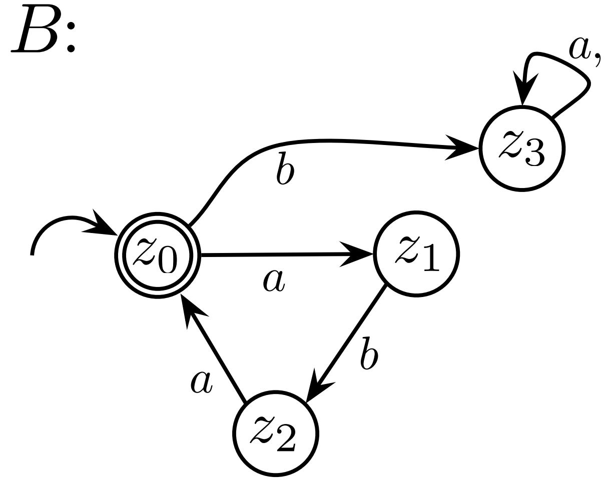
Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$

B:



War das formal genug?

$$
\begin{array}{rcll}
 & & \delta^*(z_0, (aba)^{n+1}) & \\
 & = & \delta^*(z_0, aba(aba)^n) & \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) & = \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) & = \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) & = \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F &
\end{array}
$$

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

qed.

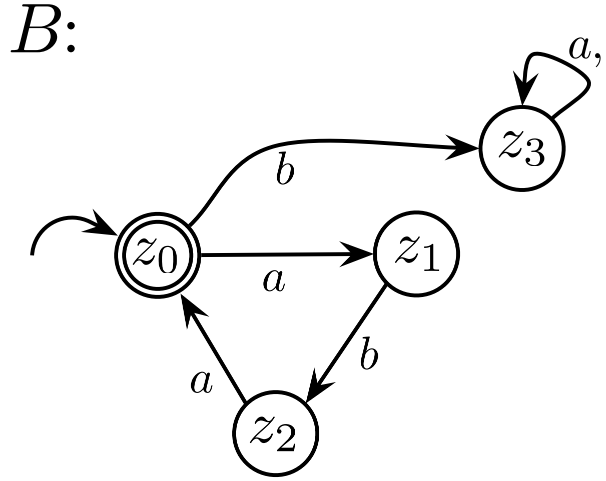
Variante 2: L(B) = L (Fortsetzung)

(b) Induktionsannahme (IA): Wir nehmen, dass für ein *festes* n auch $\delta^*(z_0, (aba)^n) = z_0$ gilt.

(c) Induktionsschritt von n auf $n+1$:

Wir betrachten also das Wort $w = (aba)^{n+1}$

B:



War das formal genug?

$$
\begin{array}{rcl}
 & & \delta^*(z_0, (aba)^{n+1}) \\
 & = & \delta^*(z_0, aba(aba)^n) \\
 & = & \delta^*(\delta(z_0, a), ba(aba)^n) \quad = \quad \delta^*(z_1, ba(aba)^n) \\
 & = & \delta^*(\delta(z_1, b), a(aba)^n) \quad = \quad \delta^*(z_2, ba(aba)^n) \\
 & = & \delta^*(\delta(z_2, a), (aba)^n) \quad = \quad \delta^*(z_0, (aba)^n) \\
\text{nach IA} & = & z_0 \in F
\end{array}
$$

...zu formal?...

Also wird $(aba)^{n+1}$ akzeptiert, d.h. $(aba)^{n+1} \in L(B)$.

Damit ist die Induktion abgeschlossen, d.h. für jedes n gilt $\delta^*(z_0, (aba)^n) = z_0$.

Da $z_0 \in F$ gilt, haben wir $w = (aba)^n \in L(B)$ gezeigt.

qed.

FGI 1

Automaten, Formale Sprachen, Berechenbarkeit

Kap. 13

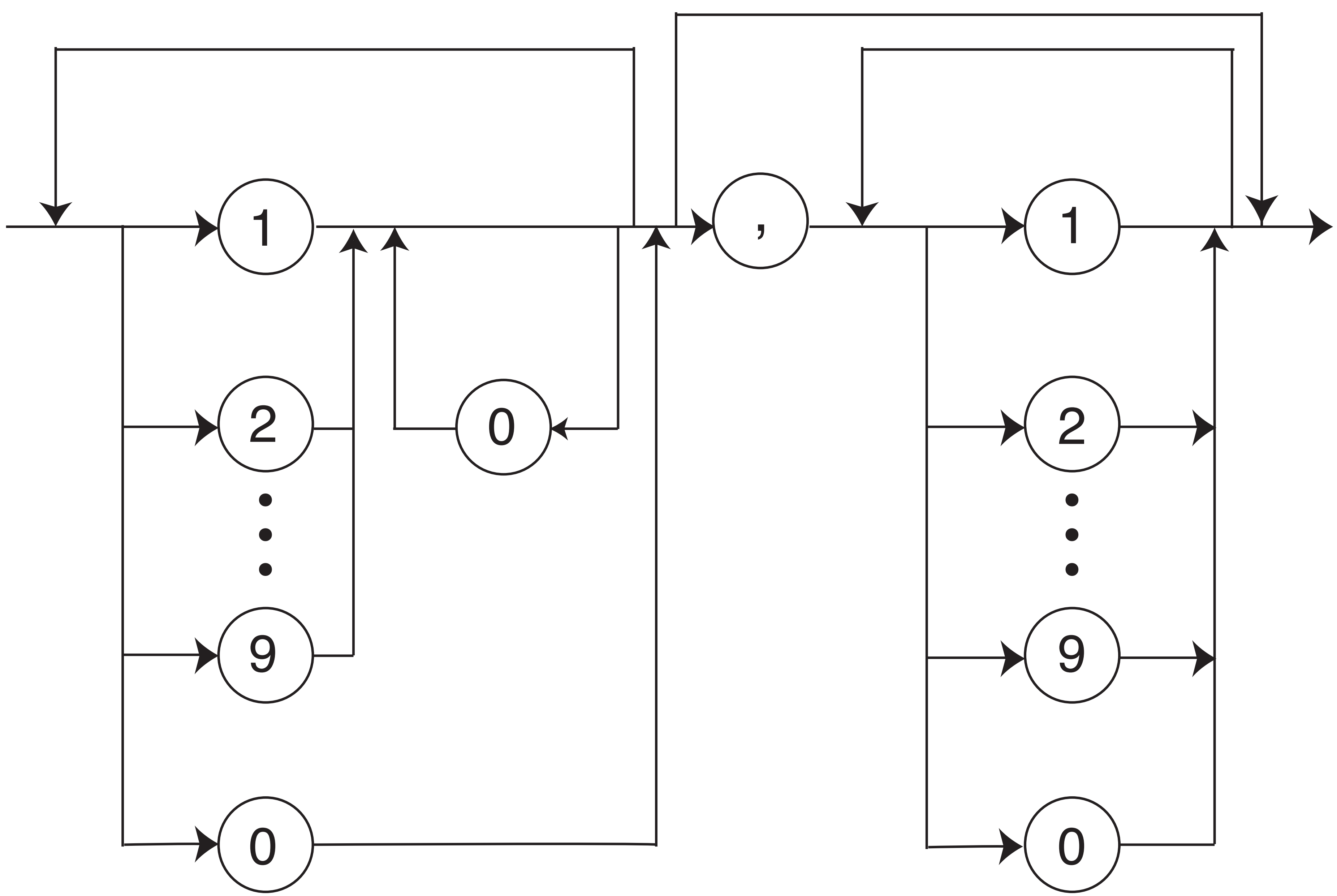
Nichtdeteministischer Automat, Potenzautomat

Michael Köhler-Bußmeier

Nichtdeterminismus: warum und wie?

Z.B. darum: (Syntaxdiagramm)

Dezimalzahl:



nichtdeterministischer endl. Automat: NFA

Definition 13.5:

Ein **nichtdeterministischer endlicher Automat** (*NFA*, engl. ***n****ondeterministic* ***f****inite* ***a****utomaton*) wird spezifiziert werden durch ein Tupel $A := (Q, \Sigma, \delta, S_0, F)$, mit:

- Q ist endliche Menge von **Zuständen**,
- Σ ist endliches Alphabet von **Eingabesymbolen**,
- $\delta : Q \times \Sigma \longrightarrow 2^Q$ ist **Überführungsfunktion**,
- $S_0 \subseteq Q$ ist **Menge der Startzustände** und
- $F \subseteq Q$ ist **Menge der Endzustände**.

akzeptierte Sprache eines NFA

Definition 13.6:

- Zu einem *NFA* $A := (Q, \Sigma, \delta, S_0, F)$ definieren wir die **erweiterte Überführungsfunktion**

$$\delta^* : 2^Q \times \Sigma^* \longrightarrow 2^Q$$

durch:

$$\delta^*(R, aw) := \delta^*\left(\overbrace{\bigcup_{p \in R} \delta(p, a)}^{\text{Menge der Zustände, die mit } a \text{ von irgendeinem Zustand } p \text{ aus der Menge } R \text{ erreicht werden.}}, w\right)$$

$\forall R \subseteq Q$, $\forall a \in \Sigma$ und $\forall w \in \Sigma^*$, sowie

$$\forall\, R \subseteq Q : \delta^*(R, \epsilon) := R.$$

- Die von A **akzeptierte Sprache** ist die Menge

$$L(A) := \{w \in \Sigma^* \mid \delta^*(S_0, w) \cap F \neq \emptyset\}.$$

Nichtdeterminismus deterministisch simulieren

Satz 13.1:

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 die Konstruktion des sogenannten Potenzautomaten.

- Die Zahl der in einem Schritt mit einem Eingabesymbol erreichbaren Folgezustände ist für jeden Zustand endlich.
- Es gibt nur endlich viele Teilmengen von Q, und bei S_0 angefangen kann jeweils eine neue Teilmenge von Q erreicht werden.
- Diese werden durch die Zustände eines neuen DFA repräsentiert:

der Potenzautomat

Beweis durch Konstruktion des Potenzautomaten:

Definition 13.7

- Sei $A := (Q, \Sigma, \delta, S_0, F)$ ein *NFA*.
- Der **Potenzautomat** zu A ist wie folgt definiert:
  - $B := (2^Q, \Sigma, \delta', S_0, F')$ mit
  - $F' := \{M \in 2^Q \mid M \cap F \neq \emptyset\}$ und
  - S_0, d.h. die Menge der Startzustände von A bildet nun den Startzustand von B.
  - δ' ist für alle $M \in 2^Q$ und jedes $x \in \Sigma$ definiert durch:
  $$\delta'(M, x) := \bigcup_{p \in M} \delta(p, x).$$

Korrektheitsbeweis (Idee)

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.

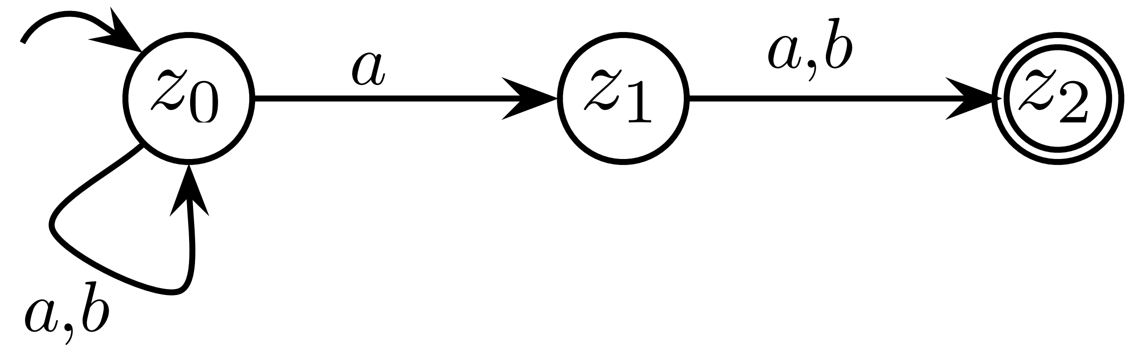
- Andererseits: Für jeden **Erfolgspfad** von A existiert eine **akzeptierende Rechnung** in B.

  – 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.

Beispiel 1

Gegeben sei dieser *NFA*:



Konstruktion des Potenzautomaten entsprechend des Beweises:

$$\emptyset$$

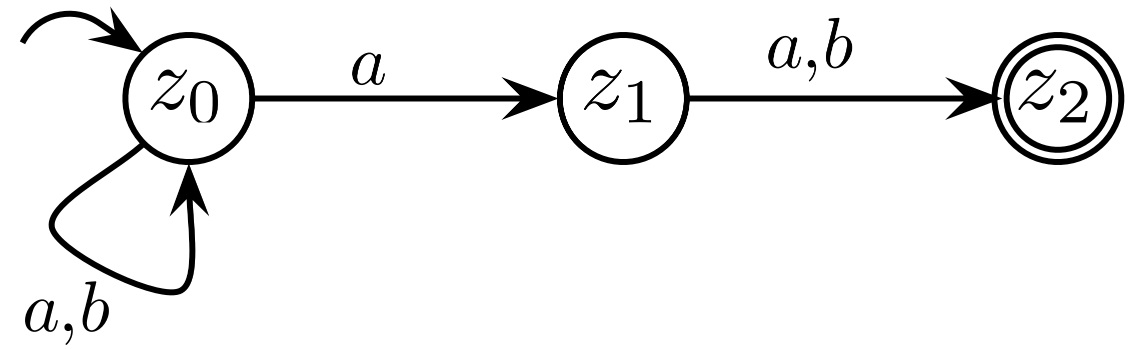
$$\{z_0\} \qquad \{z_1\} \qquad \{z_2\}$$

$$\{z_0, z_1\} \qquad \{z_0, z_2\} \qquad \{z_1, z_2\}$$

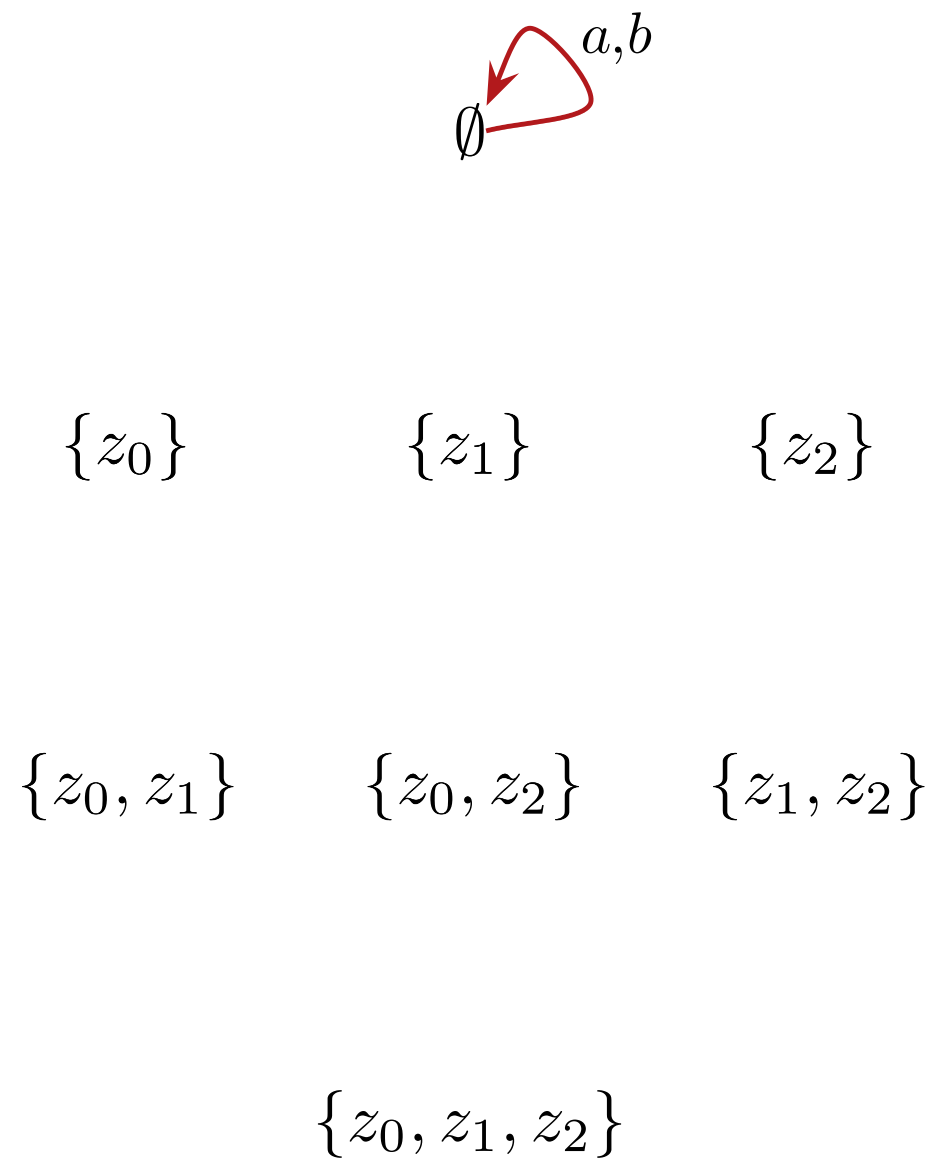
$$\{z_0, z_1, z_2\}$$

Beispiel 1

Gegeben sei dieser *NFA*:

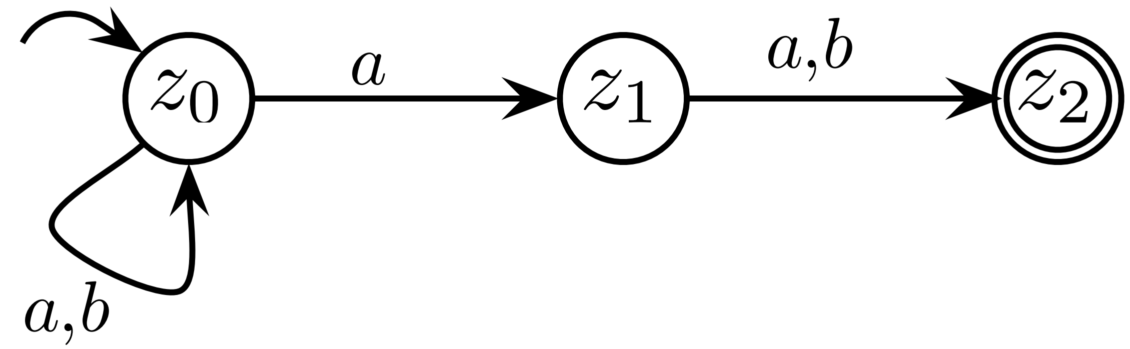


Konstruktion des Potenzautomaten entsprechend des Beweises:

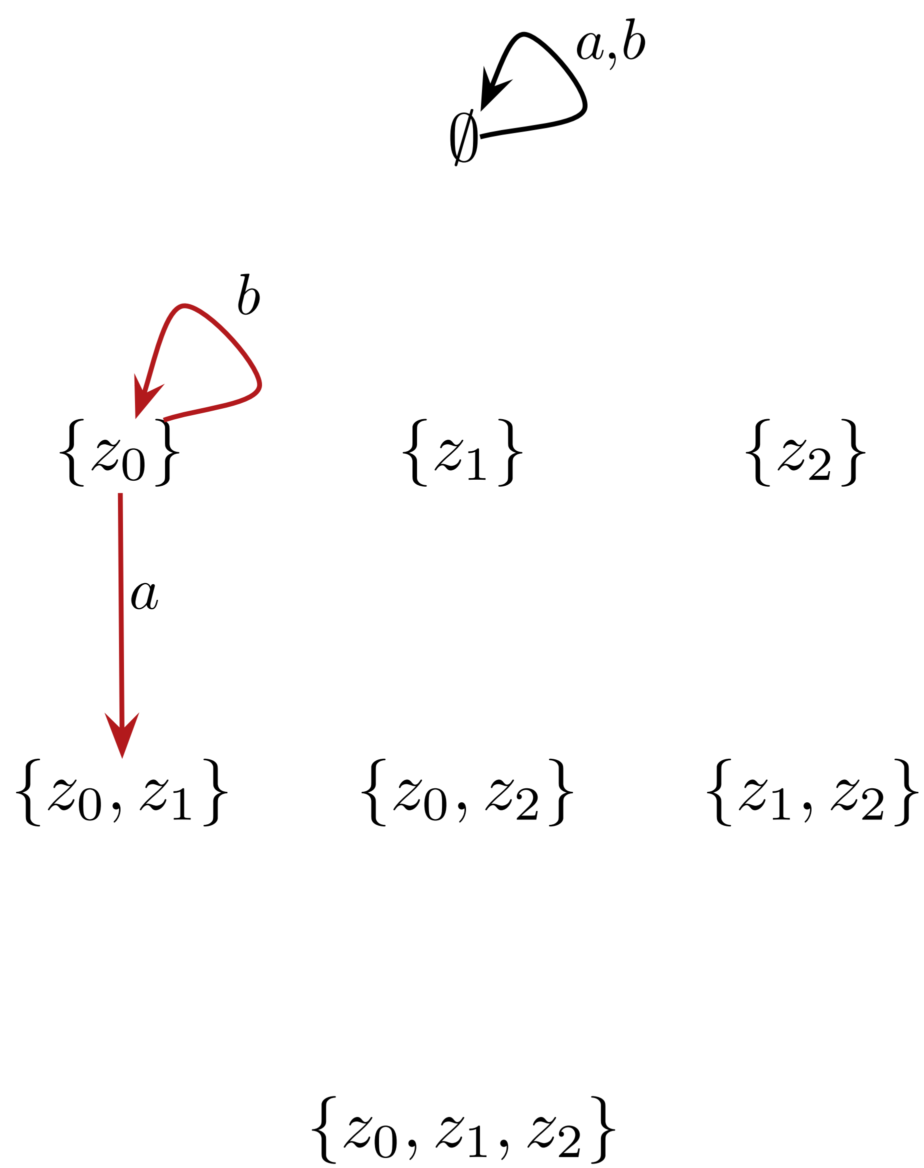


Beispiel 1

Gegeben sei dieser *NFA*:

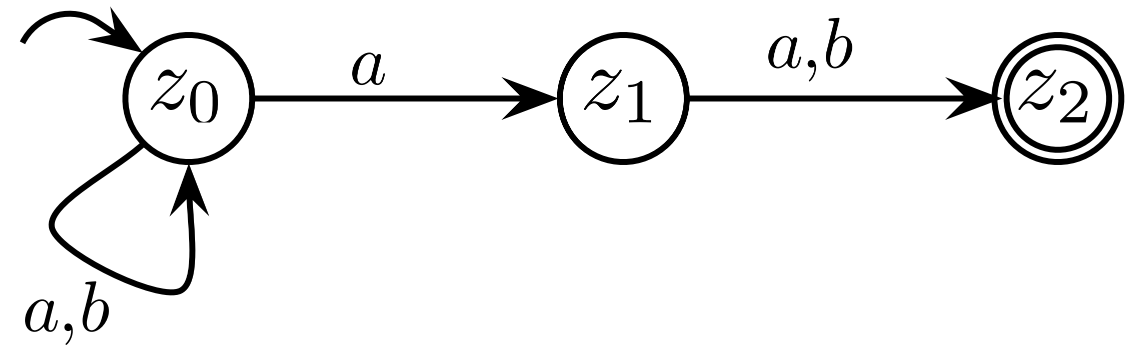


Konstruktion des Potenzautomaten entsprechend des Beweises:

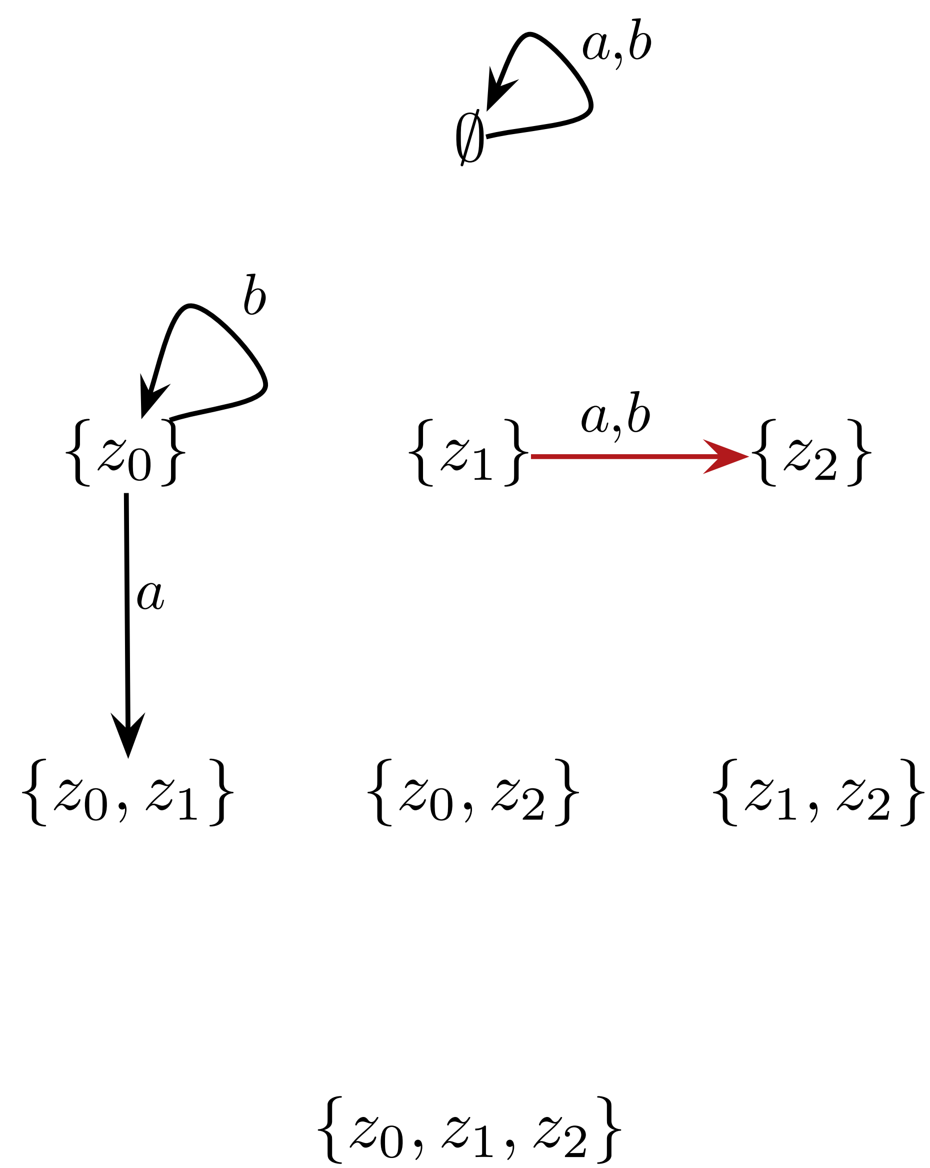


Beispiel 1

Gegeben sei dieser *NFA*:

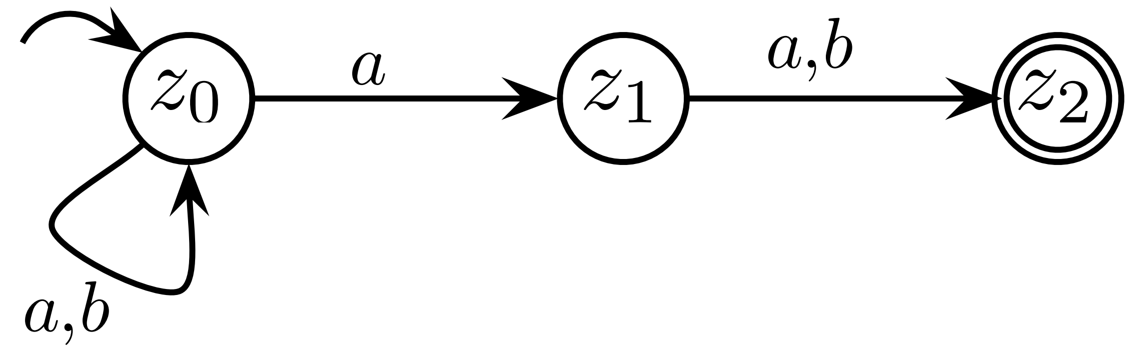


Konstruktion des Potenzautomaten entsprechend des Beweises:

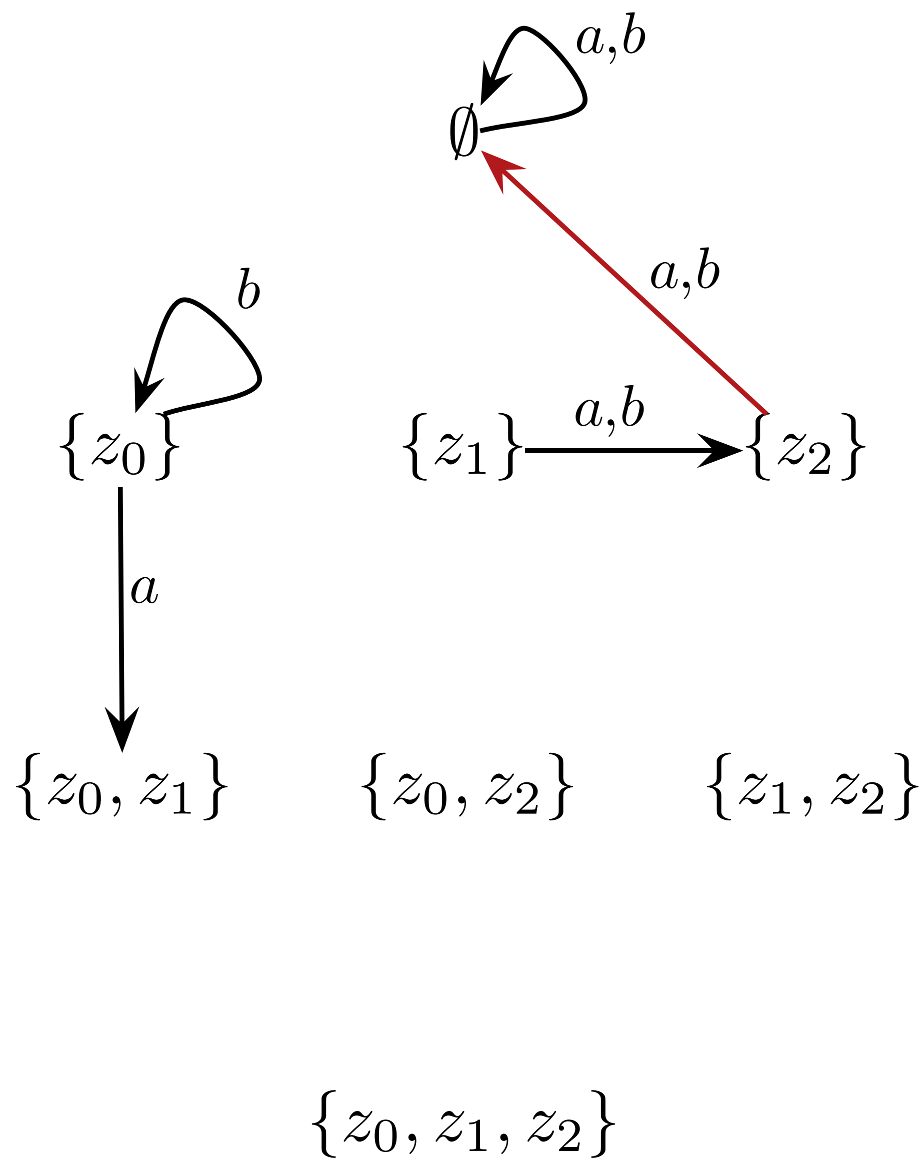


Beispiel 1

Gegeben sei dieser *NFA*:

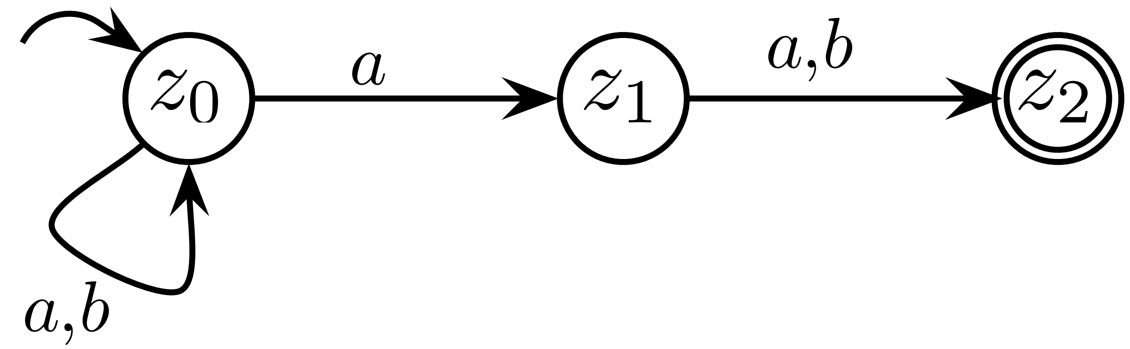


Konstruktion des Potenzautomaten entsprechend des Beweises:

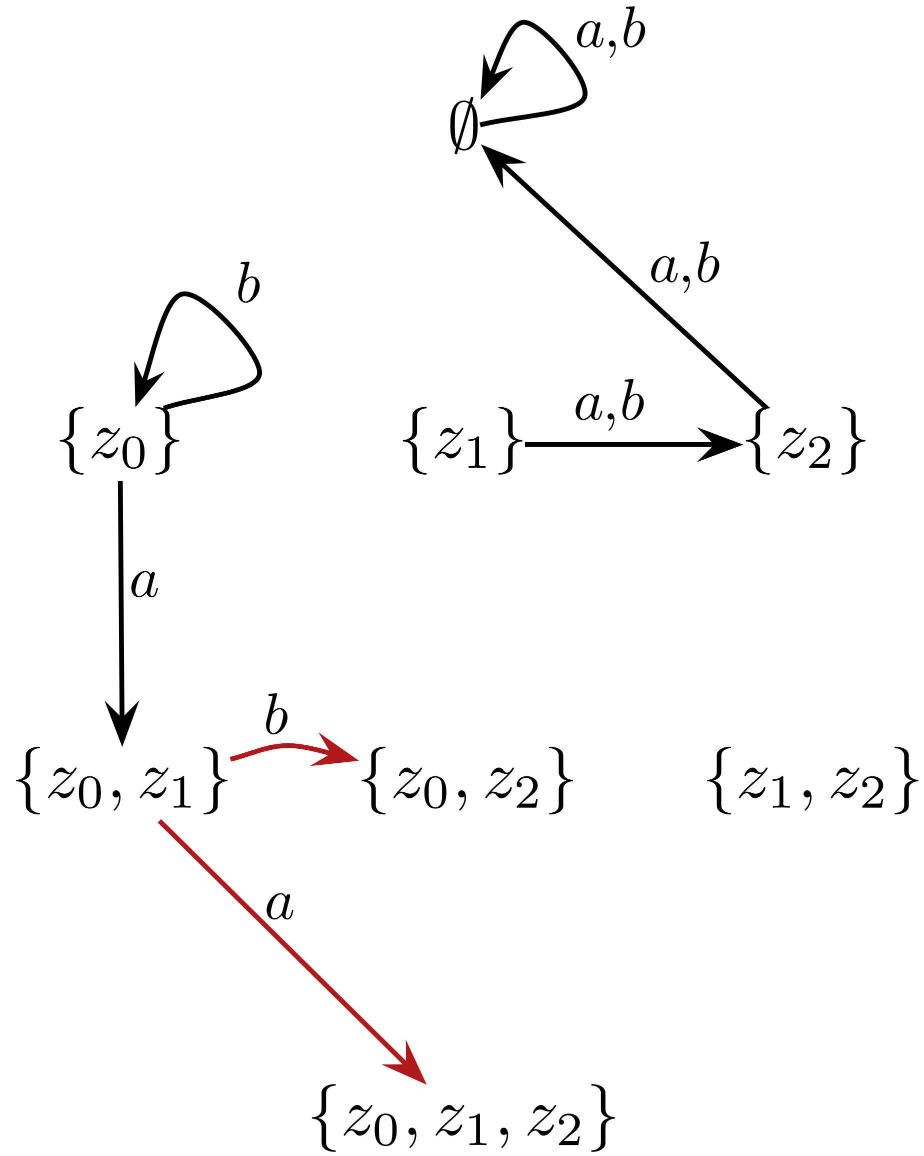


Beispiel 1

Gegeben sei dieser *NFA*:

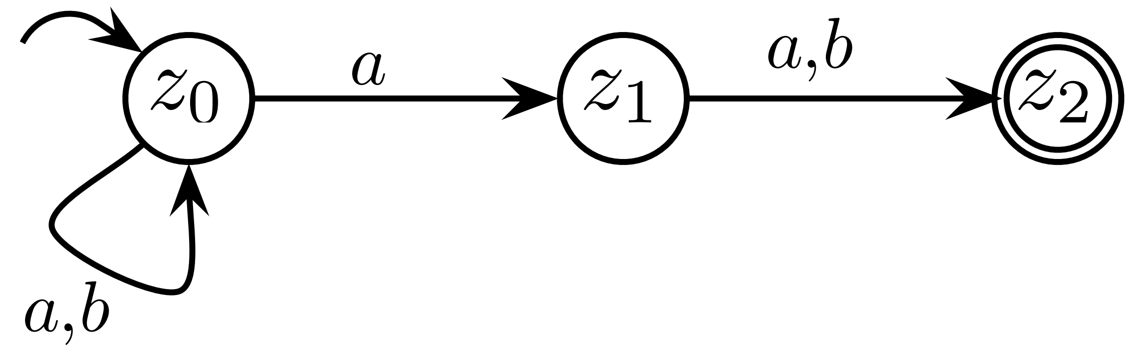


Konstruktion des Potenzautomaten entsprechend des Beweises:

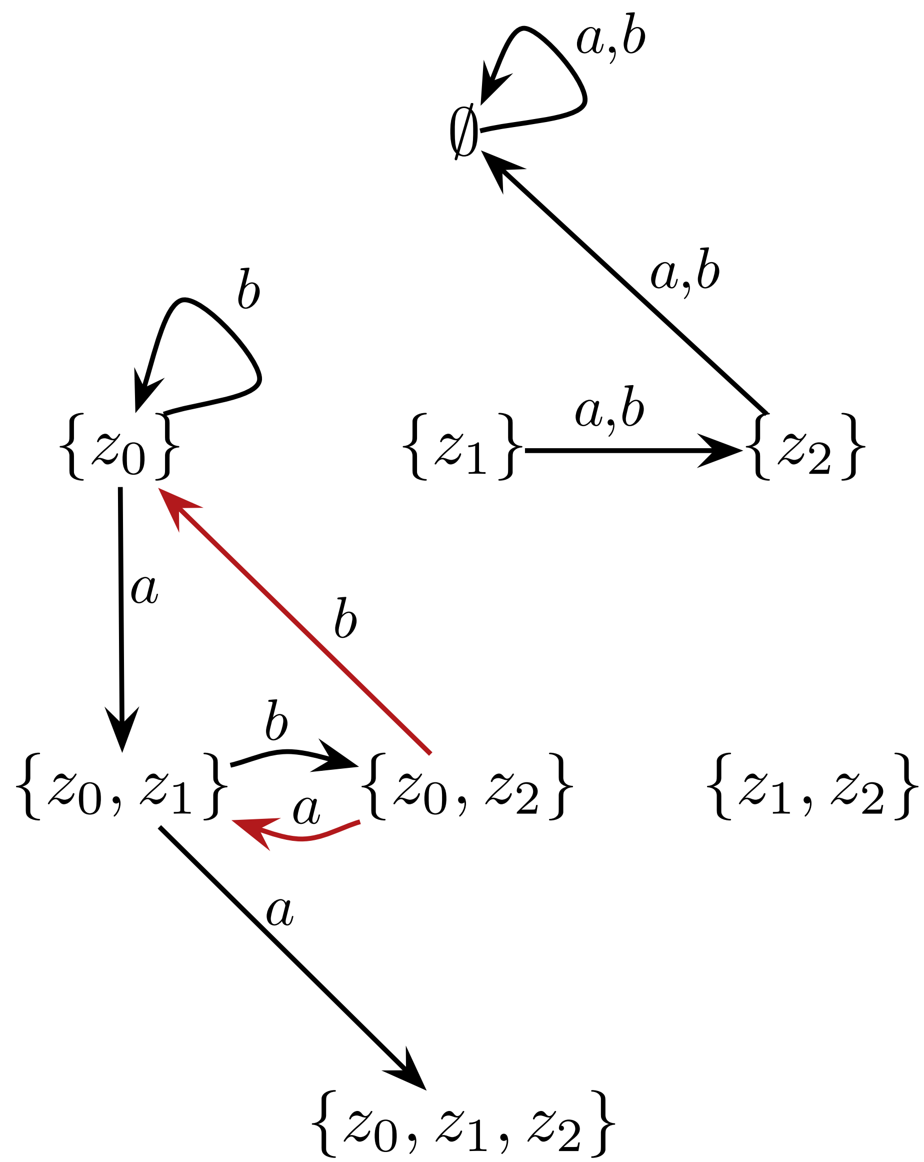


Beispiel 1

Gegeben sei dieser *NFA*:

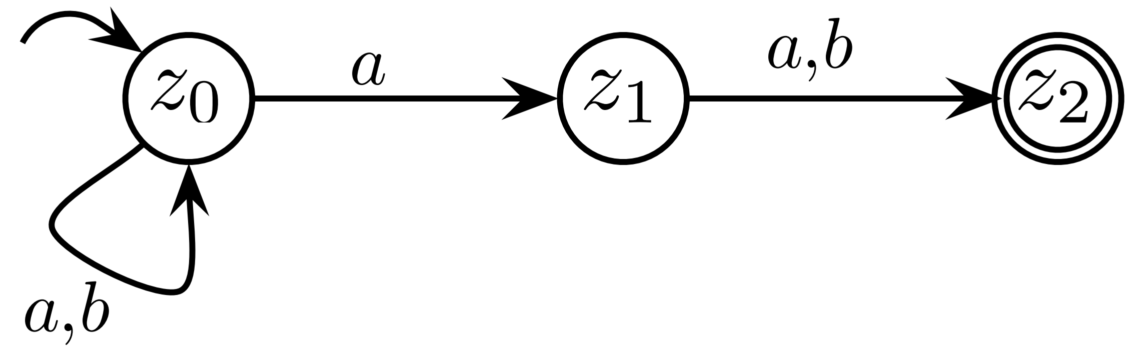


Konstruktion des Potenzautomaten entsprechend des Beweises:

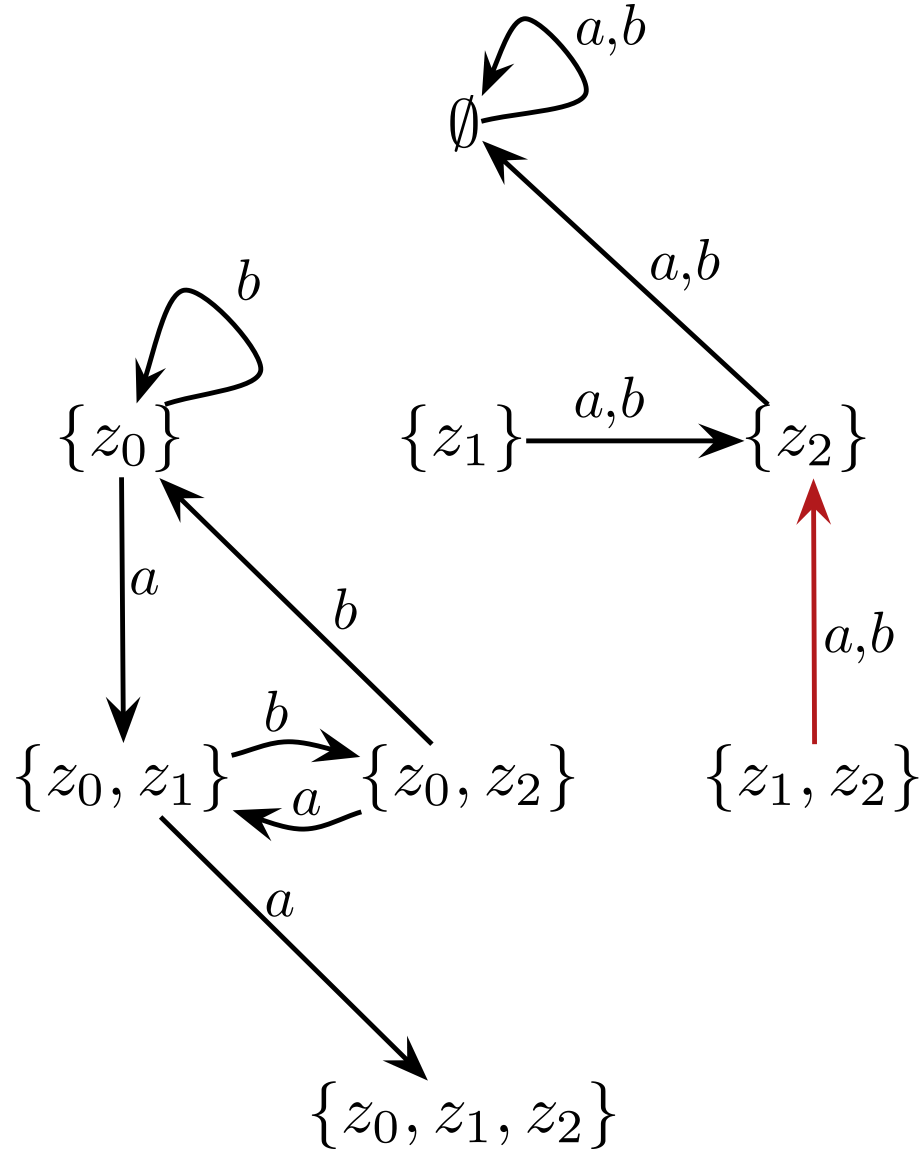


Beispiel 1

Gegeben sei dieser *NFA*:

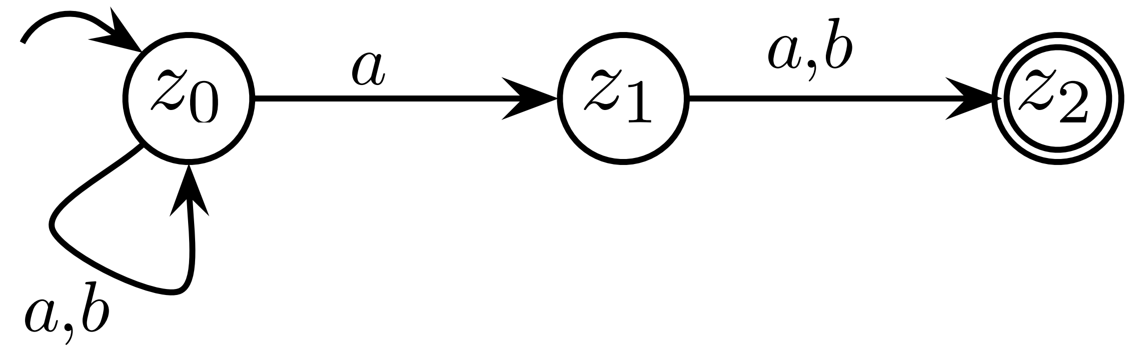


Konstruktion des Potenzautomaten entsprechend des Beweises:

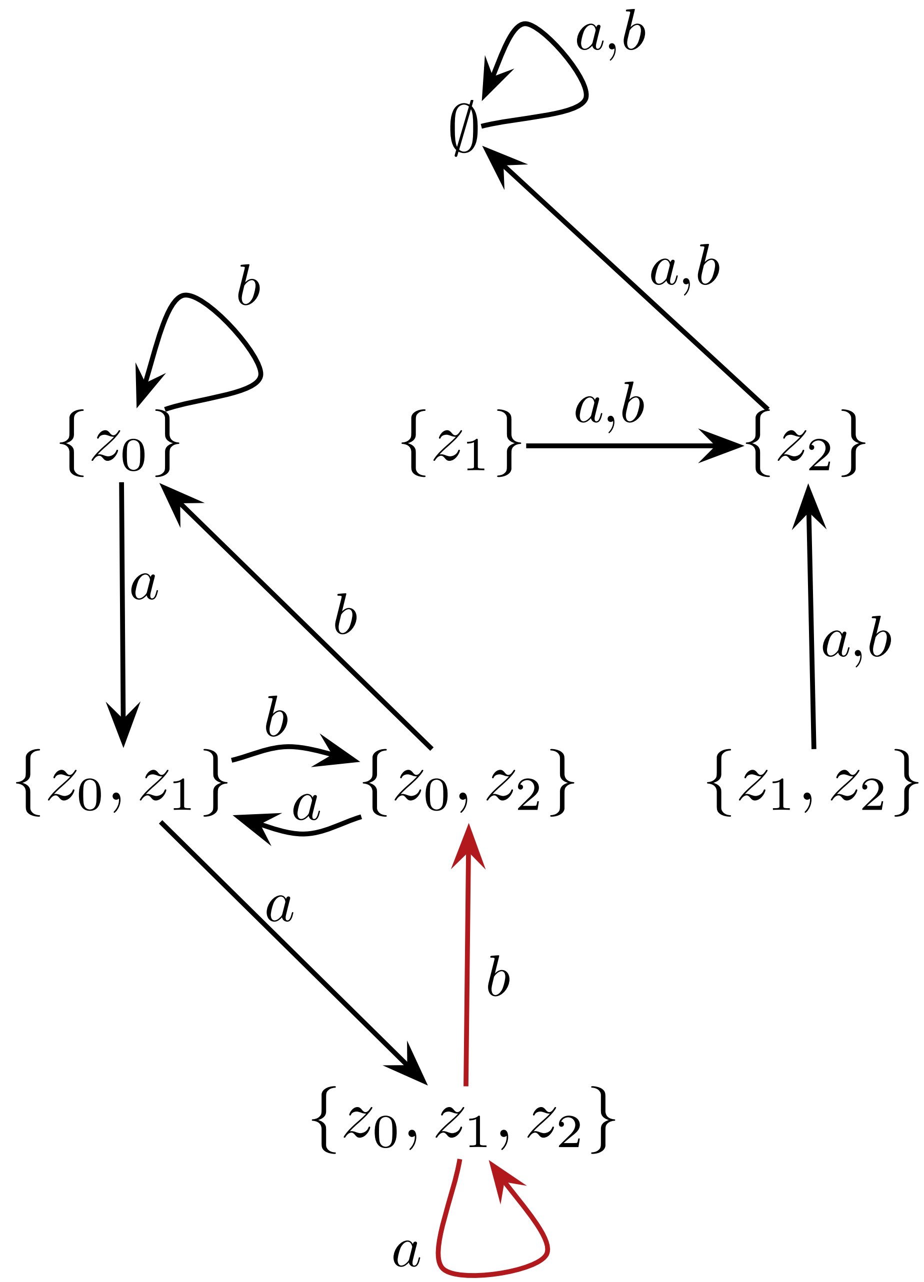


Beispiel 1

Gegeben sei dieser *NFA*:

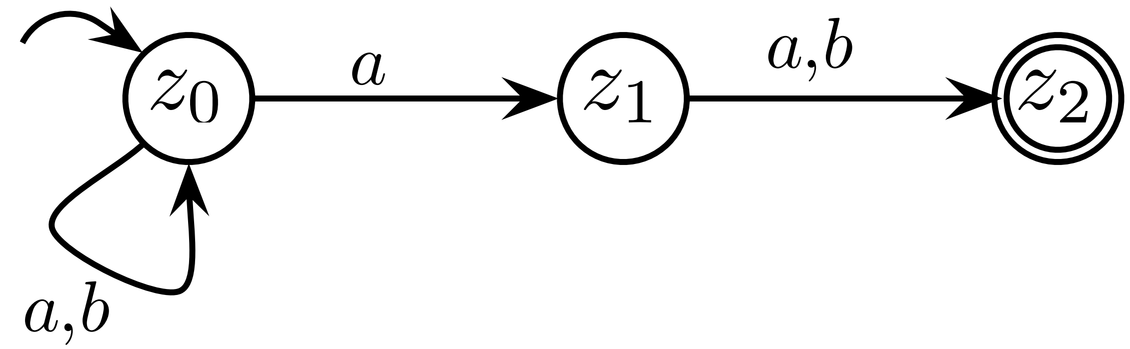


Konstruktion des Potenzautomaten entsprechend des Beweises:

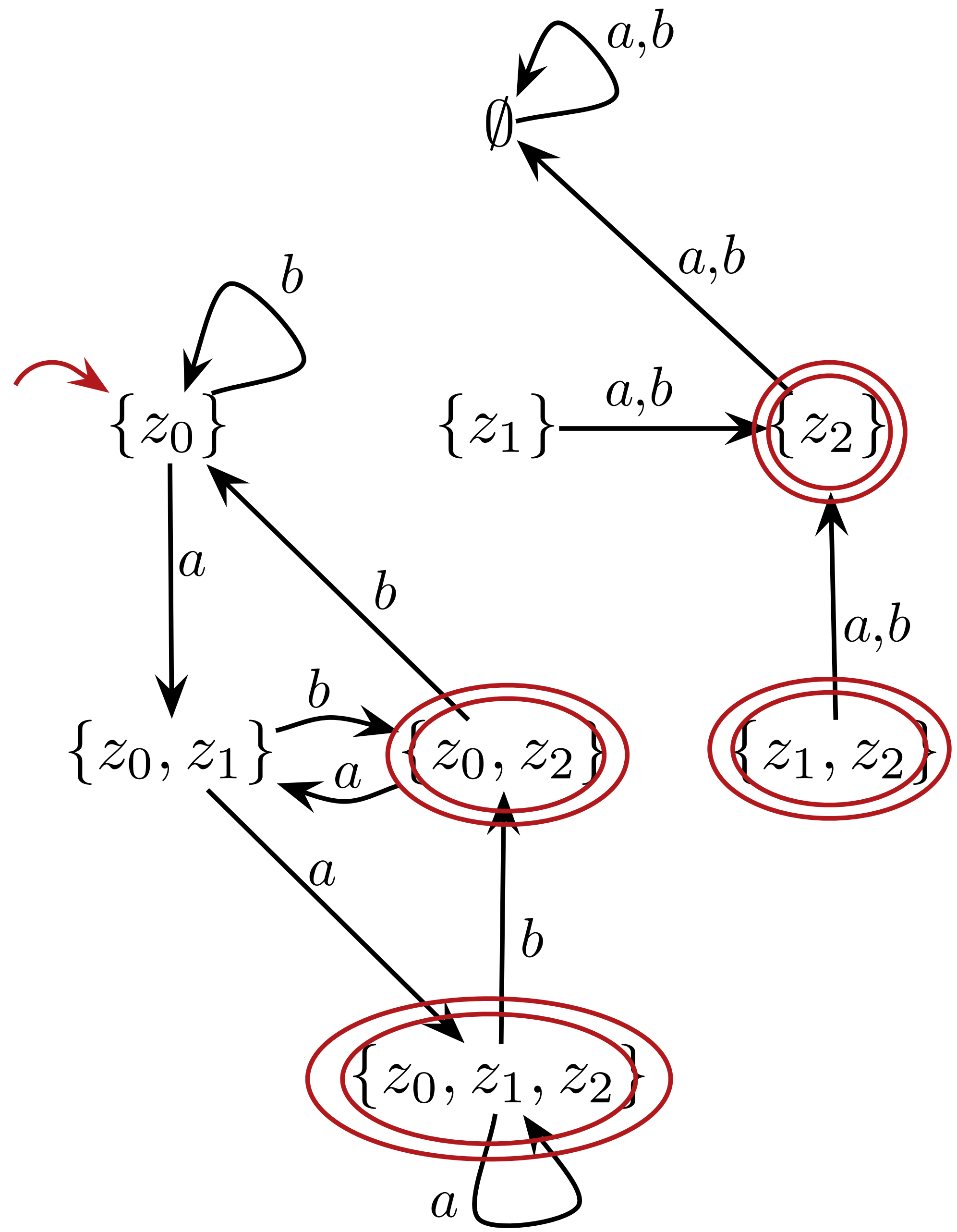


Beispiel 1

Gegeben sei dieser *NFA*:

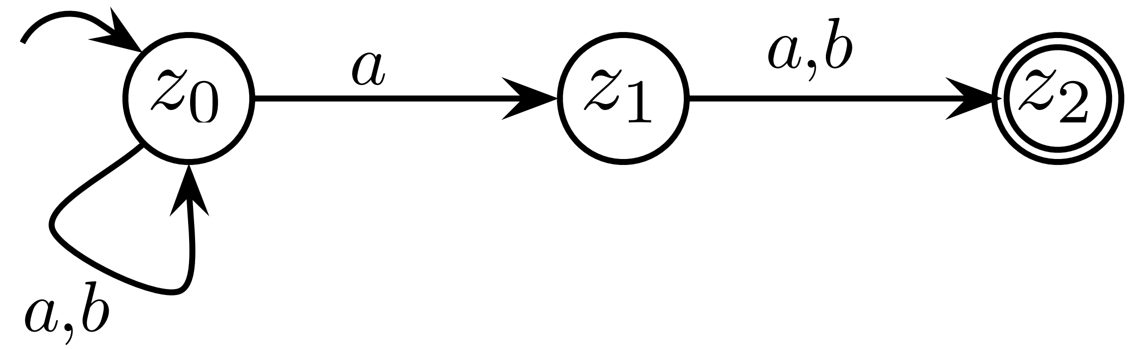


Konstruktion des Potenzautomaten entsprechend des Beweises:



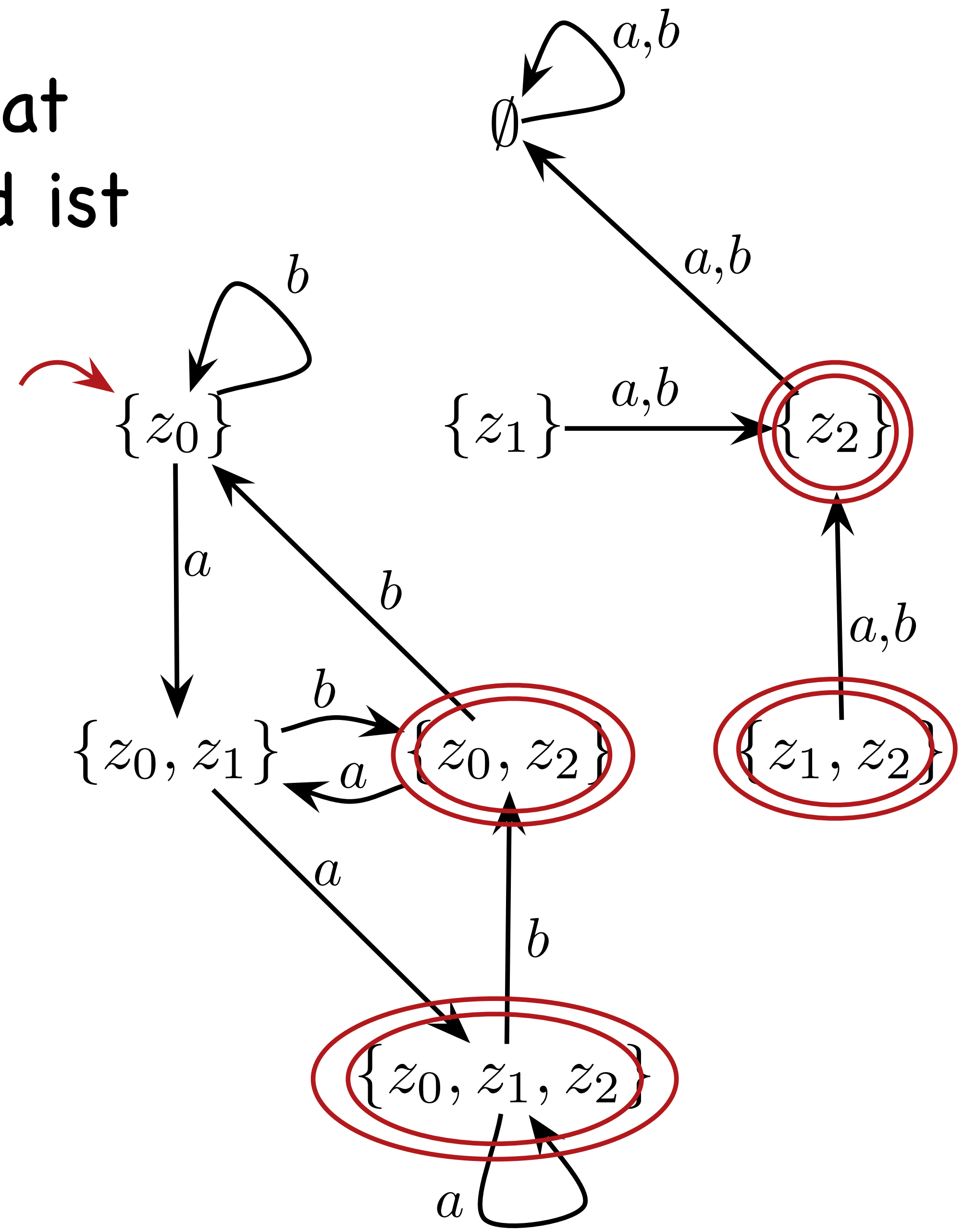
Beispiel 1

Gegeben sei dieser *NFA*:



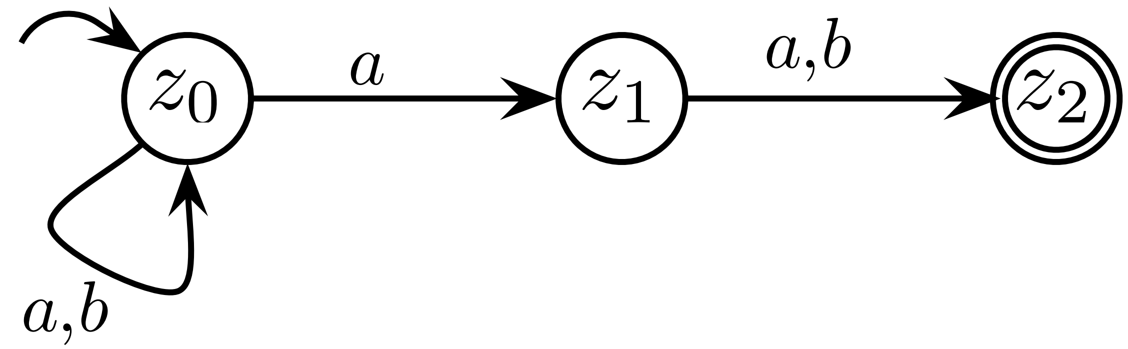
Konstruktion des Potenzautomaten entsprechend des Beweises:

Der äquivalente DFA hat zwei Komponenten und ist nicht (initial) zusammenhängend!



Beispiel 2

Gegeben sei dieser *NFA*:



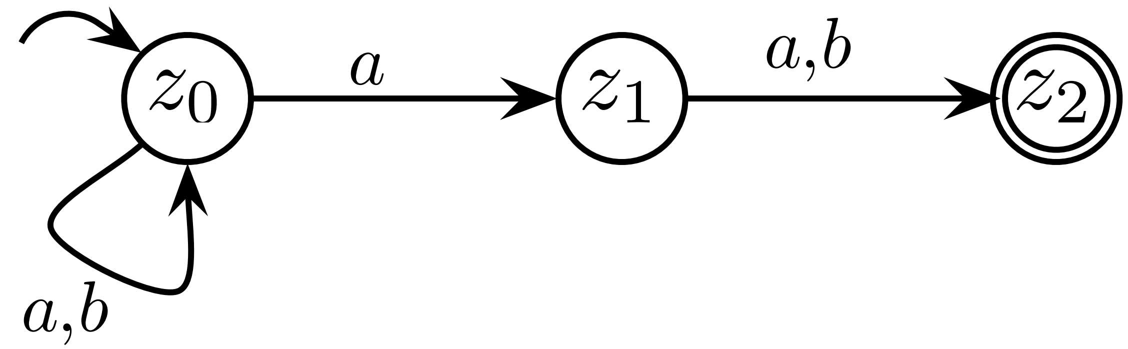
Konstruktion des äquivalenten, initial zusammenhängenden *vDFA*s:

$$\{z_0\}$$

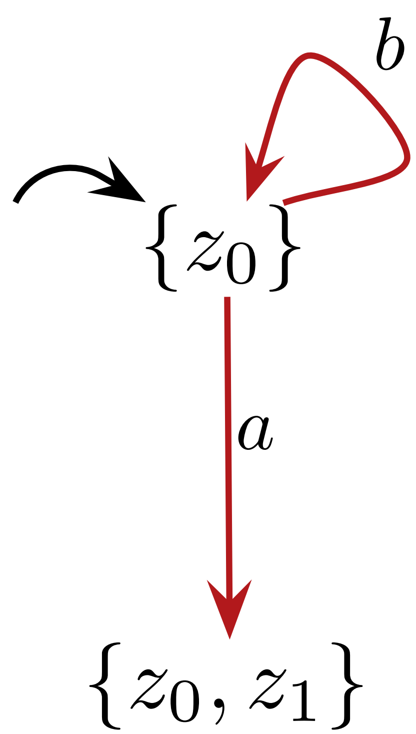
Wir beginnen mit der Menge der Startzustände des NFA, also dem neuen Startzustand des zu konstruierenden vDFA!

Beispiel 2

Gegeben sei dieser *NFA*:

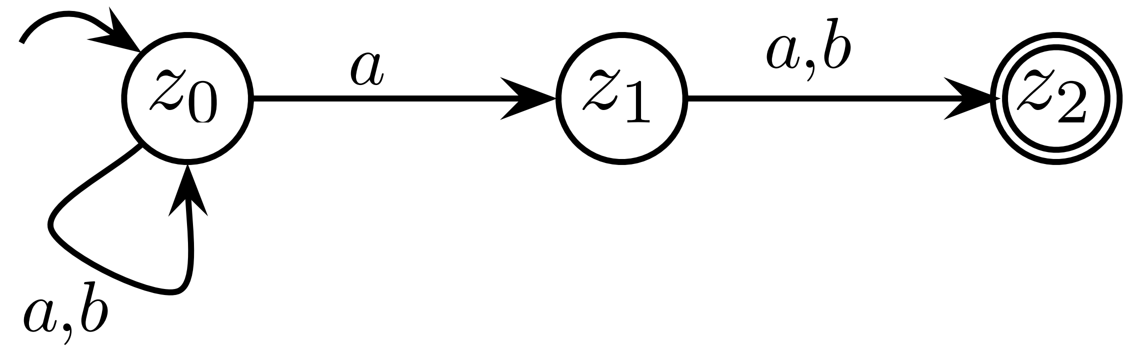


Konstruktion des äquivalenten, initial zusammenhängenden *vDFA*s:

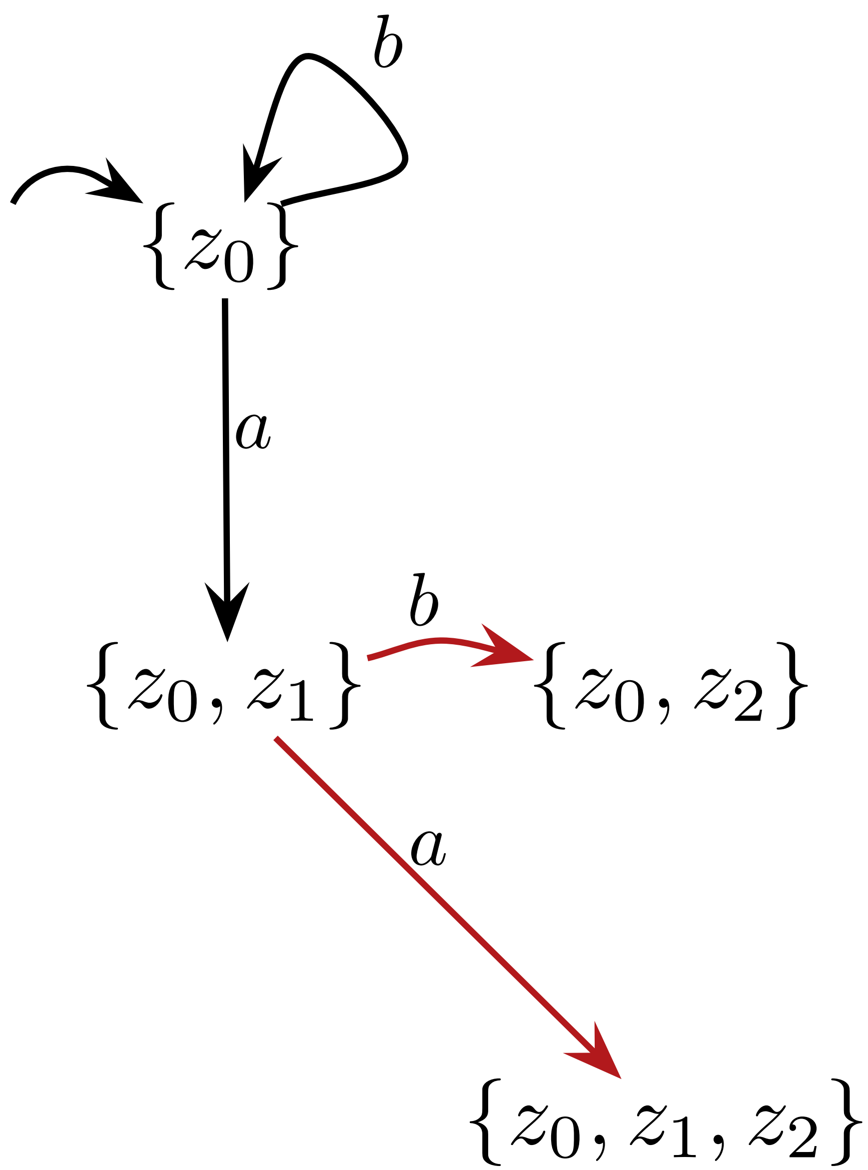


Beispiel 2

Gegeben sei dieser *NFA*:

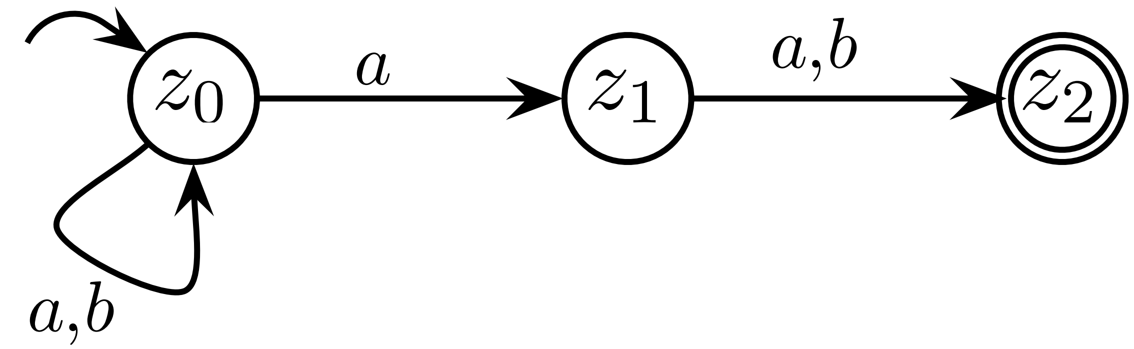


Konstruktion des äquivalenten, initial zusammenhängenden *vDFA*s:

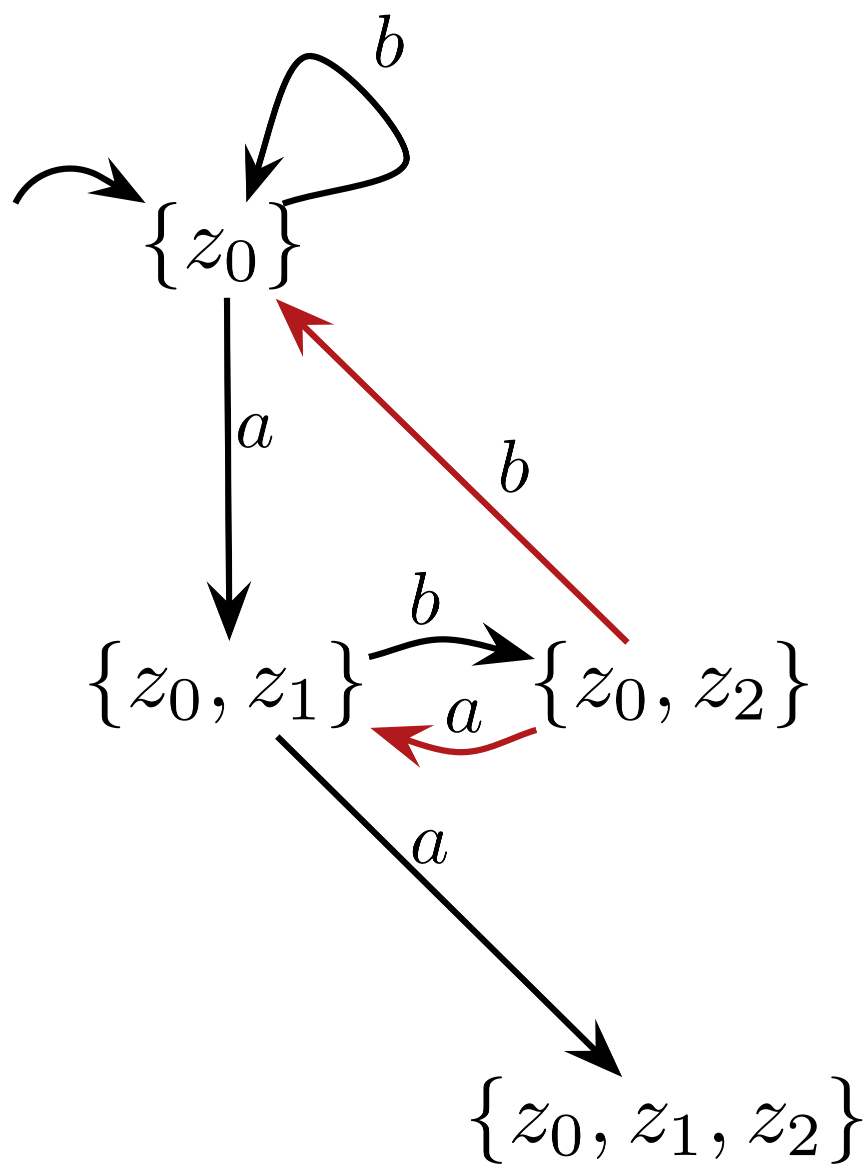


Beispiel 2

Gegeben sei dieser *NFA*:

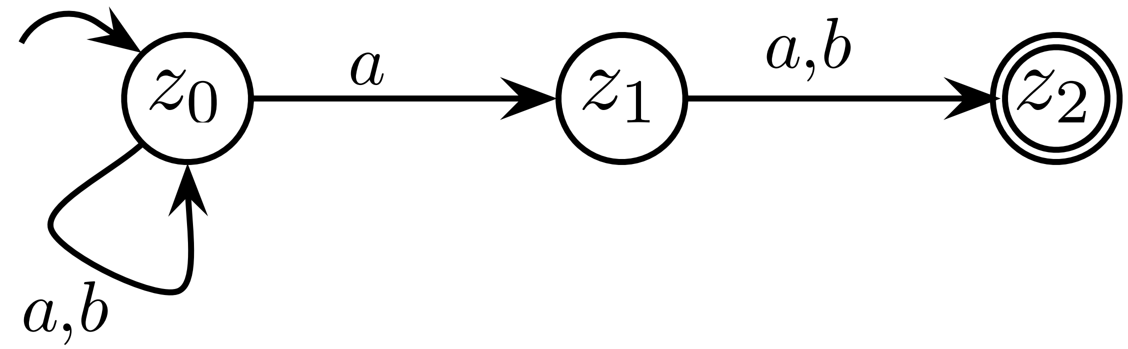


Konstruktion des äquivalenten, initial zusammenhängenden *vDFA*s:

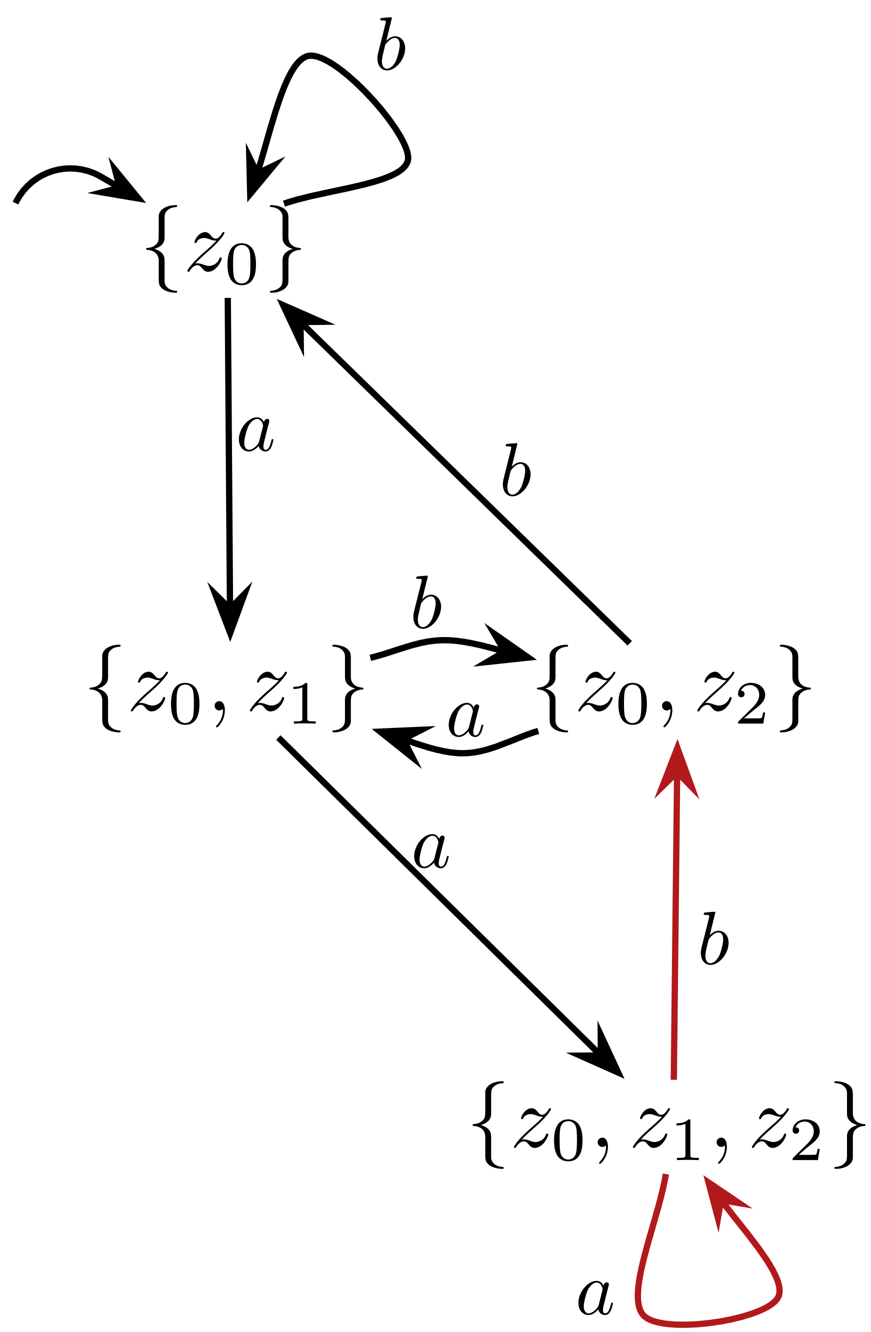


Beispiel 2

Gegeben sei dieser *NFA*:

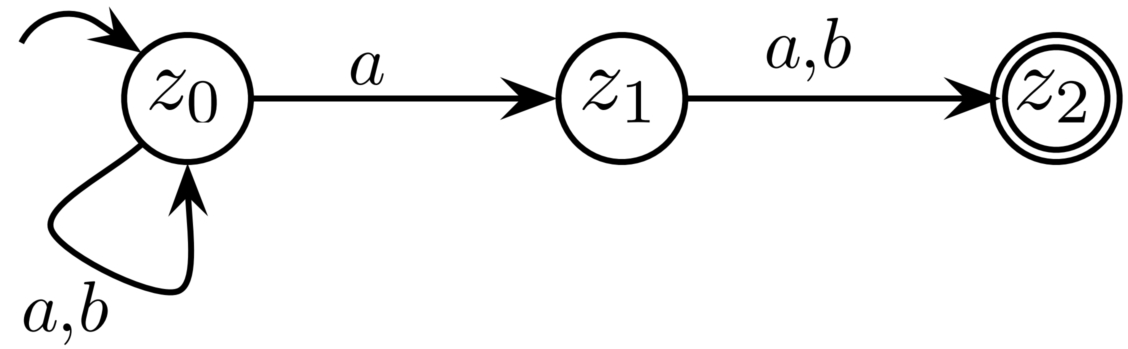


Konstruktion des äquivalenten, initial zusammenhängenden *vDFA*s:

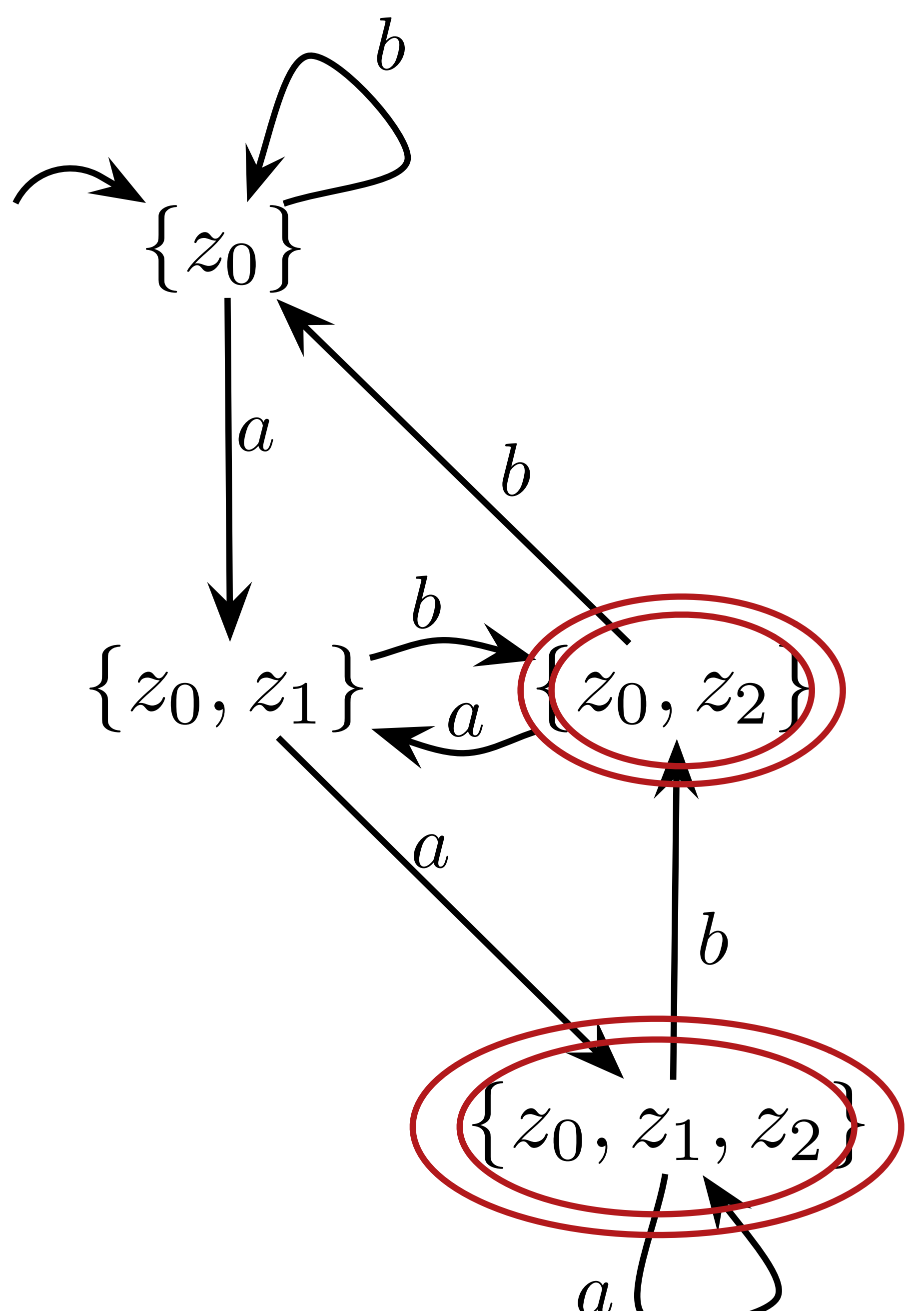


Beispiel 2

Gegeben sei dieser *NFA*:



Konstruktion des äquivalenten, initial zusammenhängenden *vDFA*s:

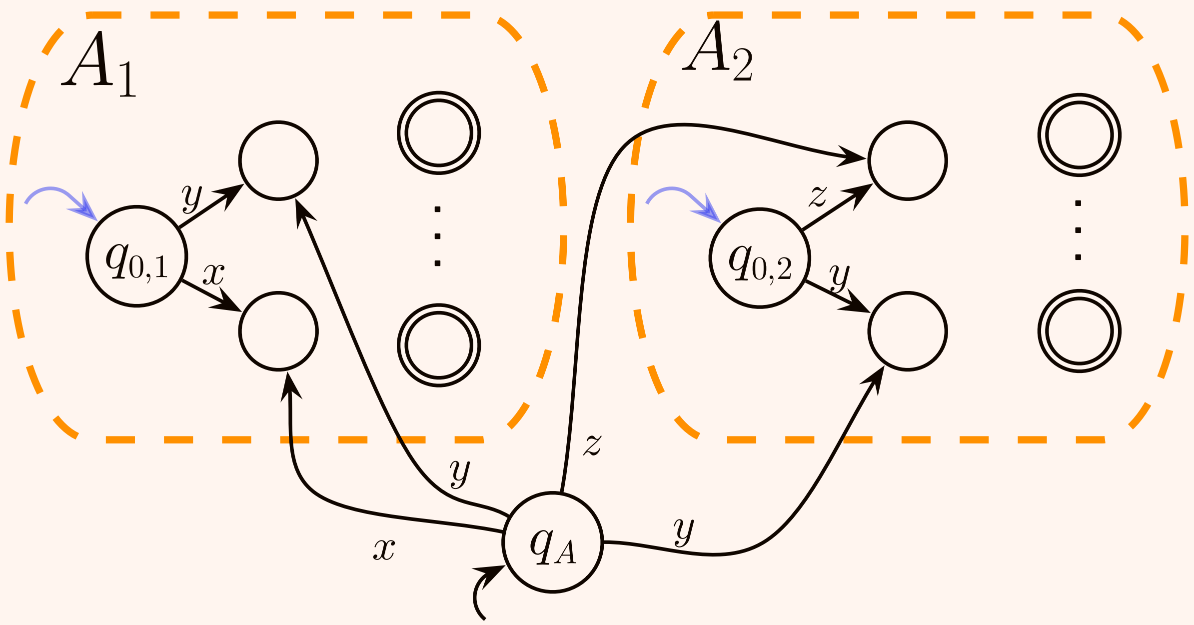


Schlussfolgerungen und Anwendungen

- *NFAs* akzeptieren dieselbe Sprachfamilie *REG*, wie *DFAs*.

- Sind L_1 und L_2 reguläre Sprachen, dann ist auch $L_1 \cup L_2$ regulär.

- graphisch: A_1 und A_2 sind *DFAs*



- formal: $A = (Q_1 \uplus Q_2 \uplus \{q_A\}, \Sigma_1 \cup \Sigma_2, \delta_A, q_A, F_1 \cup F_2)$.

$$\text{Mit } \delta_A(p,x) := \begin{cases} \{\delta_1(p,x)\}, & \text{falls } p \in Q_1 \\ \{\delta_2(p,x)\}, & \text{falls } p \in Q_2 \end{cases}$$

$$\text{und } \delta_A(q_A, x) := (\{\delta_1(q_{0,1}, x)\} \cup \{\delta_2(q_{0,2}, x)\})$$

Weitere Abschlusseigenschaften sind bekannt:

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

Viele Eigenschaften sind einfacher zu beweisen, wenn allgemeinere Varianten des Endlichen Automaten bekannt sind! Z.B. solche, die Übergänge machen können, ohne etwas von der Eingabe zu lesen!

DFA exponentiell größer als äquivalenter NFA

Nach der gegebenen Konstruktion des Potenzautomaten hat der äquivalente DFA exponentiell mehr Zustände als der zu Beginn bekannte NFA.

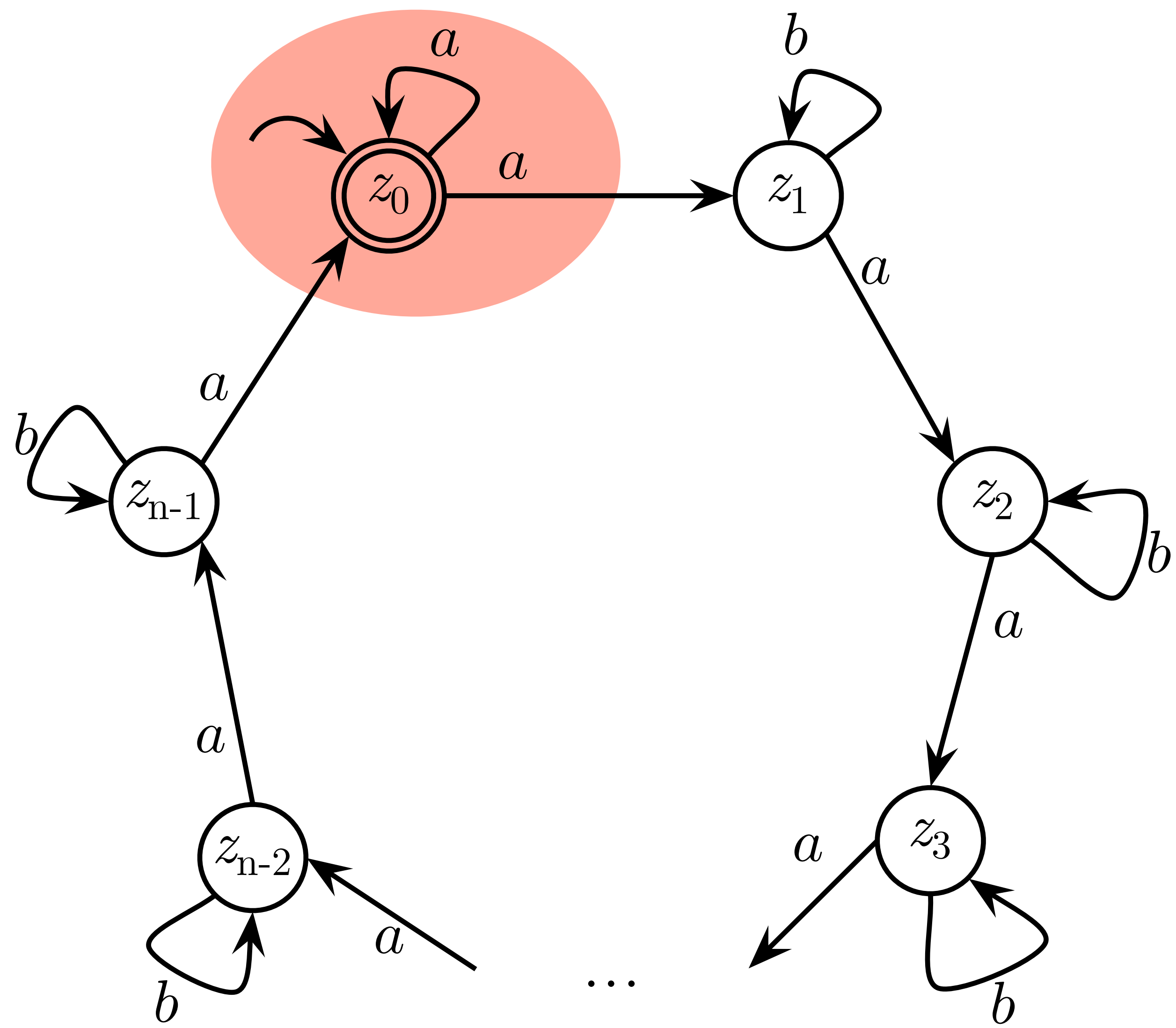
Dies ist keine Schwäche dieser Konstruktion!

In einigen Fällen sind tatsächlich so viele Zustände für den DFA nötig:

Satz 13.2:

Zu jeder natürlichen Zahl $n \geq 2$ gibt es einen buchstabierenden, ϵ-freien NFA A_n mit genau n Zuständen, so dass *jeder* äquivalente vollständige DFA mindestens 2^n Zustände besitzt.

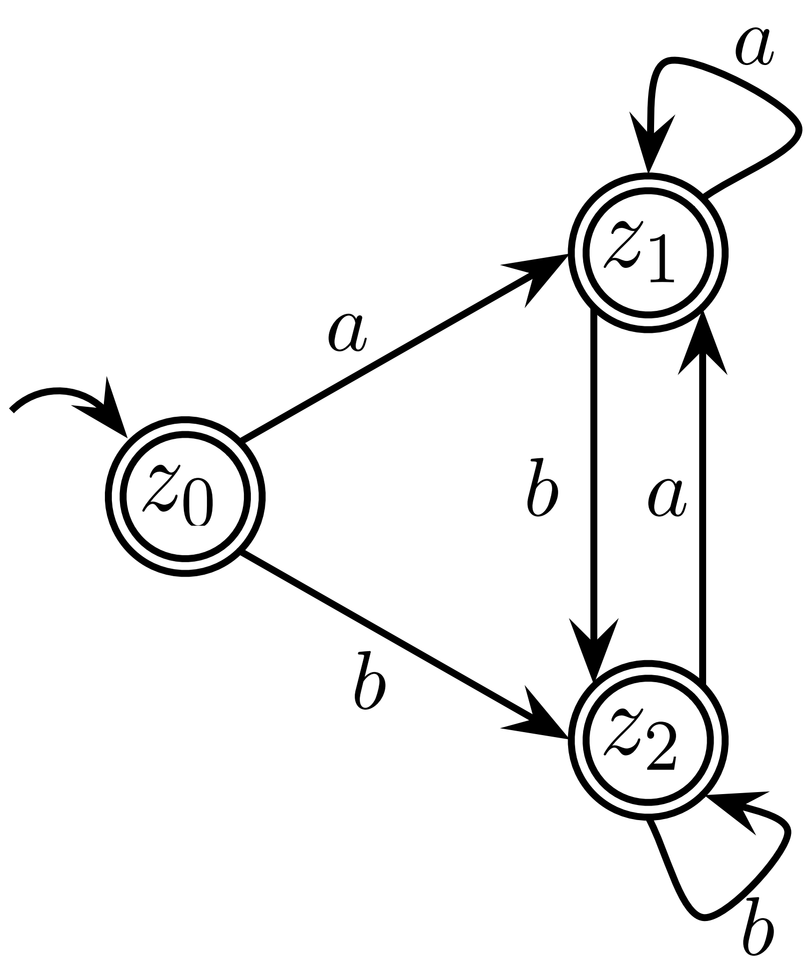
Beispiel (ohne Beweis)

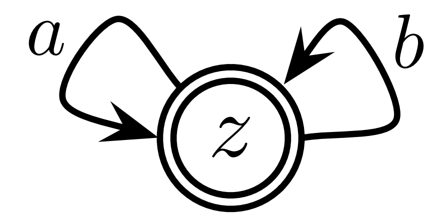


Welche Menge beschreibt solch ein NFA A_n? An den Schwierigkeiten, diese Menge mit Worten exakt zu beschreiben, merken wir, dass eine andere Methode für die Beschreibung regulärer Mengen praktisch sein kann.

minimale Automaten

Betrachtet man die beiden in Abbildung abgebildeten vDFA's, so stellt sich schnell heraus, dass die beiden endlichen Automaten beide die gleiche Sprache akzeptieren, nämlich $\{a, b\}^*$.





Es gibt für jede Sprache immer einen vDFA mit der kleinstmöglichen Anzahl von Zuständen. Alle diese äquivalenten vDFA's sind bis auf die Bezeichnung der Zustände isomorph und „dieser" vDFA wird als **minimaler vDFA** einer Sprache bezeichnet.

Es gibt Verfahren, diesen effektiv zu konstruieren, indem man äquivalente Zustände bestimmt. Wir benutzen Äquivalenzklassen der sog. Nerode-Äquivalenz, einer rechtsinvarianten Äquivalenzrelation.

Nerode-Äquivalenz

Definition 13.7:

Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA, dann ist die automatenspezifische Äquivalenzrelation $R_A \subseteq \Sigma^* \times \Sigma^*$ erklärt durch:

$$u\, R_A\, v \text{ genau dann, wenn } \delta^*(z_0, u) = \delta^*(z_0, v)$$

Sei $L \subseteq \Sigma^*$ eine reguläre Menge, dann ist die *Nerode-Äquivalenz* oder *syntaktische Rechtskongruenz* $R_L \subseteq \Sigma^* \times \Sigma^*$ erklärt durch:

$$u\, R_L\, v \quad \text{genau dann, wenn} \quad \forall w \in \Sigma^* : (uw \in L \iff vw \in L).$$

Ein wichtiges Ergebnis von Myhill und Nerode, 1957/1958, verbindet diese Definitionen mit Automaten und regulären Mengen:

Nerode-Äquivalenz

Satz 13.3:

Die folgenden Aussagen sind äquivalent:

1. $L \subseteq \Sigma^*$ ist regulär.
2. L ist Vereinigung von Äquivalenzklassen einer rechtsinvarianten Äquivalenzrelation mit endlichem Index.
3. Die Nerode-Äquivalenz R_L hat endlichen Index.

Dieser Satz kann verwendet werden, um zu zeigen, dass gewisse Sprachen nicht regulär sein können. Die häufig verwendete Menge $\{a^n b^n \mid n \in \mathbb{N}\}$ lässt sich so als nicht reguläre Menge beweisen!
Wir werden später ein anderes Verfahren kennenlernen, mit dem dieses gleichfalls möglich ist!

Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten.
Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.

i=1	$S_{1,1} := F$	$S_{1,2} := (Q \setminus F)$

Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten.
Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.

$i=1$	$S_{1,1} := F$	$S_{1,2} := (Q \setminus F)$
	....	....

Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten.
Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.

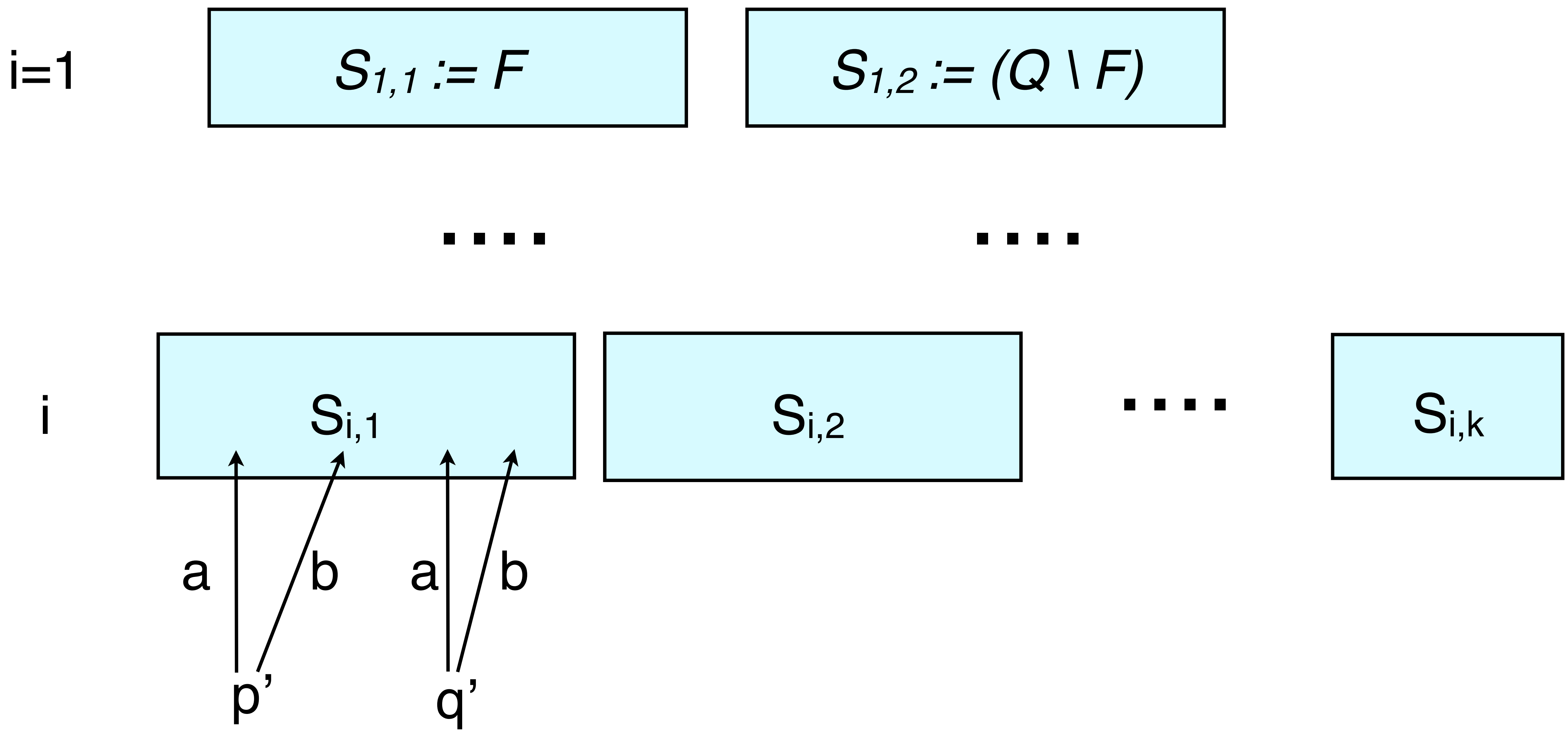
i=1 $\quad \boxed{S_{1,1} := F} \quad \boxed{S_{1,2} := (Q \setminus F)}$

....

i $\quad \boxed{S_{i,1}} \quad \boxed{S_{i,2}} \quad .... \quad \boxed{S_{i,k}}$

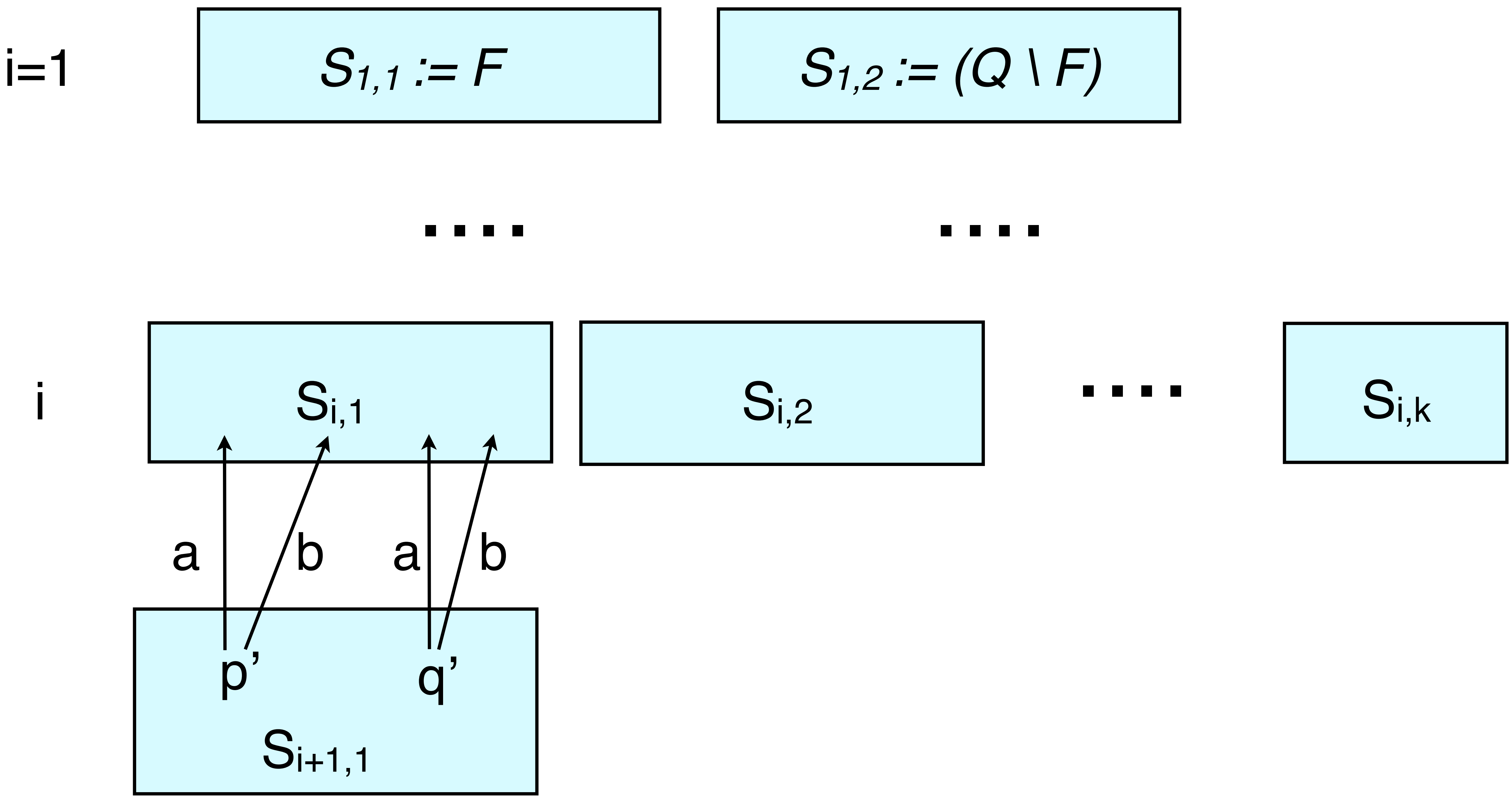
Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten.
Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.



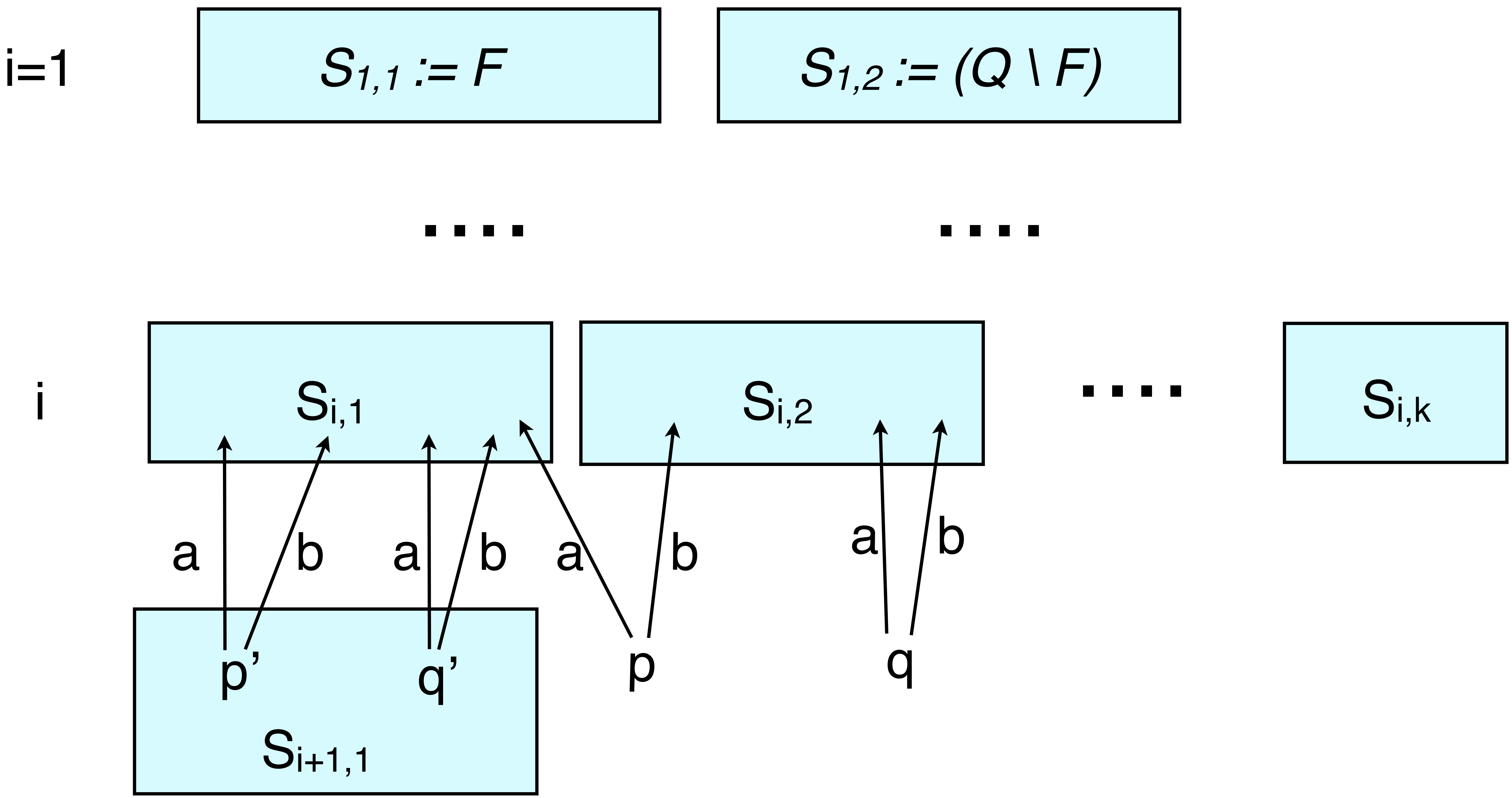
Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten. Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.



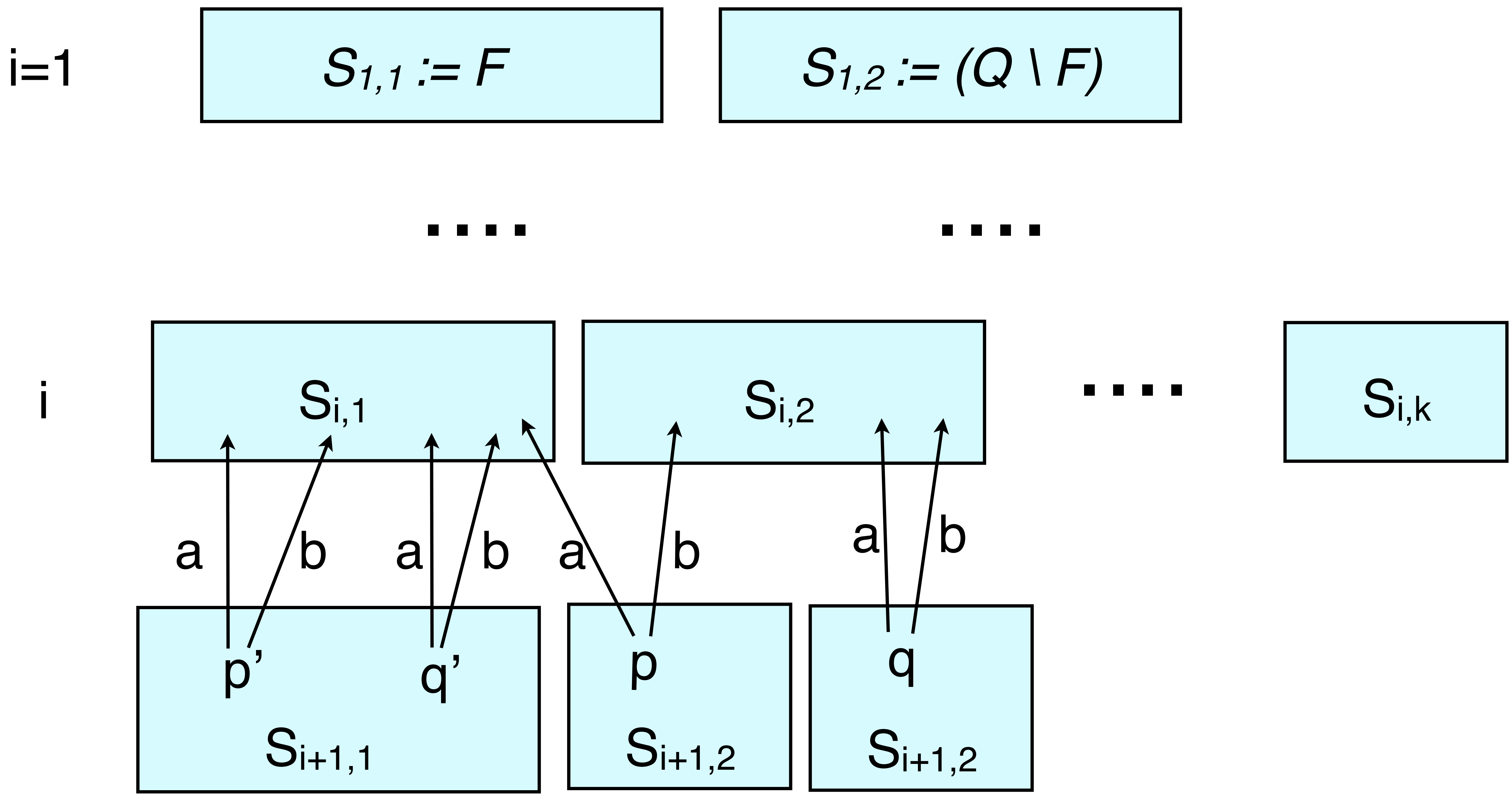
Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten.
Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.



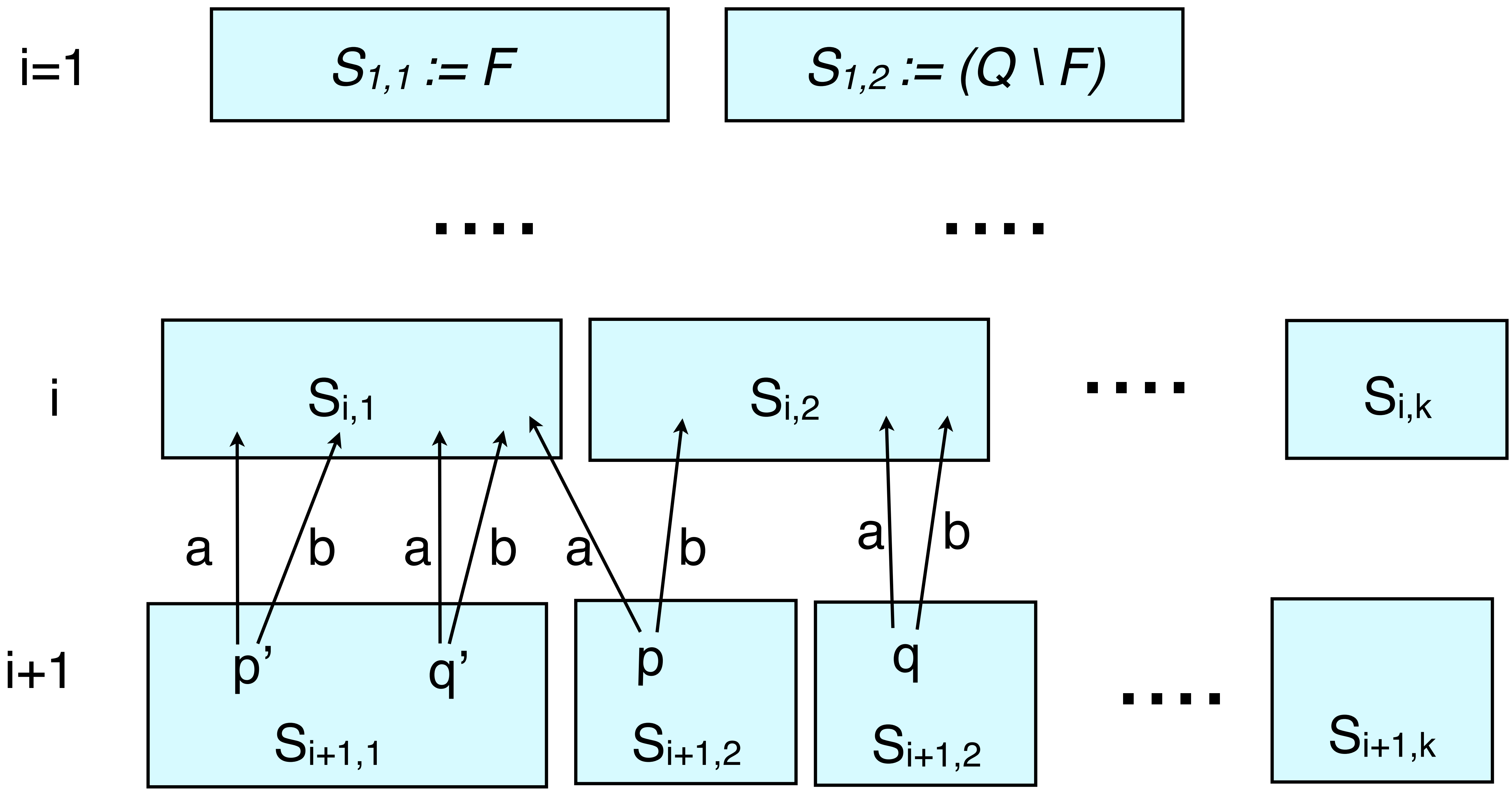
Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten.
Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.



Äquivalenz: Partition der Zustände

Wir wollen berechnen, welche Zustände das Gleiche leisten. Dazu suchen wir nach Eingaben, für die sich Zustände bzgl. der Akzeptanz unterscheiden.



Konstruktion des minimalen Automaten

Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein beliebiger *DFA*. Von Q werden wiederholt feinere, disjunkte Zerlegungen, sog. Partitionen, erzeugt:

1. die erste Partition von Q ist $\Pi_1 := \{S_{1,1}, S_{1,2}\}$, wobei $S_{1,1} := F$ und $S_{1,2} := Q \setminus F$ sei.
2. Sei $\Pi_i = \{S_{i,1}, S_{i,2}, \ldots, S_{i,k_i}\}, k_i \geq 0$. Bilde Partition Π_{i+1} gemäß folgender Bedingung:
   Zwei Zustände $p, q \in Q$ gehören genau dann zum selben Block, wenn für jedes $a \in \Sigma$ gilt: $\delta(q, a)$ und $\delta(p, a)$ liegen im gleichen Block in Π_i.
3. Falls für ein $i \in I\!N$ irgendwann $\Pi_{i+1} = \Pi_i$ gilt, dann ist Π_i die Zustandsmenge des minimalen Automaten, andernfalls wird die Konstruktion mit Schritt 2 fortgesetzt.

Jeder Zustand des minimalen Automaten ist also ein Element $S_{i,j}$ der Menge 2^Q mit $S_{i,j} \xrightarrow{x} S_{i,k} \Leftrightarrow \exists p \in S_{i,j} : \delta(p, x) \in S_{i,k}$

FGI 1

Automaten, Formale Sprachen, Berechenbarkeit

Kap. 14

(Konfiguration eines NFA, ε-NFA,

Pumping-Lemma, reguläre Ausdrücke)

Michael Köhler Bußmeier

Konfiguration eines NFA

Zur Beschreibung der aktuellen Situation, in der sich ein NFA befindet, reicht der eingenommene Zustand alleine nicht aus: Wichtig ist die noch zu verarbeitende Eingabe!

Definition 14.1:

Für einen *NFA* A heißt $k \in Q \times \Sigma^*$ **Konfiguration** gdw.

- $k = (p, w)$ mit $w \in \Sigma^*$ und $p \in Q$:
- A befindet sich im Zustand p und die noch zu verarbeitende Eingabe ist w.
- $w = \epsilon$ bedeutet, dass die Eingabe vollständig gelesen wurde.

Übergänge zwischen Konfigurationen:

Folgekonfigurationen

Definition 14.2:

Sei $A := (Q, \Sigma, \delta, S_0, F)$ ein *NFA*, $p, q \in Q$, $a \in \Sigma$ und $w \in \Sigma^*$.
Die Konfiguration (q, w) ist genau dann eine Folgekonfiguration von (p, aw), wenn $q \in \delta^*(p, a)$. Wir notieren dies durch:

$$(p, aw) \vdash (q, w)$$

$\vdash^*$ bezeichnet die reflexive, transitive Hülle von $\vdash$, d.h.:
$k \vdash^* k$ für alle Konfigurationen k und

$$k_1 \vdash^* k_3 \iff \exists k_2 : k_1 \vdash^* k_2 \wedge k_2 \vdash k_3 \quad \vee k_1 = k_3$$

Damit könnte die akzeptierte Sprache auch so definiert werden:

Definition 14.3:

Sei $A := (Q, \Sigma, \delta, S_0, F)$ ein *NFA*, dann ist

$$L(A) = \{w \in \Sigma^* \mid \exists p \in S_0 : \exists q \in F : (p, w) \vdash^* (q, \epsilon)\}$$

Definition 14.4:

Ein **endlicher Automat mit ϵ-Übergängen** (**ϵ-FA**), ist ein *NFA* $A := (Q, \Sigma, \delta, S_0, F)$, mit:

$$\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q.$$

[Die Darstellung von δ als Relation ist oft praktischer als die über eine Funktion

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q]$$

Die Relation $\vdash$ wird nun an die ϵ-Übergänge angepasst:

Definition 14.5:

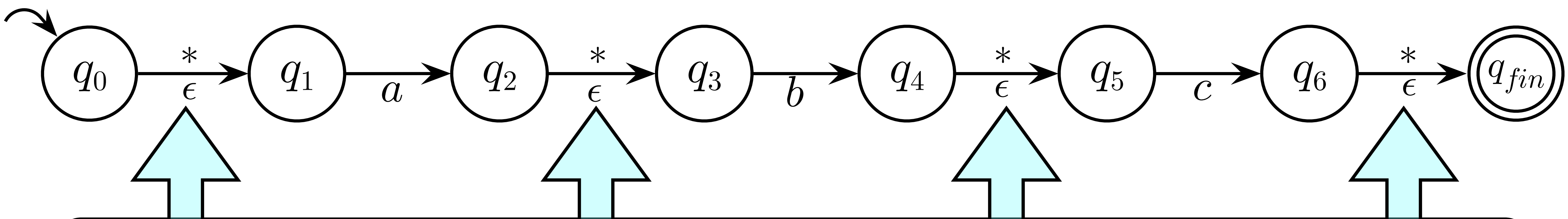
$$\forall x \in \Sigma \cup \{\epsilon\} \forall w \in \Sigma^* : (p, xw) \vdash (q, w) \iff (p, x, q) \in \delta.$$

$$L(A) := \{w \in \Sigma^* \mid (s_0, w) \vdash^* (s, \epsilon), s_0 \in S_0, s \in F\}.$$

ε-FA äquivalent zum NFA

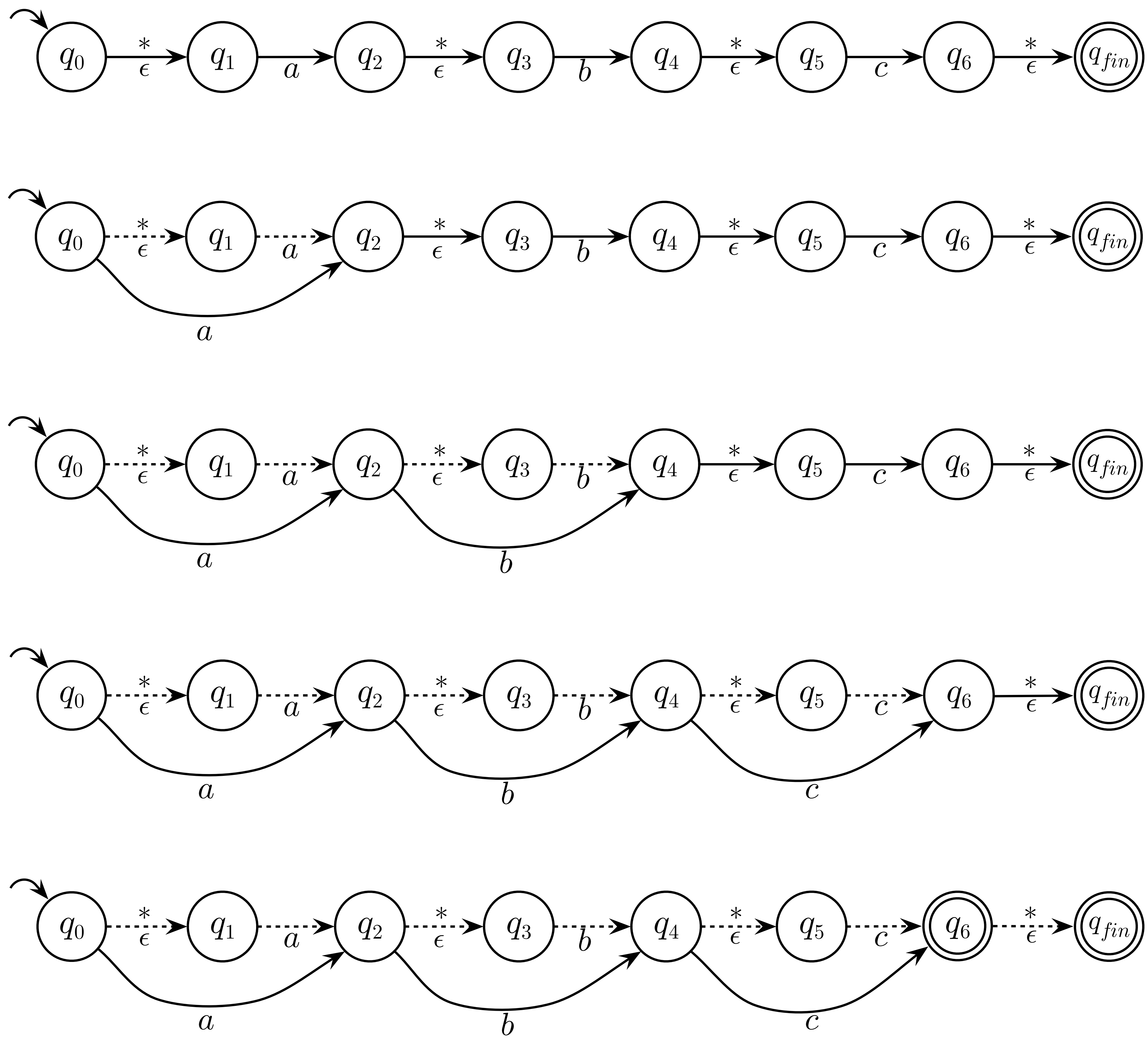
Da jeder NFA offensichtlich ein ε-FA ist, fehlt nur noch ein Beweis für die Umkehrung!

Die Idee dazu ist, Pfade im ε-FA durch geschicktes Zusammenziehen der ε-Kanten zu verkürzen, so dass ein NFA entsteht:



Das Zustandsdiagramm eines NFA ist immer endlich, also kann prinzipiell festgestellt werden, welche Zustände nur mit ε-Kanten verbunden sind.
Dieser konstruktive Aspekt ist jedoch für einen Existenzbeweis von untergeordneter Bedeutung.

ε-FA äquivalent zum NFA



mathematische Beschreibung

Satz 14.1:

Zu jedem ϵ-*FA* gibt es einen äquivalenten *NFA* ohne ϵ-Übergänge.

Beweis:

Sei $A := (Q, \Sigma, \delta, S_0, F)$ ein *NFA*, der ϵ-Kanten besitzen kann. Ein äquivalenter *NFA* $B := (Q, \Sigma, \delta', S_0, F')$ ohne ϵ-Kanten wird wie folgt erklärt:

$$\begin{aligned}\delta' \quad := \quad & \{(z, x, z'') \in Q \times \Sigma \times Q \mid \exists (z', x, z'') \in \delta : \\ & (z, \epsilon) \vdash^* (z', \epsilon) \text{ (in A)}\}\end{aligned}$$

und $F' := \{p \in Q \mid \exists q \in F : (p, \epsilon) \vdash^* (q, \epsilon)\}$.

mathematische Beschreibung

Satz 14.1:

Zu jedem ϵ-*FA* gibt es einen äquivalenten *NFA* ohne ϵ-Übergänge.

Beweis:

Sei $A := (Q, \Sigma, \delta, S_0, F)$ ein *NFA*, der ϵ-Kanten besitzen kann. Ein äquivalenter *NFA* $B := (Q, \Sigma, \delta', S_0, F')$ ohne ϵ-Kanten wird wie folgt erklärt:

$$\begin{aligned} \delta' \quad := \quad & \{(z, x, z'') \in Q \times \Sigma \times Q \mid \exists (z', x, z'') \in \delta : \\ & (z, \epsilon) \vdash^* (z', \epsilon) \text{ (in A)}\} \end{aligned}$$

und $F' := \{p \in Q \mid \exists q \in F : (p, \epsilon) \vdash^* (q, \epsilon)\}$.

Eine mathematisch elegante Konstruktion der Menge

$$\{(p, q) \mid (p, \epsilon) \vdash^* (q, \epsilon)\} \subseteq Q \times Q$$

wird vielleicht in den Übungen bearbeitet und effizientere Methoden in der Vorlesung AD ausführlicher besprochen werden!

einige Abschlusseigenschaften von $\mathcal{R}eg$

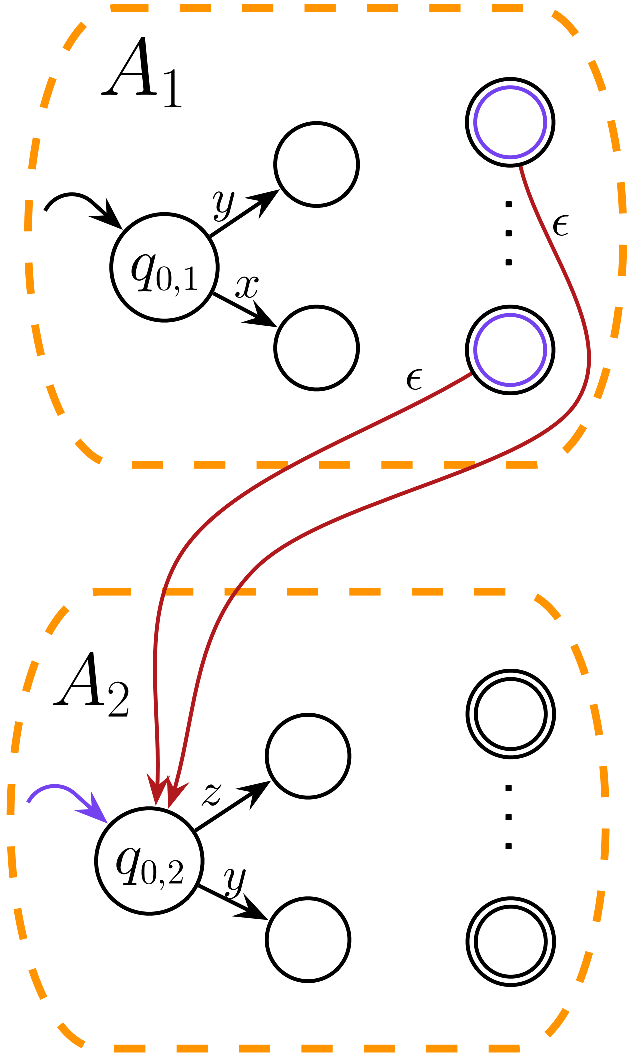
Satz 14.2:

Die Familie $\mathcal{R}eg$ der regulären Sprachen ist abgeschlossen gegenüber folgenden Operationen:

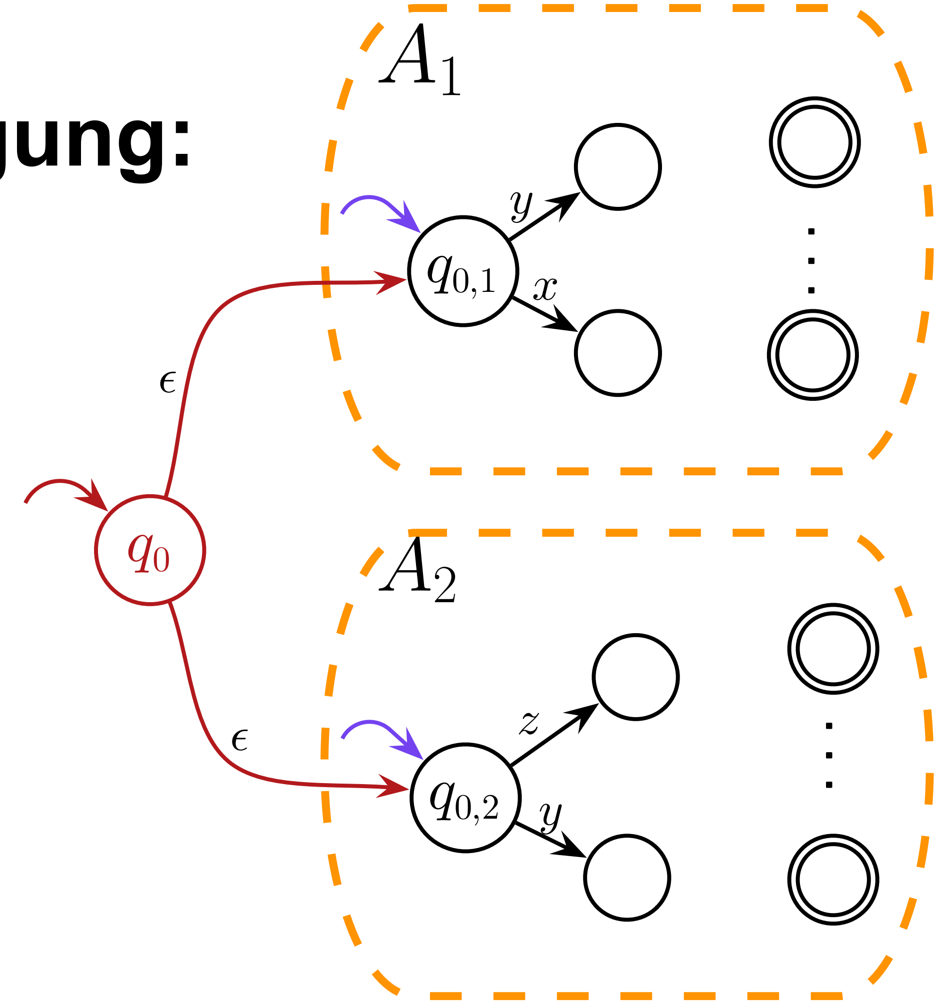
- Vereinigung
- Produkt
- Sternbildung
- Durchschnittsbildung
- Komplement
- Homomorphismen

Beweis zu Satz 14.2: (mit ε-Kanten)

Produkt:



Vereinigung:

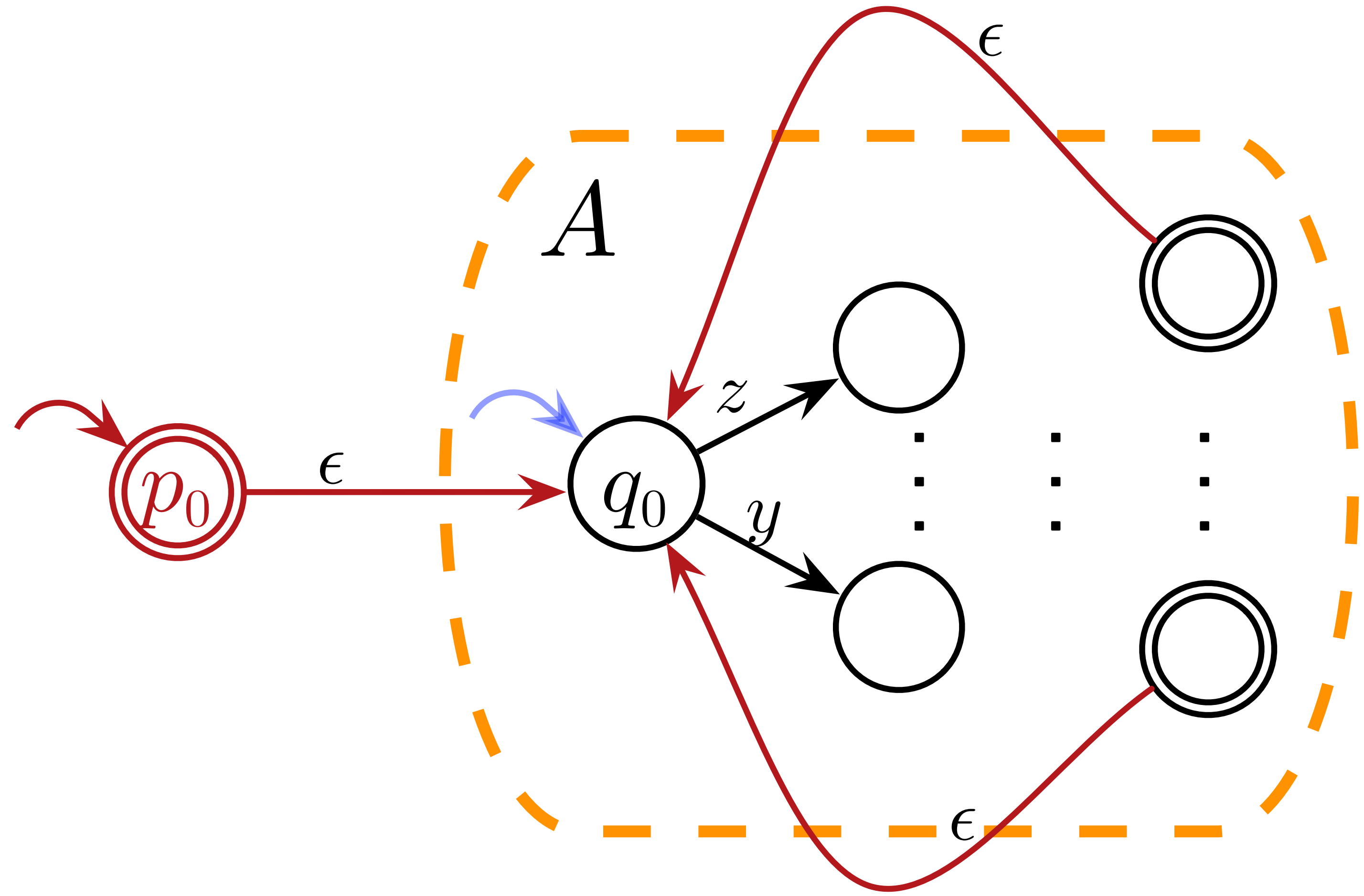


Komplementbildung: Vertauschen von End- mit nicht-Endzuständen im *DFA* ergibt einen *DFA*, der das Komplement akzeptiert.

Durchschnitt: Folgt aus Komplementbildung und Vereinigung.

Homomorphismen: ... wird später gezeigt ...

Sternbildung



Aus einem *DFA*, der die Sprache L akzeptiert,
wird mit obiger Konstruktion
(das Rote ist neu, das Lila wird entfernt)
ein ε-*FA*, der die Sprache L^* akzeptiert!

Grenzen von $\mathcal{R}eg$

- Möchte man zeigen, dass eine Sprache L nicht regulär ist, muss bewiesen werden, dass **kein** *DFA* diese Sprache akzeptiert!

- Wichtiges Hilfsmittel ist das sogenannte Pumping-Lemma oder „uvw-Theorem“.

- Es trifft eine Aussage über den Aufbau langer Wörter aus einer regulären Sprache.

- Kann man zeigen, dass gemäß Pumping-Lemma Wörter in der Sprache enthalten sein müssten, sie es aber nicht sind, so wissen wir, dass die Sprache nicht regulär sein kann!

FGI 1

Automaten, Formale Sprachen, Berechenbarkeit

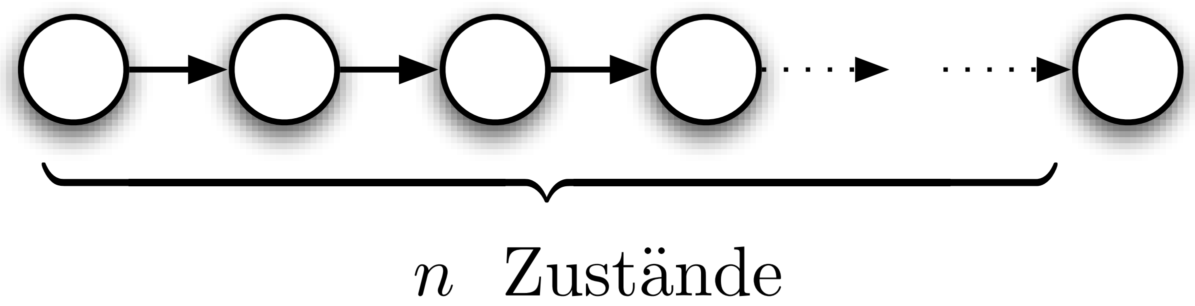
Kap. 14

Pumping-Lemma

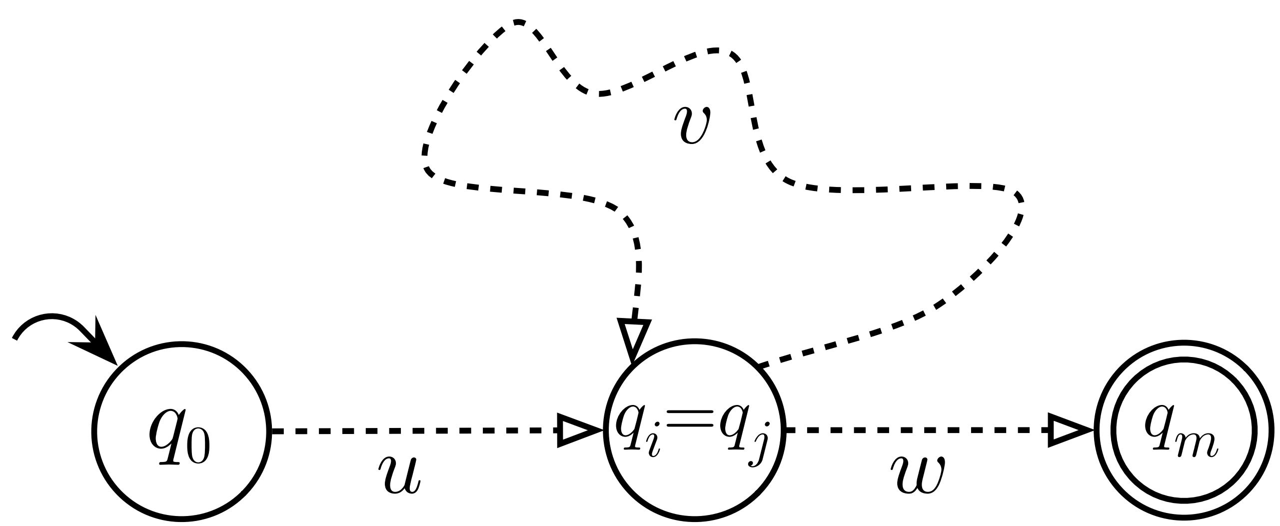
Michael Köhler Bußmeier

Motivation: Pumping-Lemma

im DFA:



Ein akzeptiertes Wort mit Länge n muss auf seiner Erfolgsrechnung mindestens n+1 Zustände besuchen. Dazu muss immer eine Schleife durchlaufen werden!



Dann wird aber auch jedes Wort der Form uv^iw akzeptiert.

Pumping-Lemma (uvw-Theorem)

Satz 14.3:

Zu jeder regulären Menge $R \in \mathcal{R}eg$ gibt es eine Zahl $n \in \mathbb{N}$, so dass jedes Wort $z \in R$ mit $|z| \geq n$ zerlegt werden kann in die Form $z = uvw$ mit:

(i) $|uv| \leq n$

(ii) $|v| \geq 1$ und

(iii) $\{u\}\{v\}^*\{w\} \subseteq R$

Was könnte n für L = {ab, bba, abba} $\in \mathcal{R}eg$ sein?

Beweis

- Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA, der R akzeptiert, d.h.: $L(A) = R$.

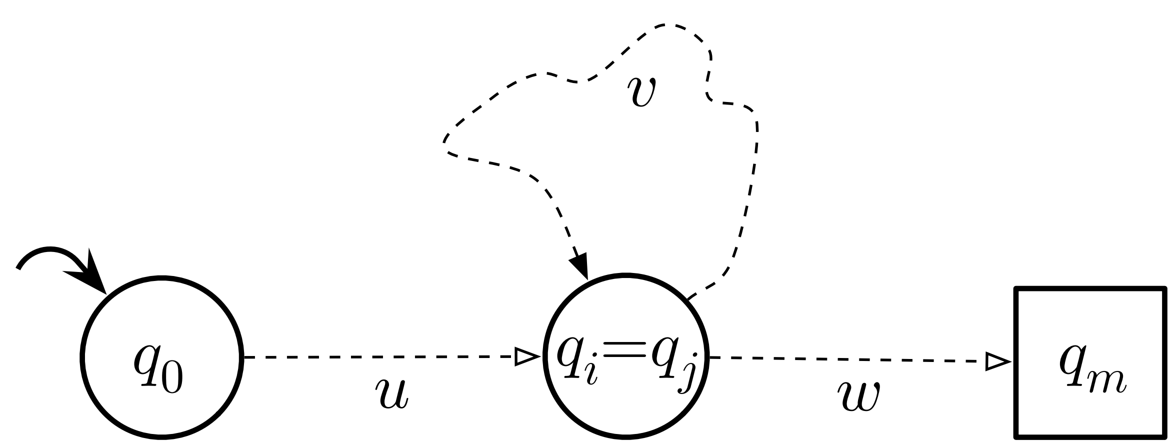
- Wähle $n := |Q|$. Sei nun $z = x_1 x_2 \ldots x_m \in R$, mit $m \geq n$ und $x_i \in \Sigma$, ein beliebiges Wort der Sprache.

- Die $m+1$ Zustände
  $q_0,\ q_1 := \delta(q_0, x_1),\ q_2 := \delta(q_1, x_2),\ \ldots,\ q_m := \delta(q_{m-1}, x_m)$
  sind von q_0 aus erreichbar und werden auf diesem Erfolgspfad für die Akzeptierung von z besucht.
  Kurz: $\forall 1 \leq i \leq m : q_i = \delta^*(q_0, x_1 x_2 \ldots x_i)$.

- Da maximal $n \leq m$ viele davon verschieden sein können, muss nach dem **Schubfachprinzip** spätestens mit dem $(n+1)$-ten, ein Zustand in dieser Folge zweimal vorgekommen sein.

Beweis: Pumping-Lemma (2)

- Für $0 \leq i < j \leq n$ gelte nun die Gleichheit $q_i = q_j$.
- Dann finden wir mit
  $u := \boxed{x_1 x_2 \ldots x_i}$, $v := \boxed{x_{i+1} x_{i+2} \ldots x_j}$ und
  $w := \boxed{x_{j+1} x_{j+2} \ldots x_m}$ gerade die verlangte Zerlegung von
  z, die (i) und (ii) erfüllt.
- Dass auch (iii) gilt, sieht man leicht ein, denn mit
  $q_i := \delta(q_0, u)$ gilt doch
  $q_i = \delta(q_i, \epsilon) = \delta(q_i, v) = \delta(q_i, vv) = \ldots$.
  Also werden alle Wörter der Form $u \cdot v^k \cdot w$, für jedes $k \geq 0$,
  von A ebenfalls akzeptiert.



Anwendungsbeispiel

- Wir wollen mit Hilfe des uvw-Theorems zeigen, dass die Sprache $DUP := \{a^n b^n \mid n \in \mathbb{N}\}$ nicht regulär ist:
- Wäre DUP regulär, so gäbe es eine Zahl $k \in \mathbb{N}$, für die die Folgerung des uvw-Theorems zutrifft.
- $z := a^k b^k \in DUP$ ist ein Wort mit $|z| \geq k$.
- Jede Zerlegung $z = uvw$ mit $|uv| \leq k$ bedeutet $v = a^m$ für ein $m \in \mathbb{N}$ mit $1 \leq m \leq k$.
- Damit müsste auch das Wort $a^{k-m} b^k$ ein Element der Sprache DUP sein, was nicht stimmt.
- Damit ist $DUP \notin \mathcal{R}eg$ nachgewiesen.

hilfreiche Abschlusseigenschaften

- Eine oftmals sehr einfache Methode, von Mengen zu zeigen, dass sie nicht regulär sind, ist die Verwendung der Abschlusseigenschaften von $\mathcal{R}eg$.

  - Zunächst nimmt man an, die fragliche Menge sei regulär.
  - Dann verwendet man Transformationen mit Abschluss-Operatoren, um solche Sprachen zu erhalten, von denen man schon weiß, oder für die man vielleicht leichter zeigen kann, dass diese nicht regulär sind.
  - Somit ist ein Widerspruch erzielt und die ursprüngliche Menge kann nicht regulär gewesen sein.

- Varianten dieses Vorgehens ...

Beispiel

- Wir zeigen, dass die Menge
  $C := \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ nicht regulär ist:
  - Setze $D := C \cap \{a\}^*\{b\}^*$. Mit $C \in \mathcal{R}eg$ wäre auch $D \in \mathcal{R}eg$, es ist aber $D = DUP = \{a^n b^n \mid n \in \mathbb{N}\}$ bekanntlich nicht regulär.

- Das uvw-Theorem ist nicht direkt benutzbar, um zu beweisen, dass die Sprache

  $$L := \{c^k a^l b^m \mid k, l, m \in \mathbb{N} : k = 0 \text{ oder } l = m\}$$

  nicht regulär ist.
  - Wir definieren nun die Menge
    $E := L \cap \{c\}\{a\}^*\{b\}^* = \{ca^n b^n \mid n \in \mathbb{N}\}$, die regulär wäre, wenn L dies ist.
    (Dass E nicht regulär ist zeigt man genauso leicht, wie die Nichtregularität von DUP.)

Beispiel

- Wir zeigen, dass die Menge
  $C := \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ nicht regulär ist:
  - Setze $D := C \cap \{a\}^*\{b\}^*$. Mit $C \in \mathcal{R}eg$ wäre auch $D \in \mathcal{R}eg$, es ist aber $D = DUP = \{a^n b^n \mid n \in \mathbb{N}\}$ bekanntlich nicht regulär.

- Das uvw-Theorem ist nicht direkt benutzbar, um zu beweisen, dass die Sprache

$$L := \{c^k a^l b^m \mid k,l,m \in \mathbb{N} : k = 0 \text{ oder } l = m\}$$

nicht regulär ist. warum nicht?

  - Wir definieren nun die Menge
    $E := L \cap \{c\}\{a\}^*\{b\}^* = \{ca^n b^n \mid n \in \mathbb{N}\}$, die regulär wäre, wenn L dies ist.

    (Dass E nicht regulär ist zeigt man genauso leicht, wie die Nichtregularität von DUP.)

FGI 1

Automaten, Formale Sprachen, Berechenbarkeit

Kap. 14
reguläre Ausdrücke

Michael Köhler Bußmeier

reguläre Ausdrücke (vergl. RA-Sprachen)

Definition 14.6:

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

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

2. Für jedes $a \in \Sigma$ ist a ein regulärer Ausdruck, der die Menge $M_a := \{a\}$ beschreibt.

3. Sind A und B reguläre Ausdrücke, welche die Mengen M_A und M_B beschreiben, dann sind **induktiv** folgende regulären 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^+

4. Nur die mit 1., 2. und 3. erzeugten Ausdrücke sind reguläre Ausdrücke.

wo ist das ε geblieben?

Das Symbol ϵ kam in Definition 14.6 **nicht** vor!

Wieso verlieren wir dadurch nichts an Ausdruckskraft?

Die Definition der Sternbildung war ja für beliebige Mengen $M \subseteq \Sigma^*$ erklärt durch

$$M^* := \bigcup_{i=0}^{\infty} M_i, \quad \text{wobei } M_0 := \{\epsilon\} \text{ und}$$

$$M_{i+1} := M_i \cdot M.$$

Damit ist nun

$$\emptyset^* = \{\epsilon\}$$

und das Symbol ϵ, kann weiterhin für das leere Wort reserviert bleiben!

weniger Klammern und die Praxis

Regeln zur Klammerersparnis:

- unäre Operatoren vor binären ($*$ vor $\cdot$ und $+$)
- Punkt vor Strich ($\cdot$ vor $+$)

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

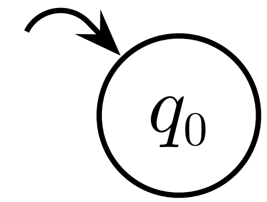
  `egrep   ^Th...i.$   /usr/dict/duden`

  ein, so könnten folgende Wörter gefunden und angezeigt werden:

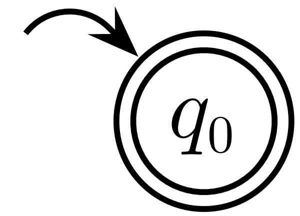
  `Thermik`   `Theorie`   `Thespis`

Einfache Mengen sind regulär:

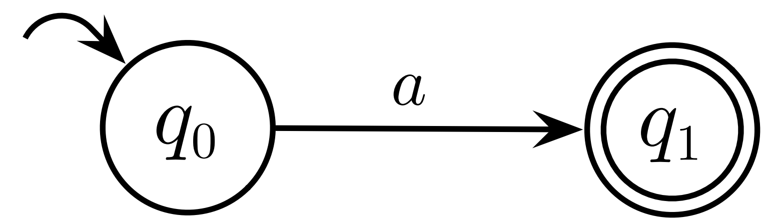
- Die leere Menge ist regulär.



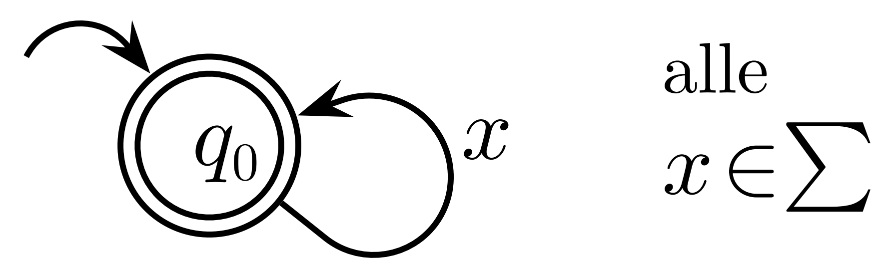
- Die Menge $\emptyset^* = \{\lambda\}$ ist regulär.



- 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 $M \cdot \emptyset = \emptyset \cdot M = \emptyset$.

reguläre Ausdrücke beschreiben reguläre Mengen

Beginnend mit den Automaten für die einfachen Mengen, konstruiert man neue Automaten entsprechend der Verknüpfungen im regulären Ausdruck!

Folgerung: Für jeden regulären Ausdruck A ist die Menge M_A regulär, weil sie von einem NFA akzeptiert wird!

Wir sahen eben, dass jeder regulärer Ausdruck eine reguläre Menge beschreibt.
Aber kann auch jede reguläre Menge durch einen regulären Ausdruck beschrieben werden?

Satz 14.4:

Für jedes Alphabet Σ gilt DFA_Σ = Menge der durch reguläre Ausdrücke über dem Alphabet Σ beschreibbaren Sprachen.

Plan des konstruktiven Beweises

Wir müssen also noch zeigen, dass es zu jedem DFA einen regulären Ausdruck gibt, der die von ihm akzeptierte Sprache beschreibt!

- **Beweisidee:** Wir stellen eine Rekursionsformel auf, die zu jedem DFA eine endliche Mengendarstellung beschreibt, mit der die akzeptierte Sprache durch endlich häufiges Anwenden von $\cup$, $\cdot$ und $*$ dargestellt wird.
  - Konstruktion eines rationalen Ausdruckes für einen beliebigen DFA.
  - Konstruktion funktioniert auch für NFA.

Kleene‘s Konstruktion

- Sei $A = (\{q_1, \ldots, q_n\}, \Sigma, \delta, q_1, F)$ ein DFA
- Sei für $0 \leq k \leq n$ und $1 \leq i,\ j \leq n$:

$$\widetilde{R}^k_{i,j} := \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \delta(q_i, w) = q_j \ \wedge\ \Big[(\forall u, v \in \Sigma^+ : w = uv \wedge \\ \delta^*(q_i, u) = q_r) \Longrightarrow r \leq k \Big] \end{array} \right\}$$

- $\widetilde{R}^k_{i,j}$ enthält alle Wörter, die auf Pfaden von q_i nach q_j über die Menge $\{q_1, \ldots, q_k\}$ von **Zwischenzuständen** gelesen werden können.
- Insbesondere $\widetilde{R}^0_{i,j} = \{x \in \Sigma \cup \{\epsilon\} \mid \delta^*(q_i, x) = q_j$ oder $(x = \epsilon) \wedge (i = j)\}$
- Somit: $L(A) = \bigcup\limits_{q_j \in F} \widetilde{R}^n_{1,j}$

Kleene's Konstruktion

- Sei $A = (\{q_1, \ldots, q_n\}, \Sigma, \delta, q_1, F)$ ein DFA
- Sei für $0 \leq k \leq n$ und $1 \leq i,\ j \leq n$:

$$\widetilde{R}^k_{i,j} := \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \delta(q_i, w) = q_j \;\; \wedge \;\; \Big[(\forall u, v \in \Sigma^+ : w = uv \wedge \\ \delta^*(q_i, u) = q_r) \Longrightarrow r \leq k\Big] \end{array} \right\}$$

- $\widetilde{R}^k_{i,j}$ enthält alle Wörter, die auf Pfaden von q_i nach q_j über die Menge $\{q_1, \ldots, q_k\}$ von **Zwischenzuständen** gelesen werden können.
- Insbesondere $\widetilde{R}^0_{i,j} = \{x \in \Sigma \cup \{\epsilon\} \mid \delta^*(q_i, x) = q_j$ oder $(x = \epsilon) \wedge (i = j)\}$
- Somit: $L(A) = \bigcup_{q_j \in F} \widetilde{R}^n_{1,j}$

Kann man die Mengen $\tilde{R}^k_{i,j}$ nur mit $\cup$, $\cdot$ und * definieren?

die Rekursionsformel

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

Zu zeigen: $\tilde{R}^k_{i,j} = R^k_{i,j}$

- **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 q_{m+1} nicht vor $\Rightarrow w \in R^m_{i,j}$.

  – Kommt q_{m+1} auf dem Pfad vor $\Rightarrow$ Zerlegung:

$$q_i \xrightarrow[u]{*} \boxed{q_{m+1}} \xrightarrow[v_1]{*} \boxed{q_{m+1}} \xrightarrow[v_2]{*} \ldots \xrightarrow[v_r]{*} \boxed{q_{m+1}} \xrightarrow[w]{*} q_j$$

Beweis (Forts.)

$$q_i \xrightarrow[u]{*} \boxed{q_{m+1}} \xrightarrow[v_1]{*} \boxed{q_{m+1}} \xrightarrow[v_2]{*} \ldots \xrightarrow[v_r]{*} \boxed{q_{m+1}} \xrightarrow[w]{*} q_j$$

- Die Pfade mit den Wörtern $u, v_1, v_2, \ldots, v_r, w$ durchlaufen nur Zustände der Menge $\{q_1, \ldots, q_m\}$.

- 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:

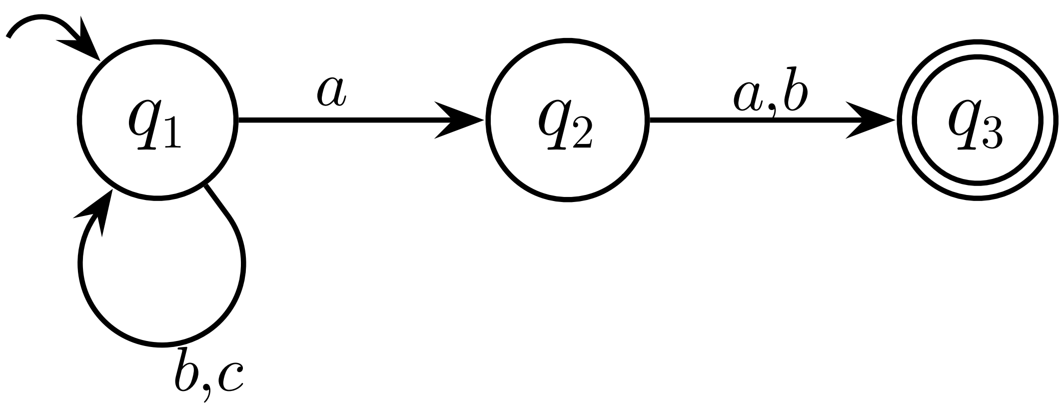
$$\widetilde{R}^{m+1}_{i,j} \;\subseteq\; R^m_{i,j} \;\cup\; R^m_{i,m+1} \cdot (R^m_{m+1,m+1})^* \cdot R^m_{m+1,j}.$$

- Die Umkehrung dieser Inklusion ist leicht ersichtlich.

Also gilt $\tilde{R}^k_{i,j} = R^k_{i,j}$.

Beispielkonstruktion

ohne Zwischenzustände:



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

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

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

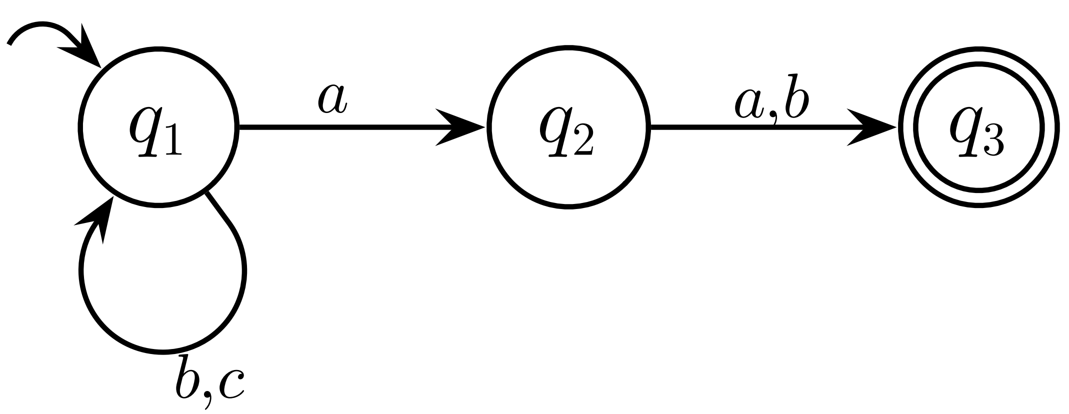
$$R^0_{2,2} = \{\epsilon\}$$

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

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

Beispielkonstruktion [mit Zwischenzustand (q_1)]

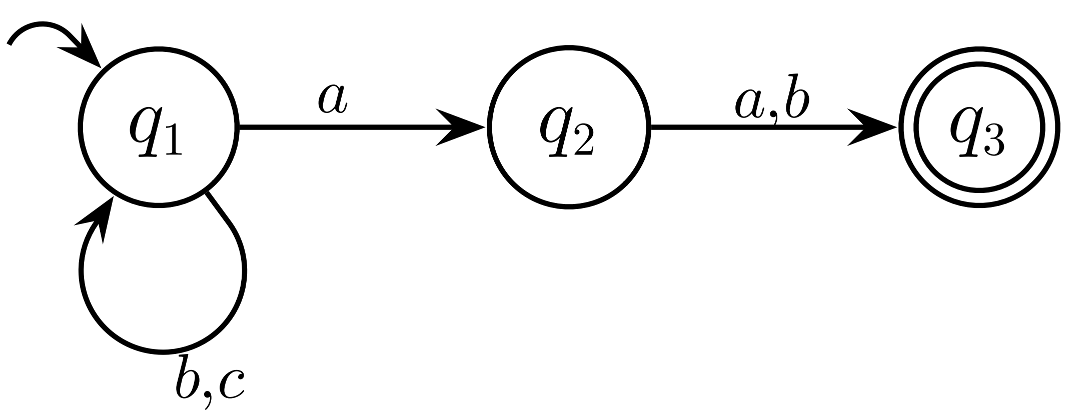
$$R^1_{1,1} \quad = \quad R^0_{1,1} \cup R^0_{1,1} \cdot (R^0_{1,1})^* \cdot R^0_{1,1}$$



Beispielkonstruktion [mit Zwischenzustand (q_1)]

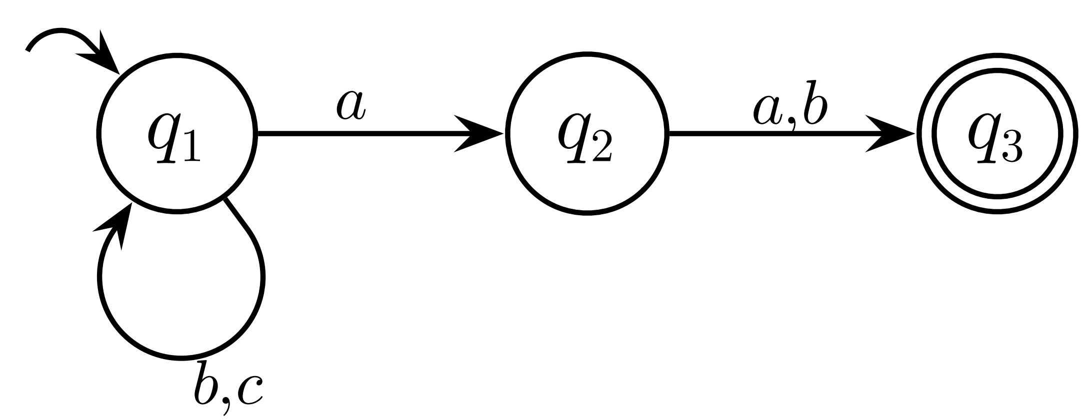
Ausklammern!

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



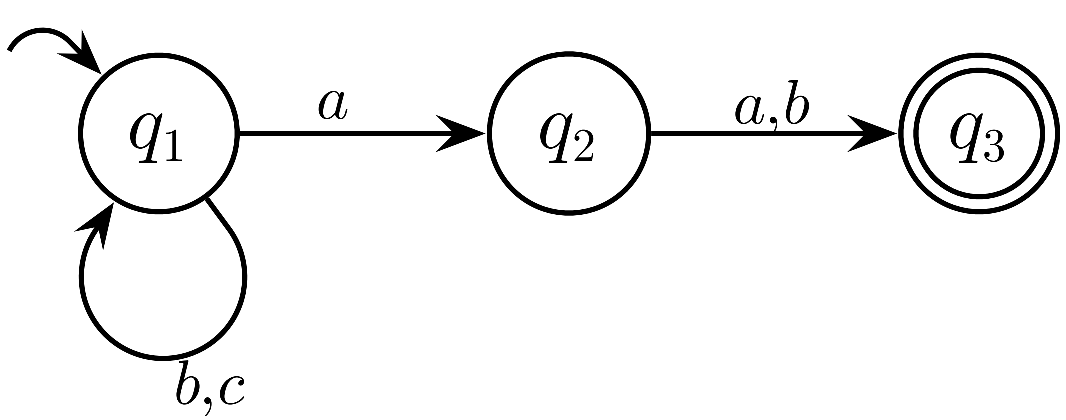
Beispielkonstruktion [mit Zwischenzustand (q_1)]

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



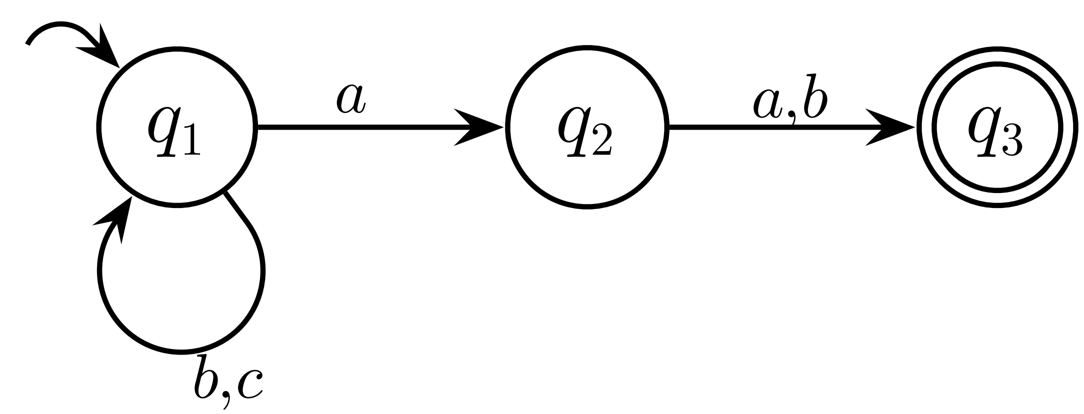
Beispielkonstruktion [mit Zwischenzustand (q_1)]

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



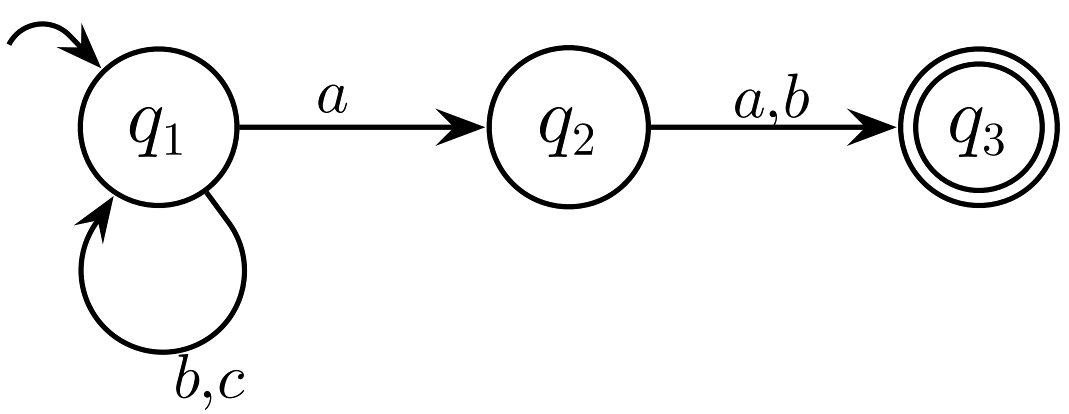
Beispielkonstruktion [mit Zwischenzustand (q_1)]

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



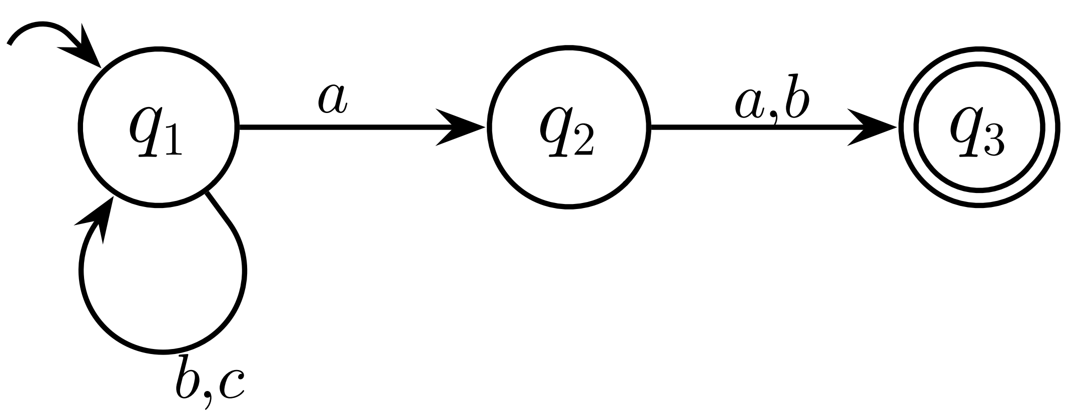
Beispielkonstruktion [mit Zwischenzustand (q_1)]

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



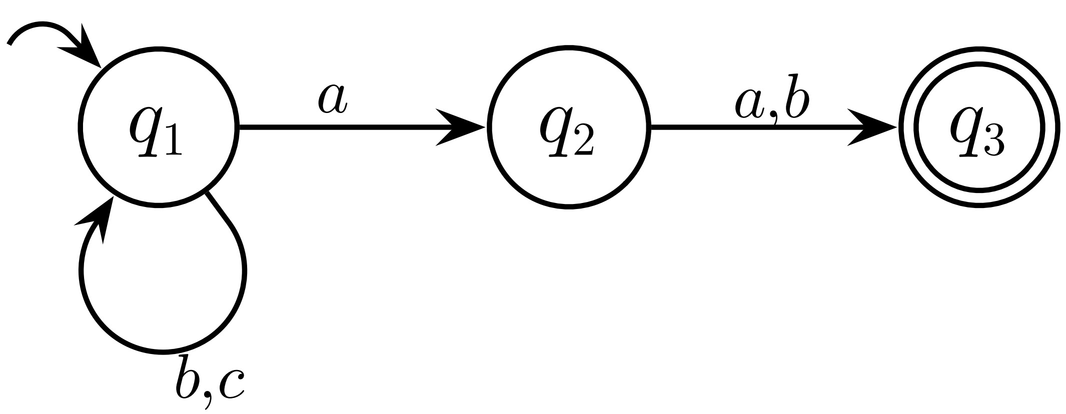
Beispielkonstruktion [mit Zwischenzustand (q_1)]

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



Beispielkonstruktion [mit Zwischenzustand (q_1)]

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



... usw. bis $R^3_{1,3}$ berechnet wurde.

Zwischenstand:

Zwischenergebnis:

Folgende Familien sind identisch:

- DFA_Σ
- NFA_Σ
- REG_Σ
- Die Menge der Sprachen, die sich durch einen regulären Ausdruck beschreiben lassen.
- ϵFA_Σ

Ausblick:
verallgemeineter NFA

verallgemeinerter ε-FA = GFA

Eingabe wird teilwortweise gelesen!

Definition 14.7:

Sei Σ ein Alphabet. Dann wird $A := (Q, \Sigma, \delta_*, S_0, F)$, mit:

$$\delta_* \subseteq Q \times (\Sigma^*) \times Q, \quad \text{bei} \quad |\delta_*| < \infty \quad \text{und} \quad S_0 \cup F \subseteq Q$$

verallgemeinerter endlicher Automat (GFA) genannt.

(in Vossen/Witt wird nur $S_0 = \{s_0\}$ betrachtet!)

Die Relation $\vdash$ wird nun an die δ_*-Übergänge angepasst:

Definition 14.8:

$$\forall v, w \in \Sigma^* : (p, vw) \vdash (q, w) \iff (p, v, q) \in \delta_*.$$

$$L(A) := \{w \in \Sigma^* \mid (s_0, w) \vdash^* (s, \epsilon), s_0 \in S_0, s \in F\}.$$

GFA = $\mathcal{R}eg$

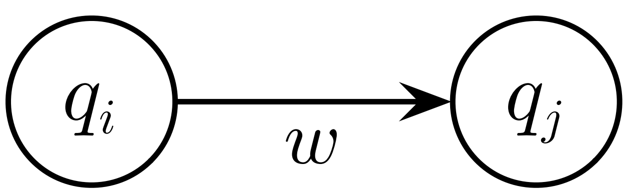
Satz 14.5:

Jeder *GFA* akzeptiert eine reguläre Menge.

Beweis:

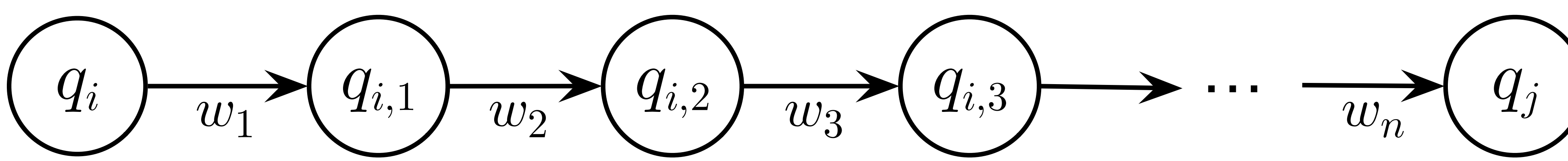
Sei $A := (Q, \Sigma, \delta_*, S_0, F)$ ein *GFA* mit Kantenbeschriftungen aus Σ^*.
Ein äquivalenter ϵ-*FA* $B := (Q, \Sigma, \delta', S_0', F')$ mit Kantenbeschriftung aus $\Sigma \cup \{\epsilon\}$, wird wie hier nur informal erklärt:

Zu jeder Kante:



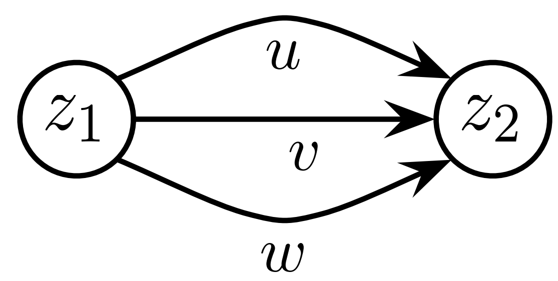
wobei $w = w_1 w_2 w_3 \dots w_n$

konstruiere neue Zustände und Kanten:

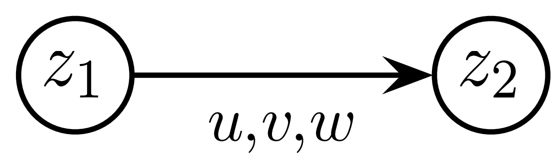


Abkürzende Notationen

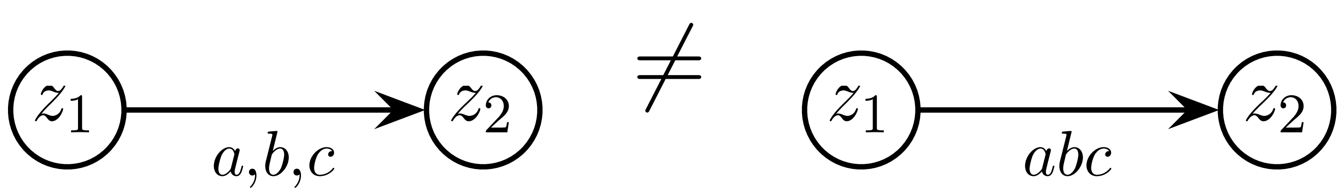
Statt



schreiben wir abkürzend:



Zu beachten:



Homomorphismen

Definition 14.43 *Seien Σ und Γ zwei Alphabete.*

Ein Homomorphismus ist eine Abbildung $h : \Sigma^* \to \Gamma^*$ zwischen den Wortmengen, die die Monoidstruktur erhält, d.h. für alle $u, v \in \Sigma^*$ gilt:

$$h(u \cdot v) = h(u) \cdot h(v)$$

Also ist ein jeder Homomorphismus $h : \Sigma^* \to \Gamma^*$ eindeutig festgelegt, wenn wir $h(a)$ für alle $a \in \Sigma$ kennen:

$$h(w) = h(a_1 \cdots a_n) = h(a_1) \cdots h(a_n)$$

Homomorphismen

Definition 14.43 *Seien Σ und Γ zwei Alphabete.*

Ein Homomorphismus ist eine Abbildung $h : \Sigma^* \to \Gamma^*$ zwischen den Wortmengen, die die Monoidstruktur erhält, d.h. für alle $u, v \in \Sigma^*$ gilt:

$$h(u \cdot v) = h(u) \cdot h(v)$$

Also ist ein jeder Homomorphismus $h : \Sigma^* \to \Gamma^*$ eindeutig festgelegt, wenn wir $h(a)$ für alle $a \in \Sigma$ kennen:

$$h(w) = h(a_1 \cdots a_n) = h(a_1) \cdots h(a_n)$$

Satz 14.1 *Für jede reguläre Menge $R \subseteq \Sigma^*$ und jeden Homomorphismus $h : \Sigma^* \to \Gamma^*$ ist auch das homomorphe Bild $h(R)$ regulär.*

$$h(R) := \{h(w) \mid w \in R\}$$

Beweis: Zu der regulären Menge $R \subseteq \Sigma^*$ existiert ein akzeptierender DFA A.

Ersetzen wir in A jede Kante $p \xrightarrow{a} q$ durch das Bild $p \xrightarrow{h(a)} q$, so erhalten wir einen GFA, der $h(R)$ akzeptiert.

Endstand für heute!

Zwischenergebnis:

Folgende Familien sind identisch:

- DFA_{Σ}
- NFA_{Σ}
- REG_{Σ}
- Die Menge der Sprachen, die sich durch einen regulären Ausdruck beschreiben lassen.
- $\epsilon - FA_{\Sigma}$
- GFA_{Σ}
- Die Menge der Sprachen, die sich durch eine einseitig lineare CFG generieren lassen.

Ausblick: reguläre und andere kontextfreie Grammatiken!