Theoretische Informatik I

Einheit 2.3

Reguläre Ausdrücke

1. Anwendungen
2. Syntax und Semantik
3. Vereinfachungsregeln
4. Beziehung zu endlichen Automaten

EINE ALGEBRAISCHE BESCHREIBUNG FÜR SPRACHEN

- **Automaten beschreiben Abarbeitung von Sprachen**
  - **Operationale Semantik**: Symbole führen zu Zustandsänderungen
  - Bestimmte Wörter bzw. Symbolketten werden durch Zustände akzeptiert
  - Für Automaten ist Sprache $\hat{=}$ Menge der akzeptierten Wörter
- **Wie beschreibt man Eigenschaften von Wörtern?**
  - **Deklarative Semantik**: äußere Form von Zeichenreihen einer Sprache

    z.B. *Wörter haben eine führende Null, dann beliebig viele Einsen*
  - Anwendungen brauchen präzise Beschreibungssprache für Wörter
    - Grundeinheiten von Programmiersprachen, Suchmuster für Browser, . . .
- **Reguläre Ausdrücke als formale Syntax**
  - Kurze, prägnante Beschreibung des Aufbaus der Wörter einer Sprache

    z.B. `01`*: “Zuerst eine Null, dann beliebig viele Einsen”

ANWENDUNG: TEXTSUCHE

- **Suche nach Mustern in Texten**
  - Suche ob/wo/wie oft eine bestimmte Zeichenkette im Text erscheint
  - Textmuster kann Platzhalter enthalten

ANWENDUNG: TEXTSUCHE

- **Suche nach Mustern in Texten**
  - Suche ob/wo/wie oft eine bestimmte Zeichenkette im Text erscheint
  - Textmuster kann Platzhalter enthalten
- **Beschreibe Textmuster durch reguläre Ausdrücke**
  - Zahl: Ziffernfolge dann evtl. Punkt und nichtleere Ziffernfolge
  - Formaler Ausdruck:

$$\underline{(0+1+..+9)}^{*}(\underline{\epsilon}+(.\,\underline{(0+1+..+9)}\boxed{(0+1+..+9)^{*}}))$$

$$[0-9]^{*}(\varepsilon+(.\ [0-9]^{+}))$$

ANWENDUNG: TEXTSUCHE

- **Suche nach Mustern in Texten**
  - Suche ob/wo/wie oft eine bestimmte Zeichenkette im Text erscheint
  - Textmuster kann Platzhalter enthalten
- **Beschreibe Textmuster durch reguläre Ausdrücke**
  - Zahl: Ziffernfolge dann evtl. Punkt und nichtleere Ziffernfolge
  - Formaler Ausdruck:

    $(0+1+..+9)^*(\epsilon+(.(0+1+..+9)(0+1+..+9)^*))$
- **Vielfältige Anwendungen**
  - Google Suche nach einfachen Texten
  - Erweiterte Google Suche nach Textmustern
  - Unix Kommando grep: suche nach Textmustern in Dateien
  - Programmiersprachen wie PERL und AWK
  - Textsuche und Textersetzung in Emacs
  - Lexikalische Analyse in Compilern

ANWENDUNG: LEXIKALISCHE ANALYSE

Wichtigster Grundbestandteil von Compilern

- **Reguläre Ausdrücke beschreiben Token**
  - Logische Grundeinheiten von Programmiersprachen
  - z.B. Schlüsselwörter, Bezeichner, Dezimalzahlen, . . .
- **“Lexer” transformieren reguläre Ausdrücke in Analyseprogramme**
  - Analyse kann die Token der Programmiersprache identifizieren
  - Zugrundeliegende Technik:
    Umwandlung regulärer Ausdrücke in DEAs

Reguläre Ausdrücke präzisiert (Syntax)

- **Syntax:** **Terme über** $\Sigma \cup \{\emptyset, \epsilon, +, \circ, {}^*, (,)\}$

  Reguläre Ausdrücke sind induktiv wie folgt definiert

  – $E = \mathtt{a}$ ist ein regulärer Ausdruck für jedes $a \in \Sigma$

  $(0\ 1) + (1)^*$

  – $E = \emptyset$ und $F = \epsilon$ sind reguläre Ausdrücke

  – Sind E und F reguläre Ausdrücke, dann sind auch

  $0 \circ (1)^*$

  $\boldsymbol{E \circ F}$, $\boldsymbol{E^*}$, $\boldsymbol{E+F}$ und $\boldsymbol{(E)}$ sind reguläre Ausdrücke

  Mehr Ausdrücke möglich, aber nicht erforderlich

- **Konventionen zur Vereinfachung**

  $a+(b \circ c)$

  – $E\cancel{\circ}F$ wird üblicherweise als $\boldsymbol{EF}$ abgekürzt

  – Definitorische Abkürzungen: $\boldsymbol{E^+} \equiv EE^*$, $[\boldsymbol{a_1 \ldots a_n}] \equiv a_1 + \ldots + a_n$

  – Prioritätsregelungen ermöglichen, überflüssige Klammern wegzulassen

  - $*$ ("Sternoperator") bindet stärker als $\circ$, und dies stärker als $+$
  - Verkettung $\circ$ und Alternative $+$ sind assoziativ

  $0+(1+2)$

  $(\qquad\quad)$

REGULÄRE AUSDRÜCKE PRÄZISIERT (SEMANTIK)

- **Reguläre Ausdrücke beschreiben Sprachen über Σ**
- **Die Sprache $L(E)$ ist induktiv definiert**
  - Für für alle $a \in \Sigma$ ist $L(\mathsf{a}) = \boxed{\{a\}}$ (einelementige Sprache, die nur a enthält)

    $\underline{L(\emptyset)}$ ist die leere Sprache (üblicherweise geschrieben als $\underline{\emptyset}$ oder $\underline{\{\}}$)

    $\underline{L(\epsilon)} = \boxed{\{\epsilon\}}$ (einelementige Sprache, die nur das leere Wort enthält)
  - $\underline{L(E\circ F)} = L(E)\circ L(F) = \{\underline{v\,w} \mid \underline{v \in L(E)} \wedge \underline{w \in L(F)}\}$

    ∘ steht für die Verkettung (der Wörter) zweier Sprachen
  - $\underline{L(E^*)} = (L(E))^* = \{\underline{w_1 w_2 .. w_n} \mid n \in \mathbb{N} \wedge \underline{w_i \in L(E)}\}$

    * steht für Verkettung beliebig vieler Wörter einer Sprache (Kleene'sche Hülle)
  - $\underline{L(E+F)} = L(E) \cup L(F) = \{w \in \Sigma^* \mid \underline{w \in L(E)} \vee \underline{w \in L(F)}\}$

    + steht für die Vereinigung zweier Sprachen
  - $\underline{L(\,(E)\,)} = L(E)$

SPRACHEN VS. AUSDRÜCKE

- **Sprachen sind Mengen von Wörtern**
  - Abstraktes semantisches Konzept: Ungeordnete Kollektion von Wörtern
  - Beschreibung von Mengen (auf Folie, Tafel,...) benötigt textuelle Notation
  - Notation benutzt Kurzschreibweisen wie ∪, ∘, * für Mengenoperationen

    ... aber ist selbst nur ein Hilfsmittel zur Kommunikation

$\{0,1,2,3,17,21\}$

- **Reguläre Ausdrücke sind Terme**
  - Eine syntaktische Beschreibungsform, die ein Computer versteht
  - Reguläre Ausdrücke werden zur Beschreibung von Sprachen benutzt

    und sind ähnlich zur Standardnotation von Mengen
- **Reguläre Ausdrücke sind selbst keine Sprachen**
  - Unterscheide Ausdruck E von Sprache des Ausdrucks $L(E)$
  - Man verzichtet auf den Unterschied wenn der Kontext eindeutig ist

Beispiele regulärer Ausdrücke

- a*ba*

$L(a^*ba^*)$

$= \{b, ab, ba, aba, aab, aaba, \ldots\}$

$= \{ w \in \{a,b\}^* \mid$ w enthält genau ein b $\}$

Beispiele regulärer Ausdrücke

- $\texttt{a}^*\texttt{ba}^*$
  - steht für die Menge aller Wörter über $\Sigma=\{a,b\}$, die genau ein b enthalten
  - $L(\texttt{a}^*\texttt{ba}^*) \;=\; \{w \in \Sigma^* \mid \underline{w \text{ enthält genau ein } b}\} \;=\; \{w \in \Sigma^* \mid \underline{\#_b(w)}=1\}$

BEISPIELE REGULÄRER AUSDRÜCKE

- $\texttt{a}^*\texttt{ba}^*$
  - steht für die Menge aller Wörter über $\Sigma=\{a,b\}$, die genau ein b enthalten
  - $L(\texttt{a}^*\texttt{ba}^*) = \{w \in \Sigma^* \mid w \text{ enthält genau ein } b\} = \{w \in \Sigma^* \mid \#_b(w)=1\}$

Kurzel

- $\Sigma^*\texttt{b}\Sigma^*$ $= (a+b)^* b \, (a+b)^*$
  - steht für $\{w \in \Sigma^* \mid w \text{ enthält mindestens ein } b\} = \{w \in \Sigma^* \mid \underline{\#_b(w)\geq 1}\}$

BEISPIELE REGULÄRER AUSDRÜCKE

- a^*ba^*
  - steht für die Menge aller Wörter über $\Sigma=\{a,b\}$, die genau ein b enthalten
  - $L(a^*ba^*) = \{w \in \Sigma^* \mid w \text{ enthält genau ein } b\} = \{w \in \Sigma^* \mid \#_b(w)=1\}$
- $\Sigma^*b\Sigma^*$
  - steht für $\{w \in \Sigma^* \mid w \text{ enthält mindestens ein } b\} = \{w \in \Sigma^* \mid \#_b(w)\geq 1\}$
- $a^*\,\underline{(b+\epsilon)}\,a^*$
  - steht für $\{w \in \Sigma^* \mid w \text{ enthält maximal ein } b\} = \{w \in \Sigma^* \mid \#_b(w)\leq 1\}$

BEISPIELE REGULÄRER AUSDRÜCKE

- $\mathtt{a^*ba^*}$
  - steht für die Menge aller Wörter über Σ={a,b}, die genau ein b enthalten
  - $L(\mathtt{a^*ba^*}) = \{w \in \Sigma^* \mid w \text{ enthält genau ein } b\} = \{w \in \Sigma^* \mid \#_b(w)=1\}$
- $\mathtt{\Sigma^*b\Sigma^*}$
  - steht für $\{w \in \Sigma^* \mid w \text{ enthält mindestens ein } b\} = \{w \in \Sigma^* \mid \#_b(w)\geq 1\}$
- $\mathtt{a^*\,(b+\epsilon)\,a^*}$
  - steht für $\{w \in \Sigma^* \mid w \text{ enthält maximal ein } b\} = \{w \in \Sigma^* \mid \#_b(w)\leq 1\}$
- $\mathtt{a\emptyset}$ $L(a\emptyset) = L(a) \circ L(\emptyset) = \{a\} \circ \emptyset$
  - steht für die <u>leere Sprache</u>, denn die Verkettung einer Sprache mit der leeren Sprache ist immer leer

$\{vw \mid v \in \{a\} \wedge \underline{w \in \emptyset}\}$

$=$

$\{\}$

BEISPIELE REGULÄRER AUSDRÜCKE

$0^* = 0$ ~~$0 =$~~ $0^0 = 1$ $n^0 = 1$

- $\mathtt{a}^*\mathtt{b}\mathtt{a}^*$
  - steht für die Menge aller Wörter über Σ={a,b}, die genau ein b enthalten
  - $L(\mathtt{a}^*\mathtt{b}\mathtt{a}^*) = \{w \in \Sigma^* \mid w$ enthält genau ein $b\} = \{w \in \Sigma^* \mid \#_b(w) = 1\}$
- $\Sigma^*\mathtt{b}\Sigma^*$
  - steht für $\{w \in \Sigma^* \mid w$ enthält mindestens ein $b\} = \{w \in \Sigma^* \mid \#_b(w) \geq 1\}$
- $\mathtt{a}^*\,(\mathtt{b}+\epsilon)\,\mathtt{a}^*$
  - steht für $\{w \in \Sigma^* \mid w$ enthält maximal ein $b\} = \{w \in \Sigma^* \mid \#_b(w) \leq 1\}$
- $\mathtt{a}\emptyset$
  - steht für die leere Sprache, denn die Verkettung einer Sprache mit der leeren Sprache ist immer leer

$\{0, 1, \ldots\}$ $\emptyset$

- $\emptyset^*$ $= \{ w_1 .. w_n \mid \underline{n \overset{=}{\in} \mathbb{N}}\ \ \underline{w_i \in \cancel{L(\emptyset)}} \}$
  - steht für die Menge $\{\epsilon\}$, denn die beliebige Verkettung von Wörtern einer Menge enthält immer das leere Wort

$L(\emptyset^*) = \{\varepsilon\}$

ENTWICKLUNG REGULÄRER AUSDRÜCKE

Beschreibe Menge aller Wörter, in denen 0 und 1 abwechseln

1. Regulärer Ausdruck für die Sprache $\{01\}$

- `0` repräsentiert $\{0\}$, `1` repräsentiert $\{1\}$
- Also ist $L(\texttt{01}) = L(\texttt{0}) \circ L(\texttt{1}) = \{0\} \circ \{1\} = \{01\}$

ENTWICKLUNG REGULÄRER AUSDRÜCKE

Beschreibe Menge aller Wörter, in denen 0 und 1 abwechseln

!

1. Regulärer Ausdruck für die Sprache $\{01\}$

- $\texttt{0}$ repräsentiert $\{0\}$, $\texttt{1}$ repräsentiert $\{1\}$
- Also ist $L(\texttt{01}) = L(\texttt{0})\circ L(\texttt{1}) = \{0\}\circ\{1\} = \{01\}$

2. Erzeuge $\{01, 0101, 010101, ..\}$ durch Sternbildung

- $L(\texttt{(01)}^*) = L(\texttt{01})^* = \{01\}^* = \{\epsilon, 01, 0101, 010101, \ldots\}$

ENTWICKLUNG REGULÄRER AUSDRÜCKE

Beschreibe Menge aller Wörter, in denen 0 und 1 abwechseln

1. Regulärer Ausdruck für die Sprache $\{01\}$

- $\texttt{0}$ repräsentiert $\{0\}$, $\texttt{1}$ repräsentiert $\{1\}$
- Also ist $L(\texttt{01}) = L(\texttt{0})\circ L(\texttt{1}) = \{0\}\circ\{1\} = \{01\}$

2. Erzeuge $\{01, 0101, 010101, ..\}$ durch Sternbildung

- $L(\texttt{(01)}^*) = L(\texttt{01})^* = \{01\}^* = \{\epsilon, 01, 0101, 010101, \ldots\}$

3. Manche Wörter nicht erfaßt

- Start mit Eins statt Null: $\underline{\texttt{(10)}^*}$
- Start und Ende mit Null: $\underline{\texttt{(01)}^*}\underline{\texttt{0}}$
- Start und Ende mit Eins: $\underline{\texttt{(10)}^*}\underline{\texttt{1}}$

Vollständiger Ausdruck: $\underline{\texttt{(01)}^* + \texttt{(10)}^* + \texttt{(01)}^*\texttt{0} + \texttt{(10)}^*\texttt{1}}$

$$(\varepsilon + 1)\,(01)^*\,(0 + \varepsilon)$$

ENTWICKLUNG REGULÄRER AUSDRÜCKE

Beschreibe Menge aller Wörter, in denen 0 und 1 abwechseln

1. Regulärer Ausdruck für die Sprache $\{01\}$

– `0` repräsentiert $\{0\}$, `1` repräsentiert $\{1\}$

– Also ist $L(\texttt{01}) = L(\texttt{0})\circ L(\texttt{1}) = \{0\}\circ\{1\} = \{01\}$

2. Erzeuge $\{01, 0101, 010101, ..\}$ durch Sternbildung

– $L(\texttt{(01)}^*) = L(\texttt{01})^* = \{01\}^* = \{\epsilon, 01, 0101, 010101, \ldots\}$

3. Manche Wörter nicht erfaßt

– Start mit Eins statt Null: $\texttt{(10)}^*$

– Start und Ende mit Null: $\texttt{(01)}^*\texttt{0}$

– Start und Ende mit Eins: $\texttt{(10)}^*\texttt{1}$

Vollständiger Ausdruck: $\texttt{(01)}^* + \texttt{(10)}^* + \texttt{(01)}^*\texttt{0} + \texttt{(10)}^*\texttt{1}$

4. Es geht auch kürzer

– Optional 1 am Anfang oder 0 am Ende: $\texttt{(}\epsilon\texttt{+1)(01)}^*\texttt{(}\epsilon\texttt{+0)}$

BESTIMMUNG DER SEMANTIK VON $(\epsilon+1)(01)^*(\epsilon+0)$

$L((\epsilon+1)(01)^*(\epsilon+0))$

BESTIMMUNG DER SEMANTIK VON $(\epsilon+1)(01)^*(\epsilon+0)$

$$
\begin{aligned}
& L((\epsilon+1)(01)^*(\epsilon+0)) \\
= {} & \underline{L((\epsilon+1))} \circ \underline{L((01)^*)} \circ \underline{L((\epsilon+0))} \\
= {} & \underline{L(\epsilon) \cup L(1)} \circ \underline{L((01))^*} \circ \underline{L(\epsilon) \cup L(0)}
\end{aligned}
$$

BESTIMMUNG DER SEMANTIK VON $\mathtt{(\epsilon+1)(01)^*(\epsilon+0)}$

$$\begin{aligned}
& L(\mathtt{(\epsilon+1)(01)^*(\epsilon+0)}) \\
= {} & L(\mathtt{(\epsilon+1)}) \circ L(\mathtt{(01)^*}) \circ L(\mathtt{(\epsilon+0)}) \\
= {} & \underline{L(\epsilon)} \cup \underline{L(\mathtt{1})} \circ \underline{L(\mathtt{(01)})}^* \circ L(\epsilon) \cup L(\mathtt{0}) \\
= {} & \underline{\{\epsilon\}} \cup \underline{\{1\}} \circ \underline{(L(\mathtt{0}) \circ L(\mathtt{1}))}^* \circ \{\epsilon\} \cup \{0\} \\
= {} & \underline{\{\epsilon,1\}} \circ \underline{\{01\}}^* \circ \underline{\{\epsilon,0\}}
\end{aligned}$$

Bestimmung der Semantik von $\texttt{(}\epsilon\texttt{+1)(01)}^*\texttt{(}\epsilon\texttt{+0)}$

$$
\begin{aligned}
& L(\texttt{(}\epsilon\texttt{+1)(01)}^*\texttt{(}\epsilon\texttt{+0)}) \\
= {} & L(\texttt{(}\epsilon\texttt{+1)}) \circ L(\texttt{(01)}^*) \circ L(\texttt{(}\epsilon\texttt{+0)}) \\
= {} & L(\epsilon) \cup L(\texttt{1}) \circ L(\texttt{(01)})^* \circ L(\epsilon) \cup L(\texttt{0}) \\
= {} & \{\epsilon\} \cup \{1\} \circ (L(\texttt{0}) \circ L(\texttt{1}))^* \circ \{\epsilon\} \cup \{0\} \\
= {} & \{\epsilon, 1\} \circ \{01\}^* \circ \{\epsilon, 0\} \\
= {} & \{\epsilon, 1\} \circ \{w \mid \exists n \in \mathbb{N}.\ w = \underbrace{01 \ldots 01}_{n-mal}\} \circ \{\epsilon, 0\} \\
= {} & \{w \mid \exists n \in \mathbb{N}.\ w = \underbrace{01 \ldots 01}_{n-mal} \lor w = 1\underbrace{01 \ldots 01}_{n-mal} \\
& \qquad \lor w = \underbrace{01 \ldots 01}_{n-mal}0 \lor w = 1\underbrace{01 \ldots 01}_{n-mal}0\}
\end{aligned}
$$

= Die Menge aller Wörter, in denen 0 und 1 abwechseln

(Mühsamer Beweis durch Induktion)

“Rechenregeln” für reguläre Ausdrücke

Wie zeigt man $\boxed{\texttt{(01)}^*\texttt{+(10)}^*\texttt{+(01)}^*\texttt{0+(10)}^*\texttt{1}} \cong \boxed{\texttt{(}\epsilon\texttt{+1) (01)}^*\texttt{(}\epsilon\texttt{+0)}}$ **?**

“RECHENREGELN” FÜR REGULÄRE AUSDRÜCKE

Wie zeigt man `(01)*+(10)*+(01)*0+(10)*1` $\cong$ `(`ϵ`+1)(01)*(`ϵ`+0)` **?**

- **Definiere Äquivalenz von Ausdrücken**
  - $\boxed{\boldsymbol{E \cong F}}$, falls $\underline{L(E) = L(F)}$

Beweise algebraische Gesetze regulärer Ausdrücke

  - Liefert Hilfsmittel zur Vereinfachung regulärer Ausdrücke

“RECHENREGELN” FÜR REGULÄRE AUSDRÜCKE

Wie zeigt man $\texttt{(01)}^*\texttt{+(10)}^*\texttt{+(01)}^*\texttt{0+(10)}^*\texttt{1} \cong \texttt{(}\epsilon\texttt{+1)(01)}^*\texttt{(}\epsilon\texttt{+0)}$ **?**

- **Definiere Äquivalenz von Ausdrücken**
  - $E \cong F$, falls $L(E) = L(F)$

  Beweise algebraische Gesetze regulärer Ausdrücke
  - Liefert Hilfsmittel zur Vereinfachung regulärer Ausdrücke

~~403 ≠ 304~~

- **Gesetze für Einheiten und Annihilatoren**
  - $\emptyset+E \cong E \cong E+\emptyset$: $\quad L(\emptyset+E)$ ~~$= L(\emptyset)\cup L(E) = \emptyset\cup L(E) =$~~ $L(E)$
  - $\epsilon\circ E \cong E \cong E\circ\epsilon$: $\quad L(\epsilon\circ E) = L(\epsilon)\circ L(E) = \{\epsilon\}\circ L(E) = L(E)$
  - $\emptyset\circ\cancel{E} \cong \emptyset \cong E\circ\emptyset$: $\quad L(\emptyset\circ E) = L(\emptyset)\circ L(E) = \emptyset\circ L(E) = \emptyset = L(\emptyset)$

  Kommutativitätsgesetz für +
  - $E+F \cong F+E$: $\quad L(E+F) = L(E)\cup L(F) = L(F)\cup L(E) = L(F+E)$
  - Kommutativität von ∘ gilt nicht: $\quad = L(\texttt{01}) = \{01\} \neq \{10\} = L(\texttt{10})$

“RECHENREGELN” FÜR REGULÄRE AUSDRÜCKE II

$[w_1 [w_2] w_3]$

- **Gesetze für Assoziativität von ∘ und +**
  - $(E \circ F) \circ G \cong E \circ (F \circ G)$:

    $L((E \circ F) \circ G) = L(E \circ F) \circ L(G) = L(E) \circ L(F) \circ L(G) = L(E) \circ L(F \circ G) = L(E \circ (F \circ G))$
  - $(E + F) + G \cong E + (F + G)$:

    $L((E + F) + G) = L(E + F) \cup L(G) = L(E) \cup L(F) \cup L(G) = \ldots = L(E + (F + G))$

- **Distributivgesetze**

  $(a+b)\cdot c$

  $= a\cdot c + b \cdot c$

  $a + a$
  - $(E + F) \circ G \cong E \circ G + F \circ G$:

    $$\begin{aligned} L((E+F)\circ G) &= (L(E) \cup L(F)) \circ L(G) \\ &= \{w \in \Sigma^* \mid \exists u \in L(E) \cup L(F). \exists v \in L(G). w = uv\} \\ &= \{w \in \Sigma^* \mid \exists u \in L(E). \exists v \in L(G). w = uv \vee \exists u \in L(F). \exists v \in L(G). w = uv\} \\ &= \{w \in \Sigma^* \mid \exists u \in L(E). \exists v \in L(G). w = uv\} \cup \{w \in \Sigma^* \mid \exists u \in L(F). \exists v \in L(G). w = uv\} \\ &= L(E) \circ L(G) \cup L(F) \circ L(G) = \boxed{L(E \circ G + F \circ G)} \end{aligned}$$
  - $G \circ (E + F) \cong G \circ E + G \circ F$

  $\{w \mid w \in L_1 \vee w \in L_2\}$

  $= L_1 \cup L_2$

- **Idempotenz von +:** $E + E \cong E$
- **Hüllengesetze:** $\emptyset^* \cong \epsilon, \quad \epsilon^* \cong \epsilon, \quad (E^*)^* \cong E^*$

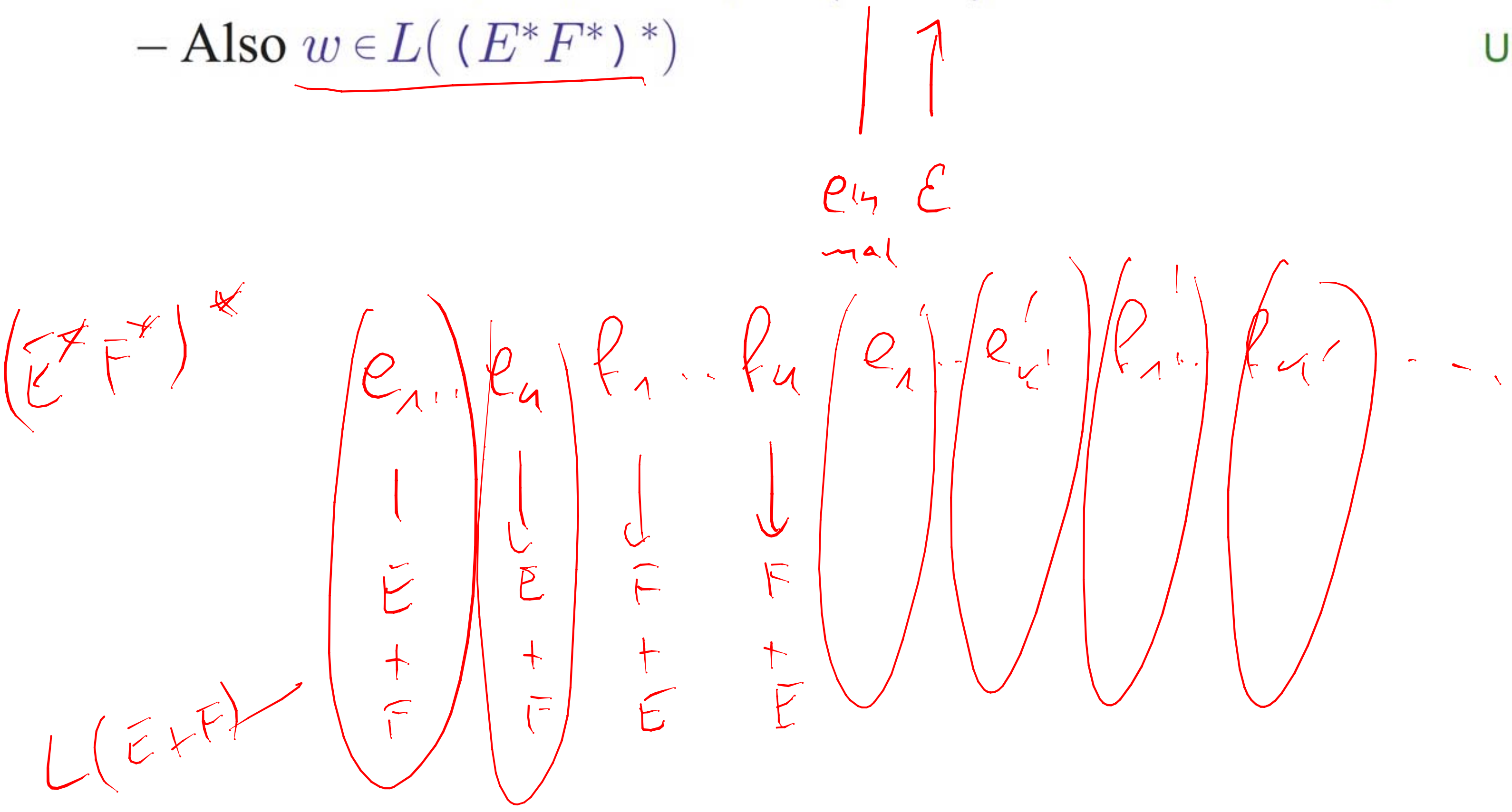
  $E^+ \cong E \circ E^* \cong E^* \circ E, \quad E^* \cong \epsilon + E^+$

$L \cup L = L$

BEWEISMETHODIK FÜR WEITERE ÄQUIVALENZEN

- **Beispiel: Nachweis von** $(E+F)^* \cong (E^*F^*)^*$
  - Sei $w \in L((E+F)^*)$
  - Dann $w = w_1..w_k$ mit $w_i \in L(E)$ oder $w_i \in L(F)$ für alle i
  - Dann $w = w_1..w_k$ mit $w_i \in L(E^*F^*)$ für alle i (semantisches Argument)
  - Also $w \in L((E^*F^*)^*)$

Umkehrung analog



BEWEISMETHODIK FÜR WEITERE ÄQUIVALENZEN

- **Beispiel: Nachweis von $(E+F)^* \cong (E^*F^*)^*$**
  - Sei $w \in L((E+F)^*)$
  - Dann $w = w_1..w_k$ mit $w_i \in L(E)$ oder $w_i \in L(F)$ für alle i
  - Dann $w = w_1..w_k$ mit $w_i \in L(E^*F^*)$ für alle i (semantisches Argument)
  - Also $w \in L((E^*F^*)^*)$ Umkehrung analog
- **Beweis verwendet keine Information über E und F**
  - Man könnte genauso gut $(a+b)^* \cong (a^*b^*)^*$ testen

    Ja → Beweis

    $(E+F)^* \cong (E^*F^*)^*$ gilt, weil $(a+b)^* \cong (a^*b^*)^*$ gilt
- **Allgemeines Beweisverfahren**
  - E regulärer Ausdruck mit Metavariablen $E_1,..,E_m$ für Sprachen $L_1,..,L_m$
  - Ersetze im Beweis für $E \cong F$ alle Metavariablen durch Symbole $a \in \Sigma$
  - Teste Äquivalenz der konkreten Ausdrücke mit automatischem Prüfverfahren ↦ Einheit 2.5

Korrektheitsbeweis: Induktion über Struktur regulärer Ausdrücke

UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

Sprachen regulärer Ausdrücke sind endlich erkennbar

Für jeden regulären Ausdruck E gibt es einen ϵ-NEA A mit

– A hat genau einen akzeptierenden Zustand q_f

– Der Startzustand von A ist in keinem $\delta_A(q,a)$ enthalten

– Für alle $a \in \Sigma$ ist $\delta_A(q_f,a) = \emptyset$

– $L(E) = L(A)$

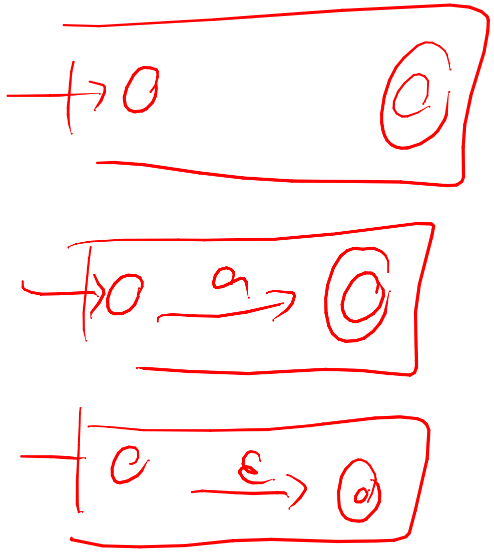
Beweis durch strukturelle Induktion über Aufbau regulärer Ausdrücke

+ , ∘ , *

$\emptyset$

a

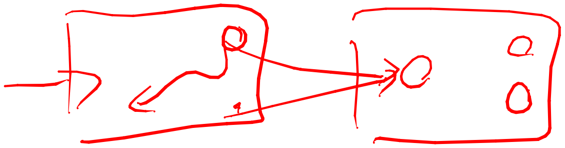
ϵ



✓

✓

✓



Umwandlung regulärer Ausdrücke in Automaten

Sprachen regulärer Ausdrücke sind endlich erkennbar

Für jeden regulären Ausdruck E gibt es einen ϵ-NEA A mit

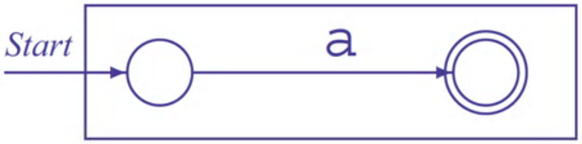
- A hat genau einen akzeptierenden Zustand q_f
- Der Startzustand von A ist in keinem $\delta_A(q,a)$ enthalten
- Für alle $a \in \Sigma$ ist $\delta_A(q_f,a) = \emptyset$
- $L(E) = L(A)$

Beweis durch strukturelle Induktion über Aufbau regulärer Ausdrücke

- **Induktionsanfänge**
  - Für $E = \epsilon$ wähle $A =$

  - Für $E = \emptyset$ wähle $A =$

  - Für $E = \mathtt{a}$ wähle $A =$



  - **Korrektheit offensichtlich**, da jeweils maximal ein Zustandsübergang

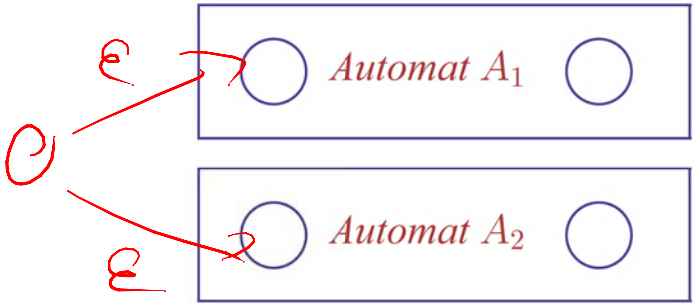
UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**

- **Induktionsschritt**

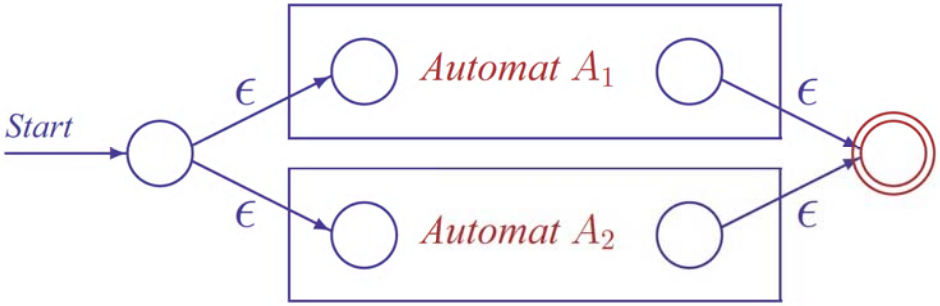
UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**
- **Induktionsschritt**
  - Für $E = E_1 + E_2$ wähle $A =$



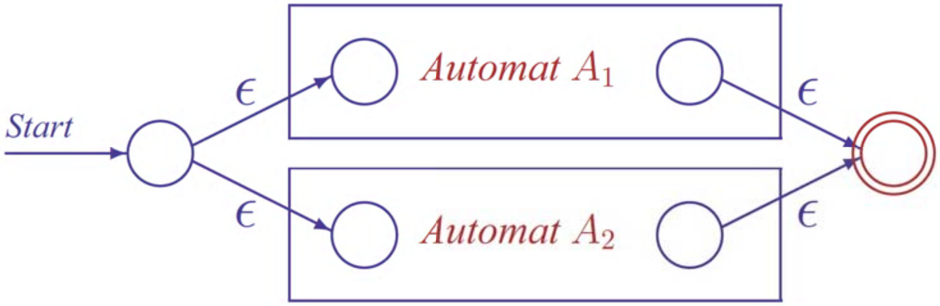
UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**
- **Induktionsschritt**
  - Für $E = E_1 + E_2$ wähle $A =$



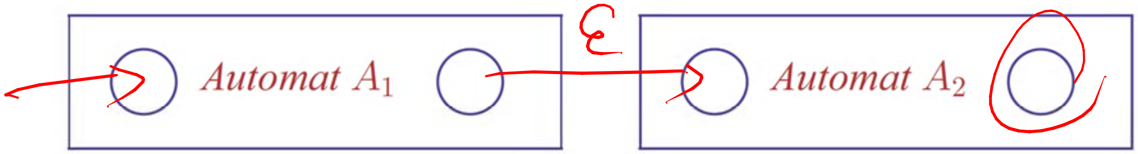
UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**
- **Induktionsschritt**
  - Für $E = E_1 + E_2$ wähle $A =$



  - Für $E = E_1 \circ E_2$ wähle

$A =$

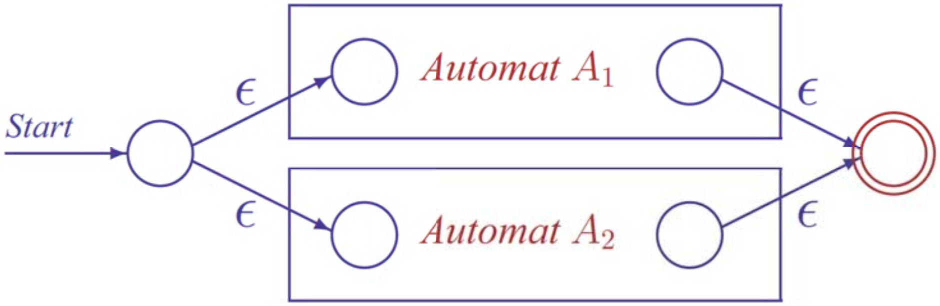


UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**

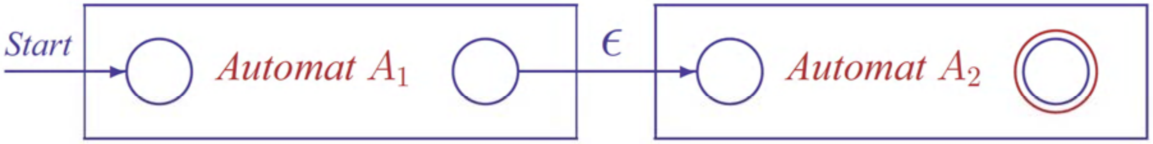
- **Induktionsschritt**

  – Für $E = E_1 + E_2$ wähle $A =$



  – Für $E = E_1 \circ E_2$ wähle

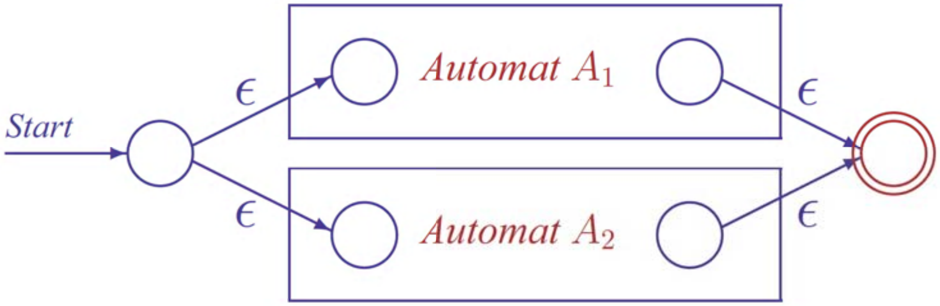
$A =$



UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

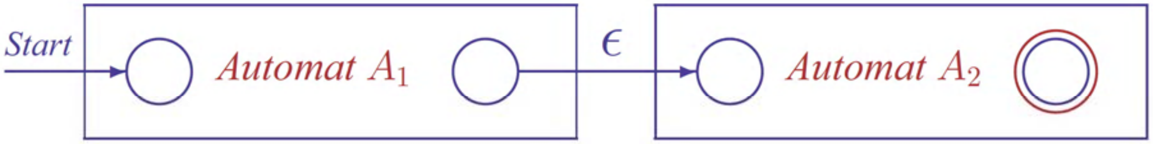
- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**

- **Induktionsschritt**
  - Für $E = E_1 + E_2$ wähle $A =$

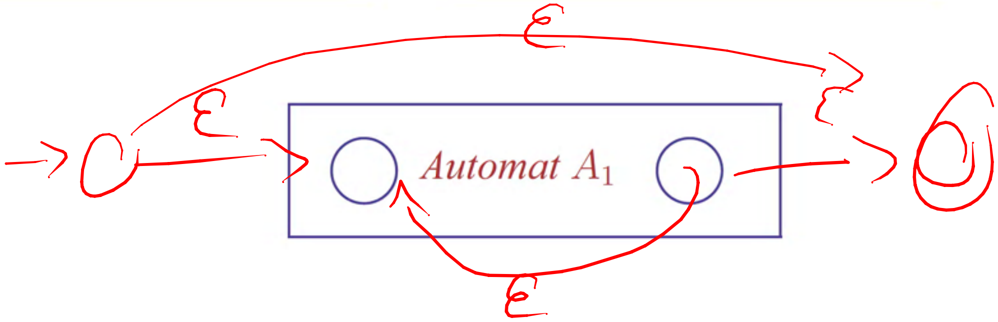


  - Für $E = E_1 \circ E_2$ wähle

$A =$

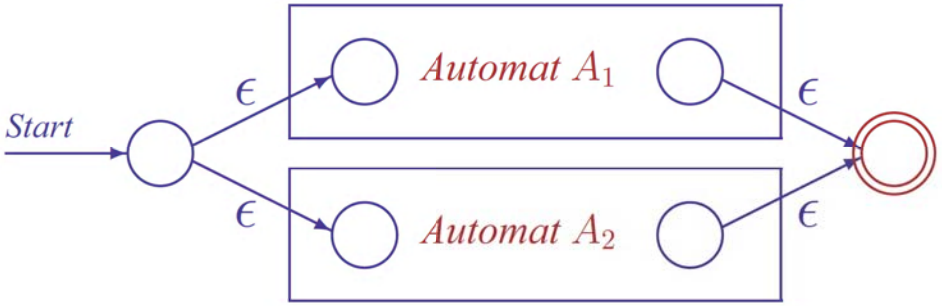


  - Für $E = E_1^*$ wähle $A =$



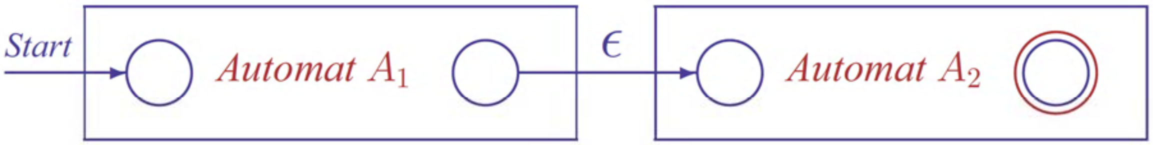
UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**
- **Induktionsschritt**
  - Für $E = E_1 + E_2$ wähle $A =$

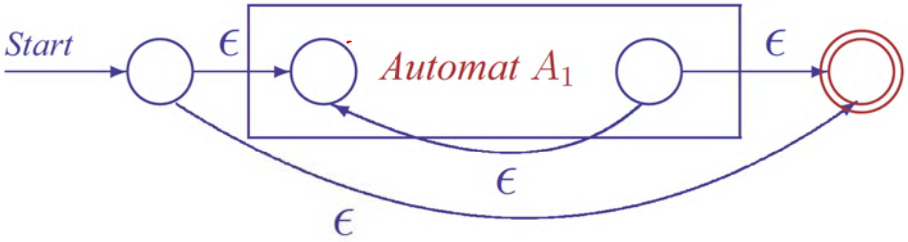


  - Für $E = E_1 \circ E_2$ wähle

$A =$

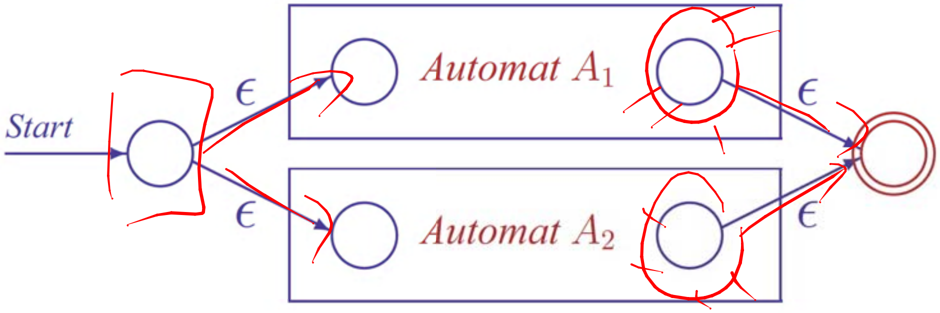


  - Für $E = E_1^*$ wähle $A =$



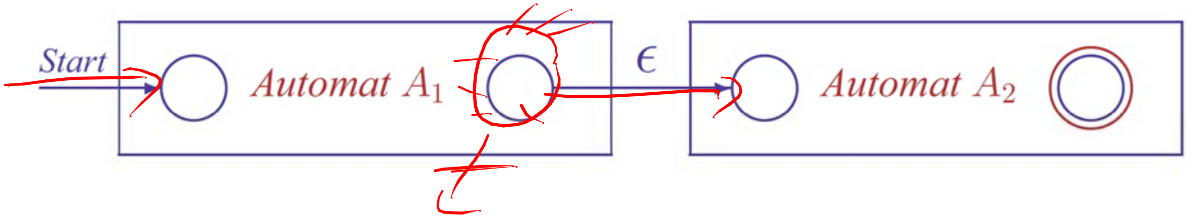
UMWANDLUNG REGULÄRER AUSDRÜCKE IN AUTOMATEN

- **Induktionsannahme: seien A_1 und A_2 ϵ-NEAs für E_1 und E_2**
- **Induktionsschritt**
  - Für $E = E_1 + E_2$ wähle $A =$

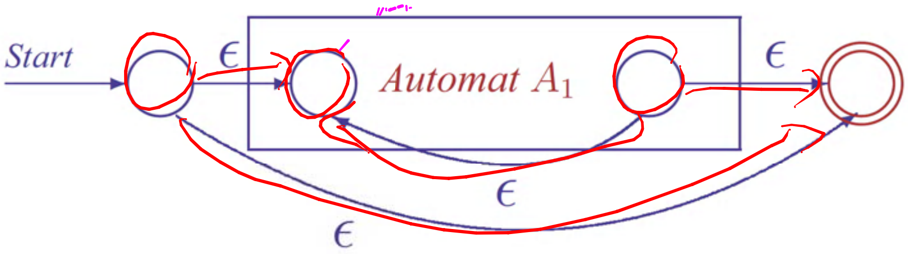


  - Für $E = E_1 \circ E_2$ wähle

$A =$



  - Für $E = E_1^*$ wähle $A =$



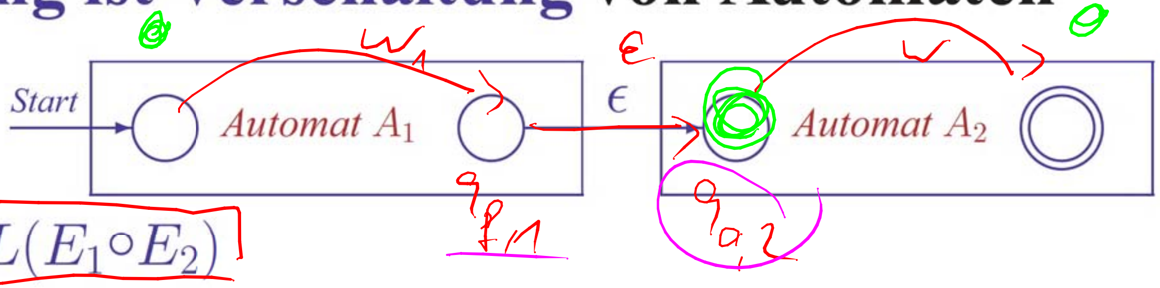
  - Für $E = (E_1)$ wähle $A = A_1$

KORREKTHEIT DER UMWANDLUNGEN

- **Klammern ändern nichts**
  - Es ist $L(\,(E_1)\,) = L(E_1) = L(A_1) = L(A)$

KORREKTHEIT DER UMWANDLUNGEN

- **Klammern ändern nichts** ✓
  - Es ist $L(\,(E_1)\,) = L(E_1) = L(A_1) = L(A)$
- **Verkettung ist Verschaltung von Automaten**



Es gilt $w \in L(E_1 \circ E_2)$

$\Rightarrow\ w \in L(E_1) \circ L(E_2) = L(A_1) \circ L(A_2)$

$\Rightarrow\ \exists u \in L(A_1).\exists v \in L(A_2).w = uv$

$\Rightarrow\ \exists u, v \in \Sigma^*.w = uv \wedge q_{f,1} \in \hat{\delta}_1(q_{0,1}, u) \wedge q_{f,2} \in \hat{\delta}_2(q_{0,2}, v)$

?? $\Rightarrow\ \exists u, v \in \Sigma^*.w = uv \wedge q_{0,2} \in \hat{\delta}(q_{0,1}, u) \wedge q_{f,2} \in \hat{\delta}(q_{0,2}, v)$ $\quad$ ($q_{0,2} \in \epsilon$-Hülle($q_{f,1}$))

$\Rightarrow\ q_{f,2} \in \hat{\delta}(q_{0,1}, w)$ $\quad$ (Definition $\hat{\delta}$)

$\Rightarrow\ w \in L(A)$

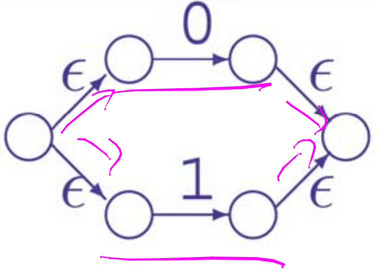
Argument ist umkehrbar, also $w \in L(A) \ \Rightarrow\ w \in L(E_1 \circ E_2)$

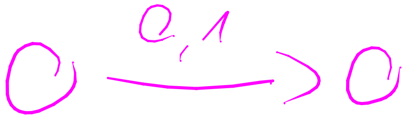
- **Sternbildung und Vereinigung ähnlich** ✓

Umwandlung regulärer Ausdrücke am Beispiel

Konstruiere endlichen Automaten für $(0+1)^*1\,(0+1)$

- **Teilautomat für** $(0+1)$

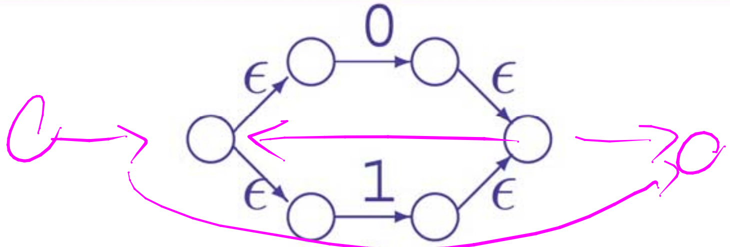




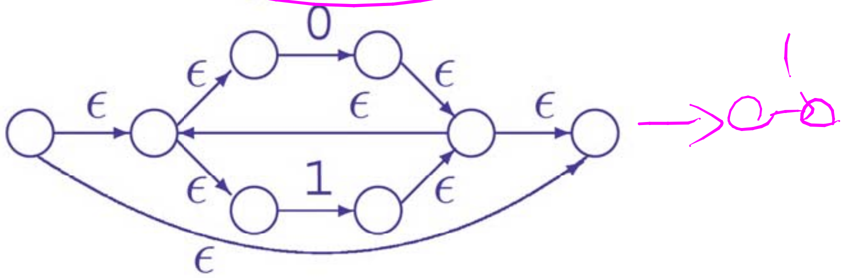
UMWANDLUNG REGULÄRER AUSDRÜCKE AM BEISPIEL

Konstruiere endlichen Automaten für `(0+1)*1(0+1)`

- **Teilautomat für** `(0+1)`



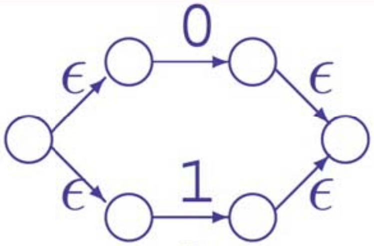
- **Teilautomat für** `(0+1)*`



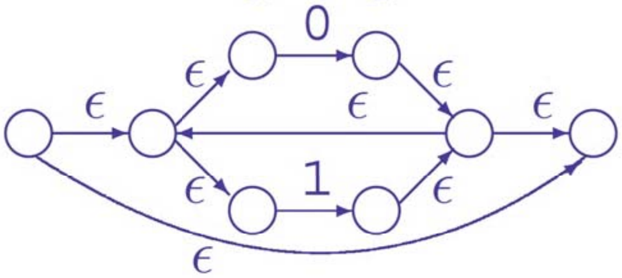
UMWANDLUNG REGULÄRER AUSDRÜCKE AM BEISPIEL

Konstruiere endlichen Automaten für `(0+1)*1(0+1)`

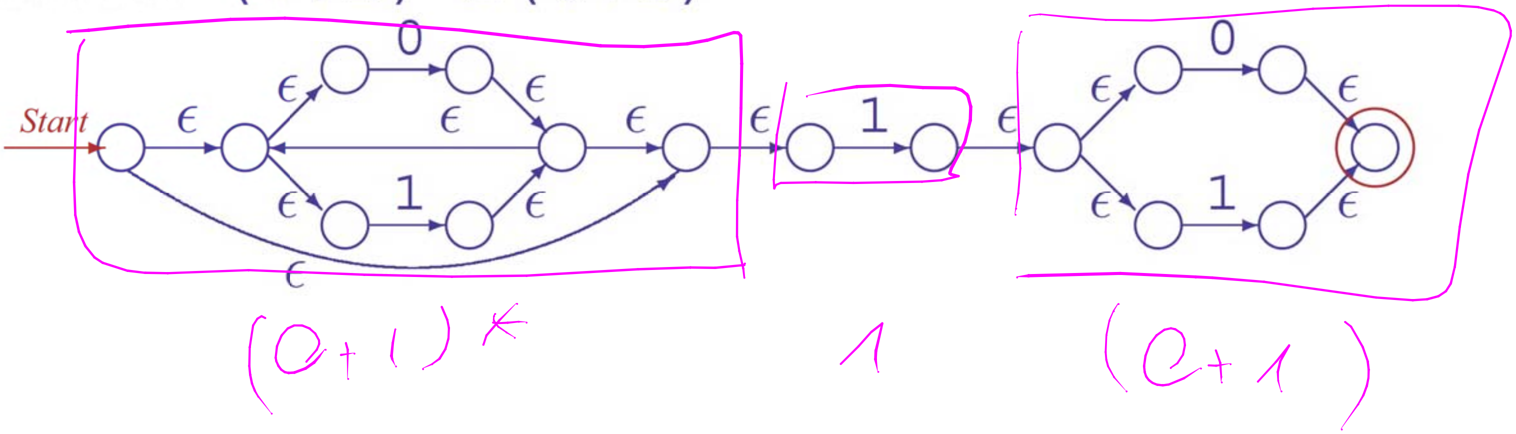
- **Teilautomat für** `(0+1)`



- **Teilautomat für** `(0+1)*`



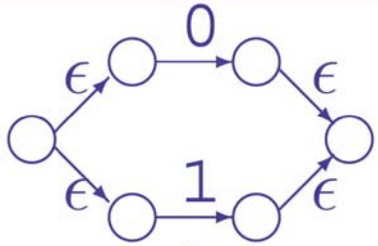
- **Automat für** `(0+1)*1(0+1)`



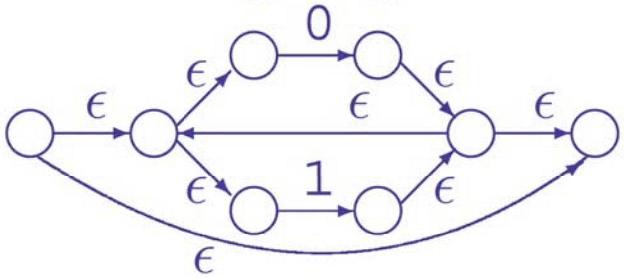
UMWANDLUNG REGULÄRER AUSDRÜCKE AM BEISPIEL

Konstruiere endlichen Automaten für `(0+1)*1(0+1)`

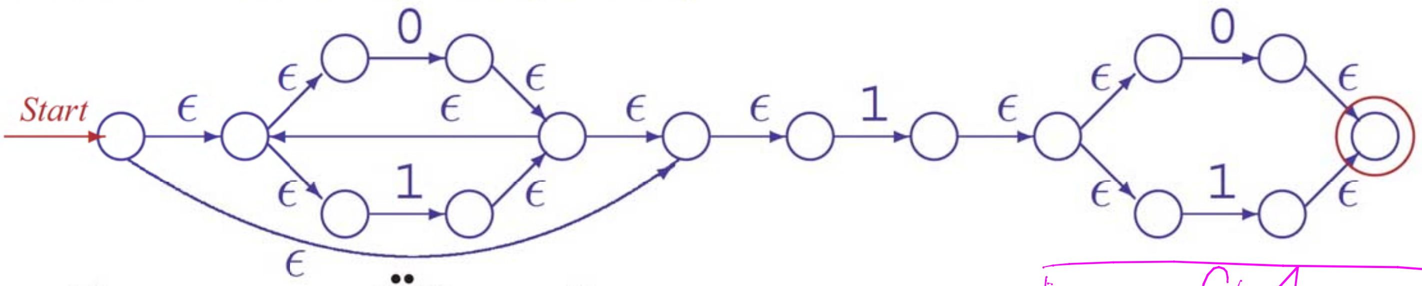
- **Teilautomat für** `(0+1)`



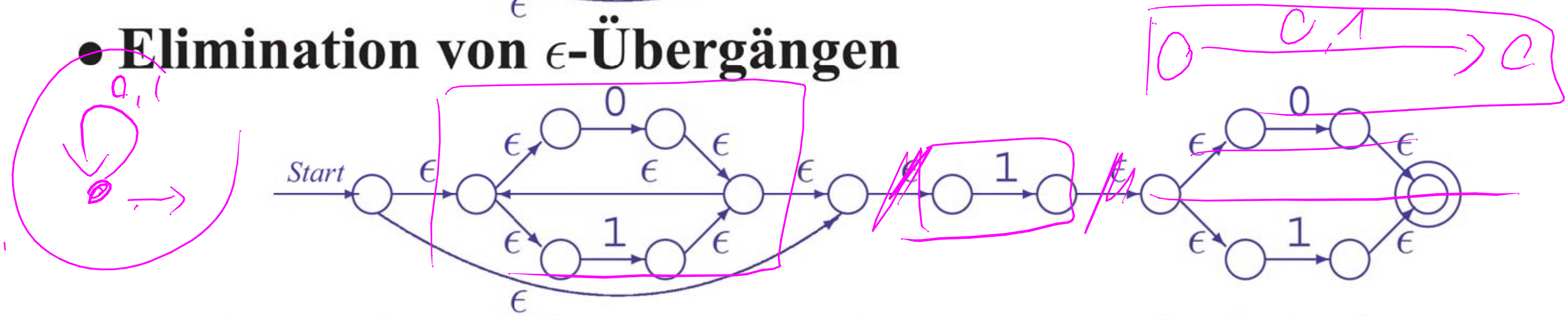
- **Teilautomat für** `(0+1)*`



- **Automat für** `(0+1)*1(0+1)`



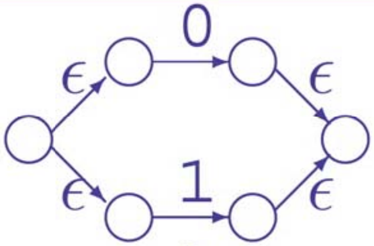
- **Elimination von** ϵ**-Übergängen**



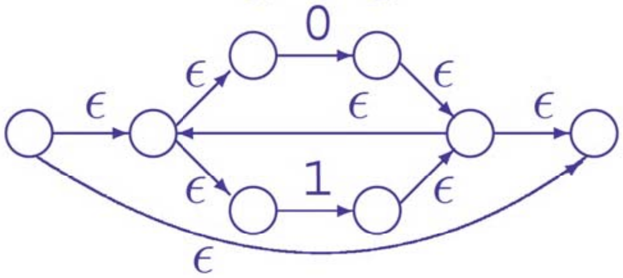
Umwandlung regulärer Ausdrücke am Beispiel

Konstruiere endlichen Automaten für $(0+1)^*1(0+1)$

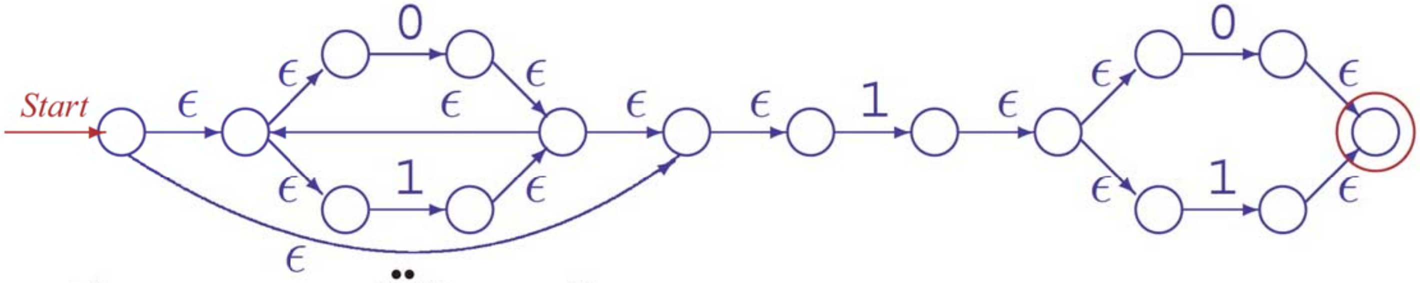
- **Teilautomat für** $(0+1)$



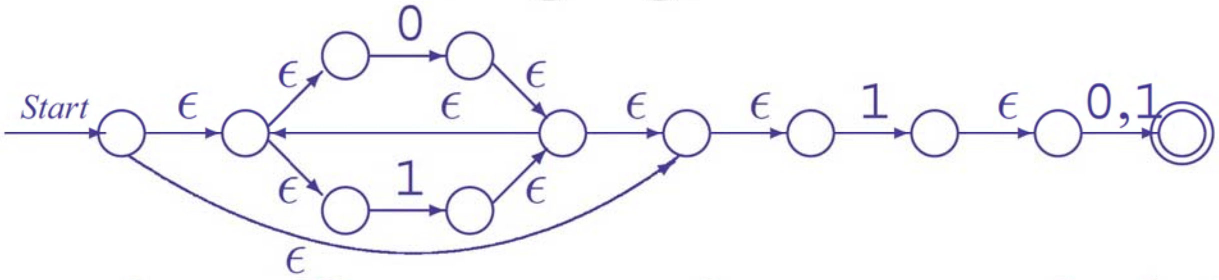
- **Teilautomat für** $(0+1)^*$



- **Automat für** $(0+1)^*1(0+1)$



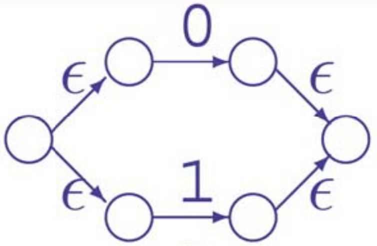
- **Elimination von ϵ-Übergängen**



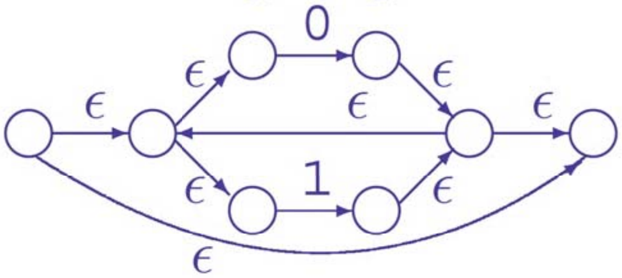
Umwandlung regulärer Ausdrücke am Beispiel

Konstruiere endlichen Automaten für $(0+1)^*1(0+1)$

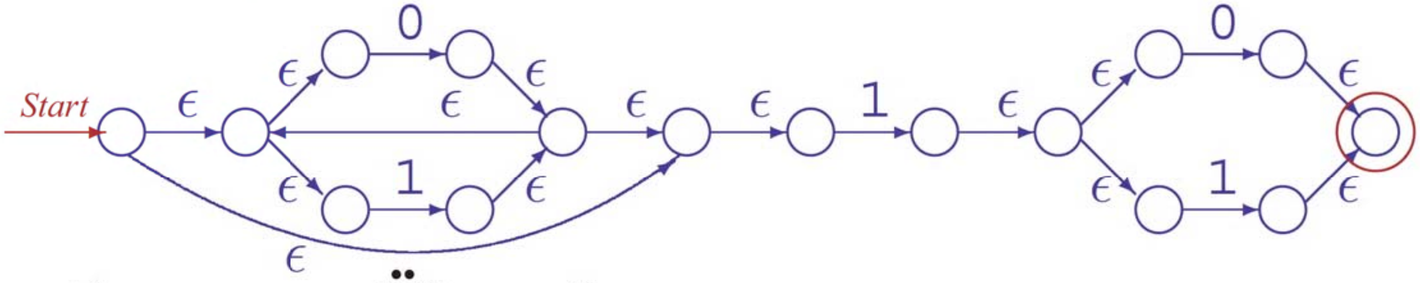
- **Teilautomat für** $(0+1)$



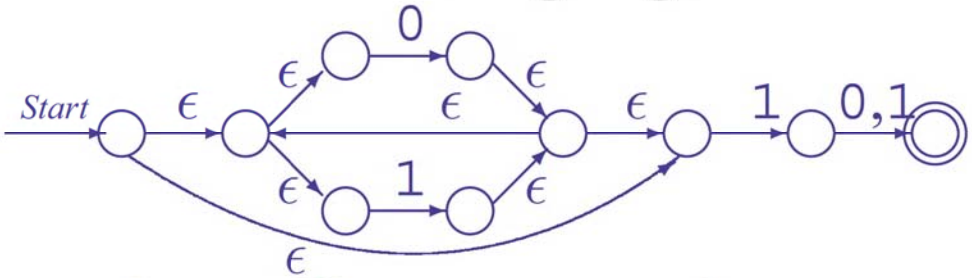
- **Teilautomat für** $(0+1)^*$



- **Automat für** $(0+1)^*1(0+1)$



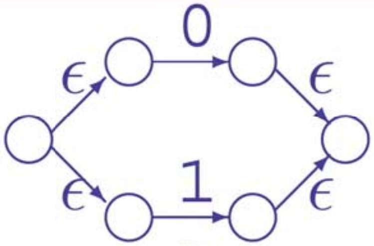
- **Elimination von ϵ-Übergängen**



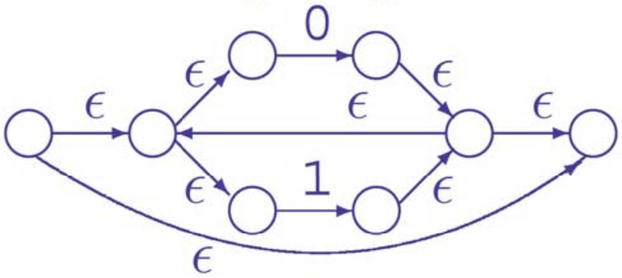
Umwandlung regulärer Ausdrücke am Beispiel

Konstruiere endlichen Automaten für `(0+1)*1(0+1)`

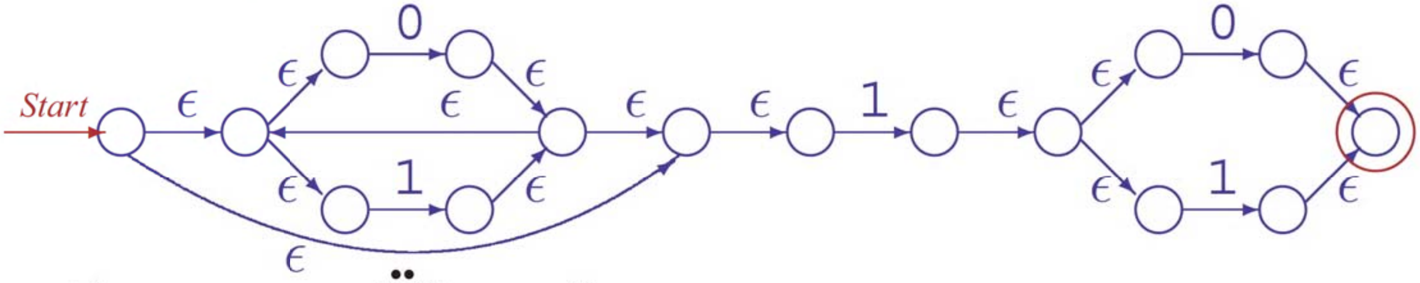
- **Teilautomat für** `(0+1)`



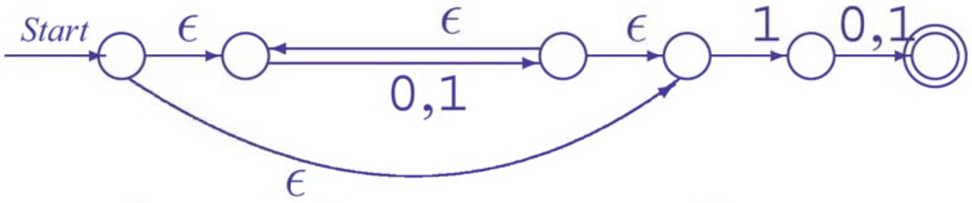
- **Teilautomat für** `(0+1)*`



- **Automat für** `(0+1)*1(0+1)`



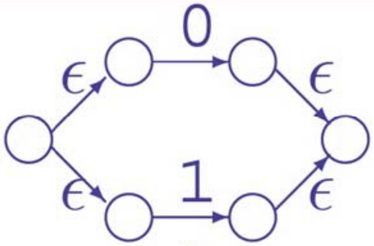
- **Elimination von ϵ-Übergängen**



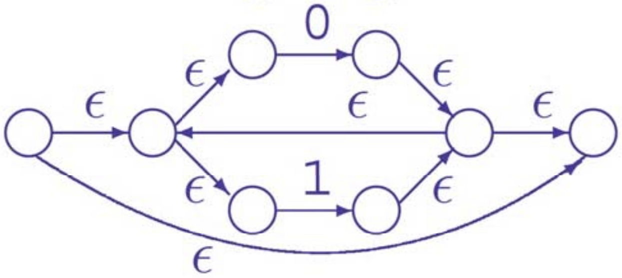
UMWANDLUNG REGULÄRER AUSDRÜCKE AM BEISPIEL

Konstruiere endlichen Automaten für `(0+1)*1(0+1)`

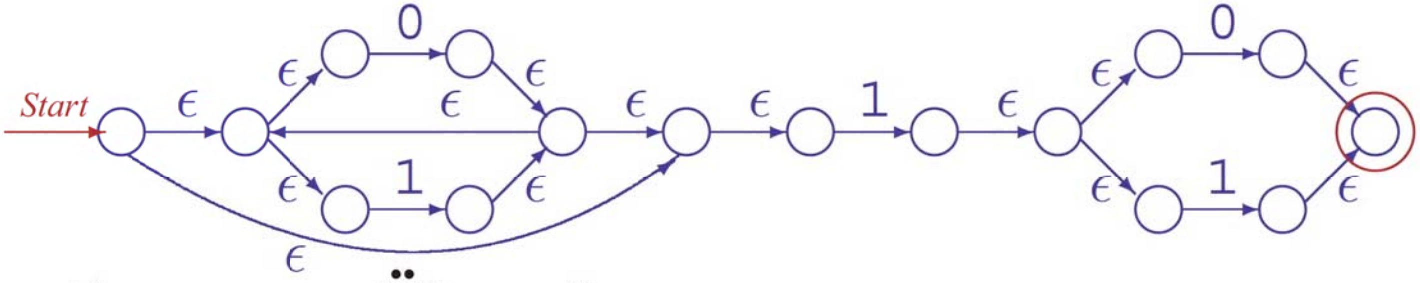
- **Teilautomat für** `(0+1)`



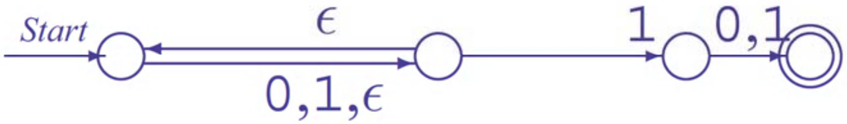
- **Teilautomat für** `(0+1)*`



- **Automat für** `(0+1)*1(0+1)`



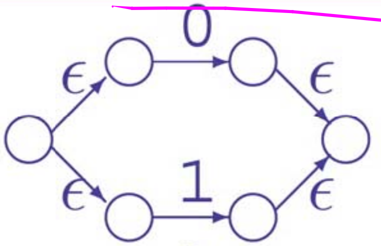
- **Elimination von ϵ-Übergängen**



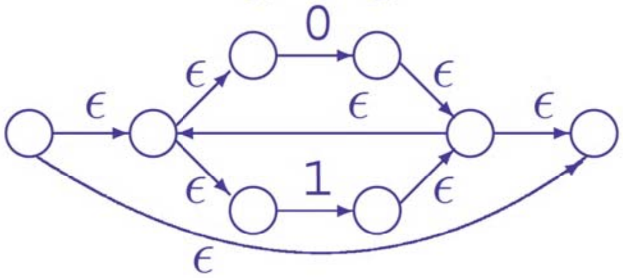
UMWANDLUNG REGULÄRER AUSDRÜCKE AM BEISPIEL

Konstruiere endlichen Automaten für `(0+1)*1(0+1)`

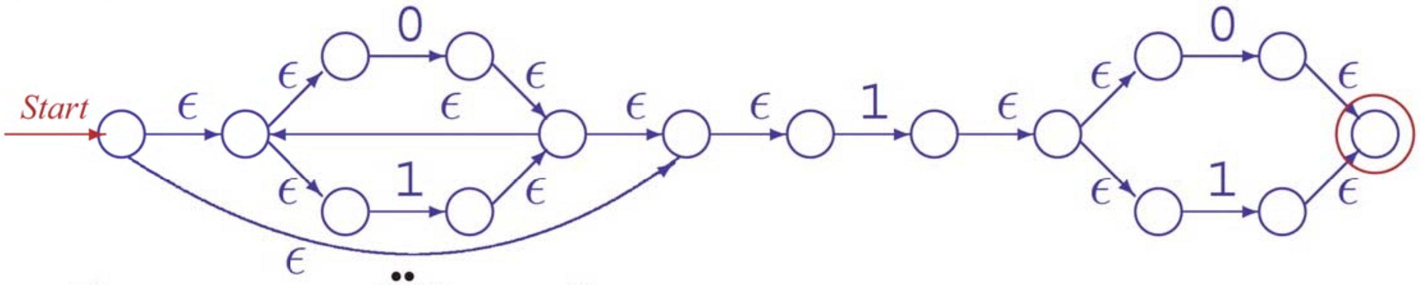
- **Teilautomat für** `(0+1)`



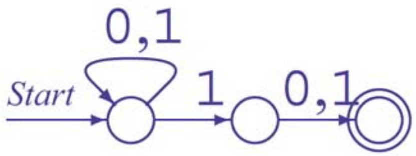
- **Teilautomat für** `(0+1)*`



- **Automat für** `(0+1)*1(0+1)`



- **Elimination von ϵ-Übergängen**



UMWANDLUNG VON (ϵ-)NEAS IN REGULÄRE AUSDRÜCKE

- **Ursprünglich: Pfadanalyse im Übergangsdiagramm**
  - Spezialisierung eines allgemeinen Verfahrens für Pfadanalyse in Graphen
  - Definiere reguläre Ausdrücke für Pfade durch Automaten
  - Berechnung Ausdrücke iterativ und kombiniere alle relevanten Ausdrücke
  - Kompliziertes und aufwendiges Verfahren

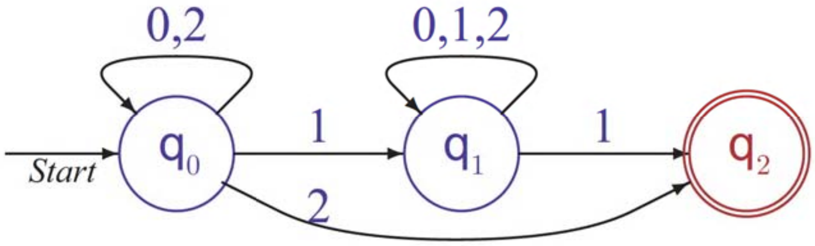
Mehr dazu im Anhang

$E G^* F$

- **Effizienterer Zugang: Elimination von Zuständen**
  - Beschreibe Übergänge $q_i \xrightarrow{a \in \Sigma} q_j$ durch reguläre Ausdrücke
  - Beginne mit regulären Ausdrücken für direkte Übergänge
  - Entferne einzelne Zustände und beschreibe die entstehenden Ausdrücke
  - Liefert Ausdrücke für Übergänge zwischen Start- und Endzuständen
- **Technisches Hilfsmittel: verallgemeinerte NEAs (VNEAs)**
  - NEA, dessen Überführungsfunktion δ auf regulären Ausdrücken arbeitet
  - A akzeptiert w, wenn es einen Pfad $w = v_1..v_m$ von q_0 zu einem $q \in F$ gibt und alle v_i in der Sprache des entsprechenden regulären Ausdrucks liegen
  - Konsistente Formalisierung mühsam und ohne Erkenntnisgewinn

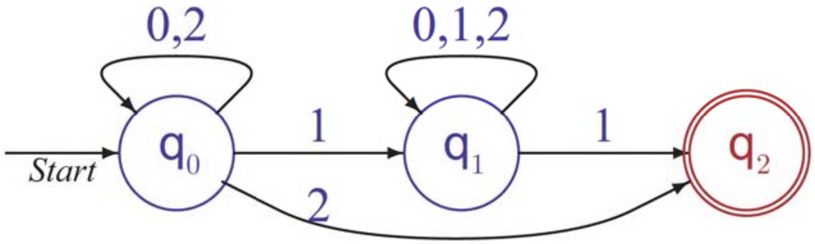
ZUSTANDSELIMINATION IN VNEAS

- **Ursprünglicher NEA**

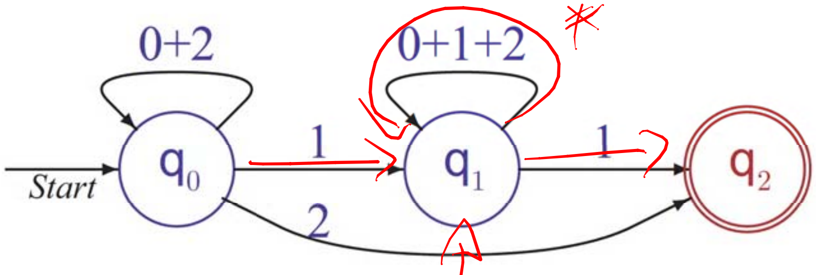


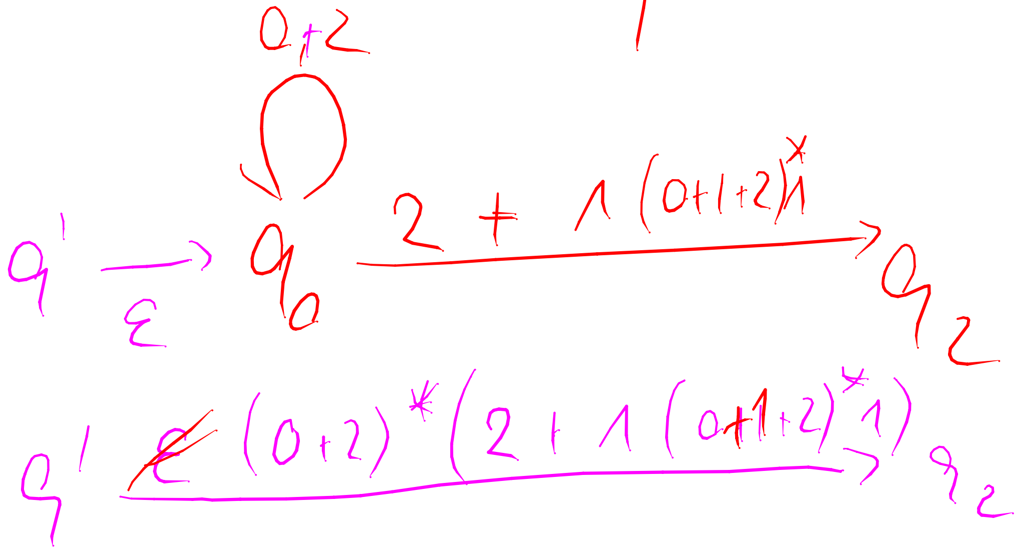
ZUSTANDSELIMINATION IN VNEAS

- **Ursprünglicher NEA**



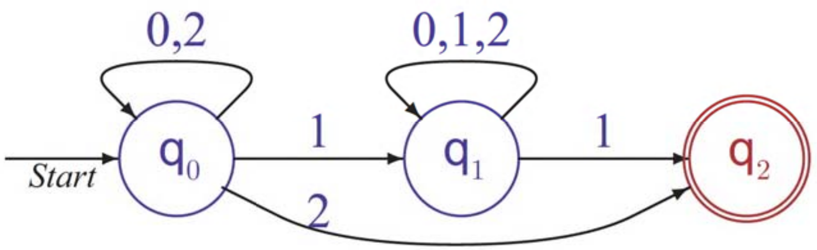
- **Zugehöriger VNEA**



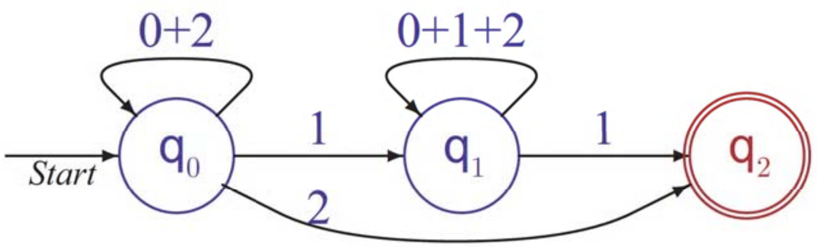


ZUSTANDSELIMINATION IN VNEAS

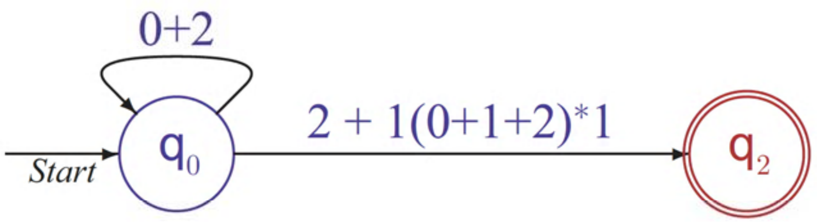
- **Ursprünglicher NEA**



- **Zugehöriger VNEA**

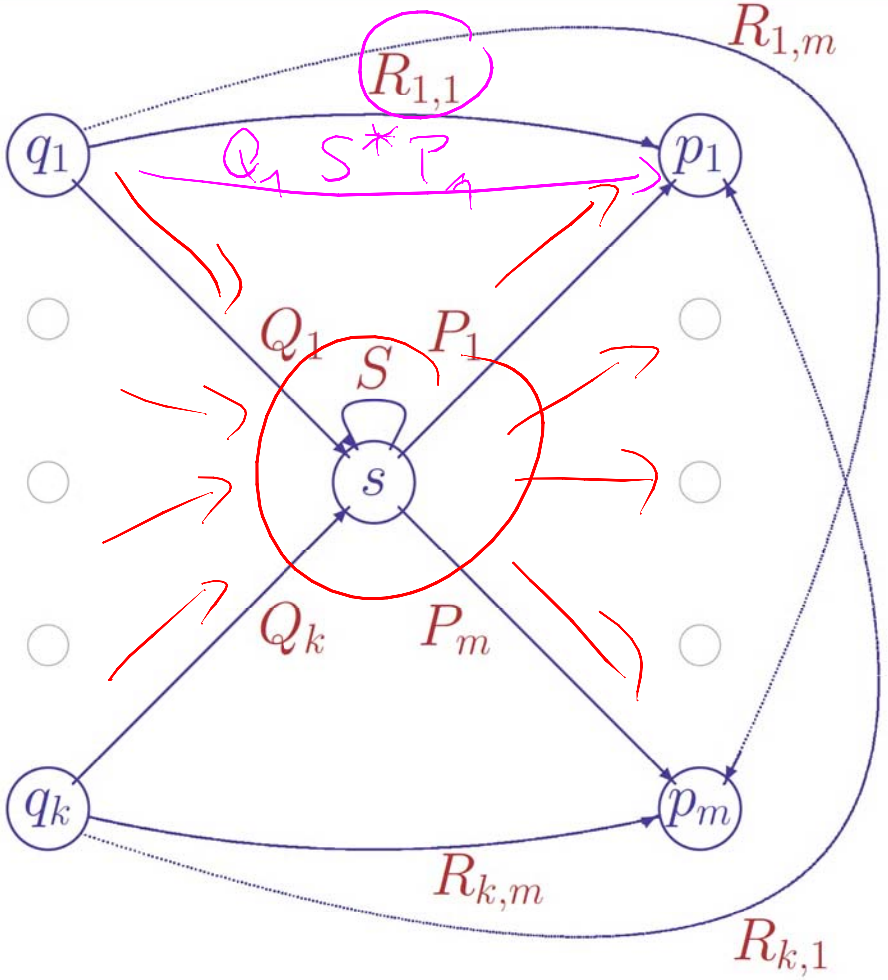


- **Nach Elimination von q_1**



  - Ausdruck für Übergang von q_0 nach q_2 ergibt sich aus Übergang q_0 nach q_1, Schleife bei q_1, Übergang q_1 nach q_2 und existierendem Ausdruck für direkten Übergang von q_0 nach q_2

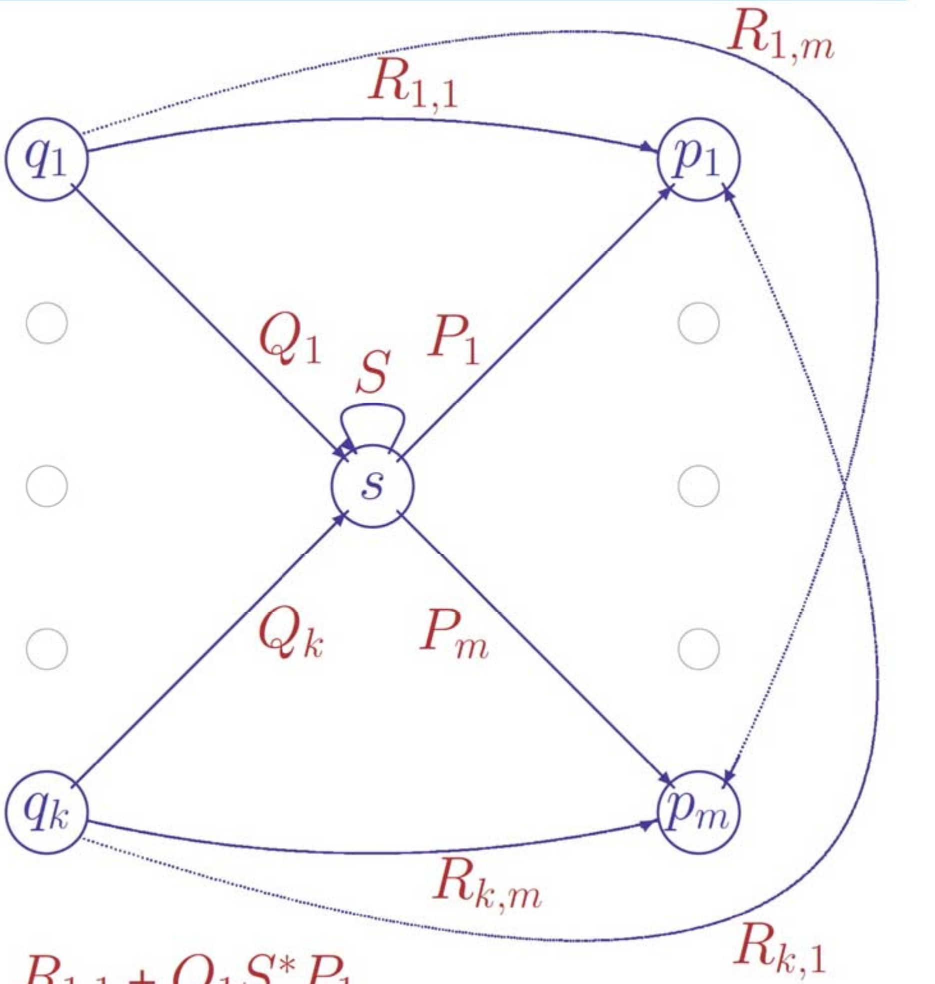
ALLGEMEINE ZUSTANDSELIMINATION IN VNEAS



Eliminiere Zustand s
mit Vorgängern $q_1, .., q_k$
und Nachfolgern $p_1, .., p_m$

q_i , p_j können gleich sein!

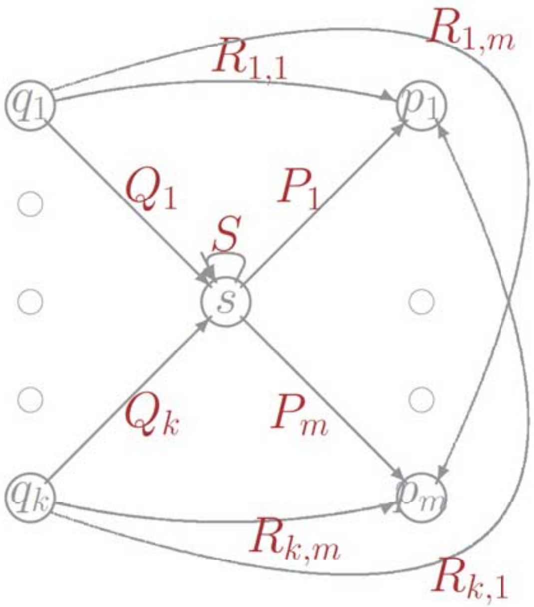
ALLGEMEINE ZUSTANDSELIMINATION IN VNEAS

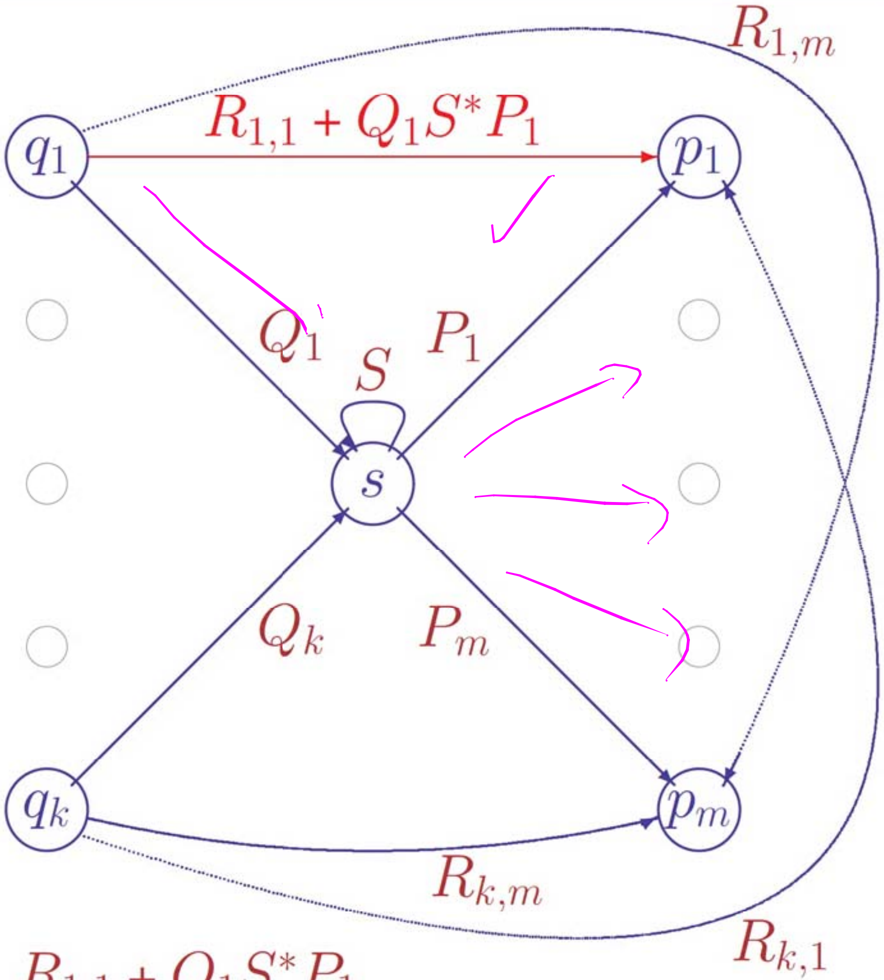


Eliminiere Zustand s
mit Vorgängern $q_1, .., q_k$
und Nachfolgern $p_1, .., p_m$

– Eliminiere Pfad von q_1 nach p_1 über s: $\underline{R_{1,1}} + \underline{Q_1}\underline{S^*}\underline{P_1}$

ALLGEMEINE ZUSTANDSELIMINATION IN VNEAS

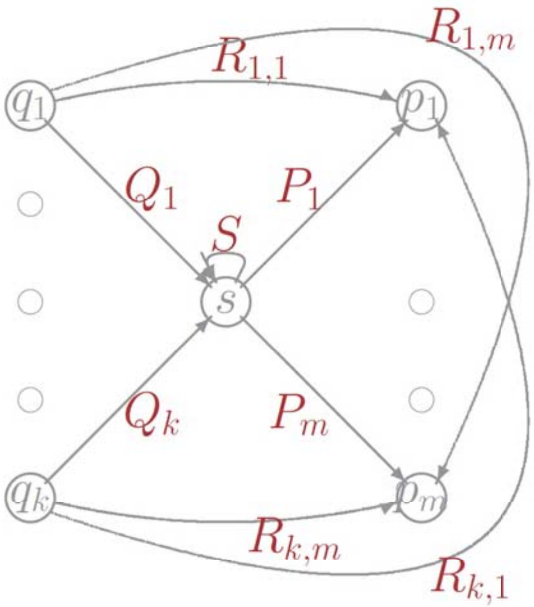


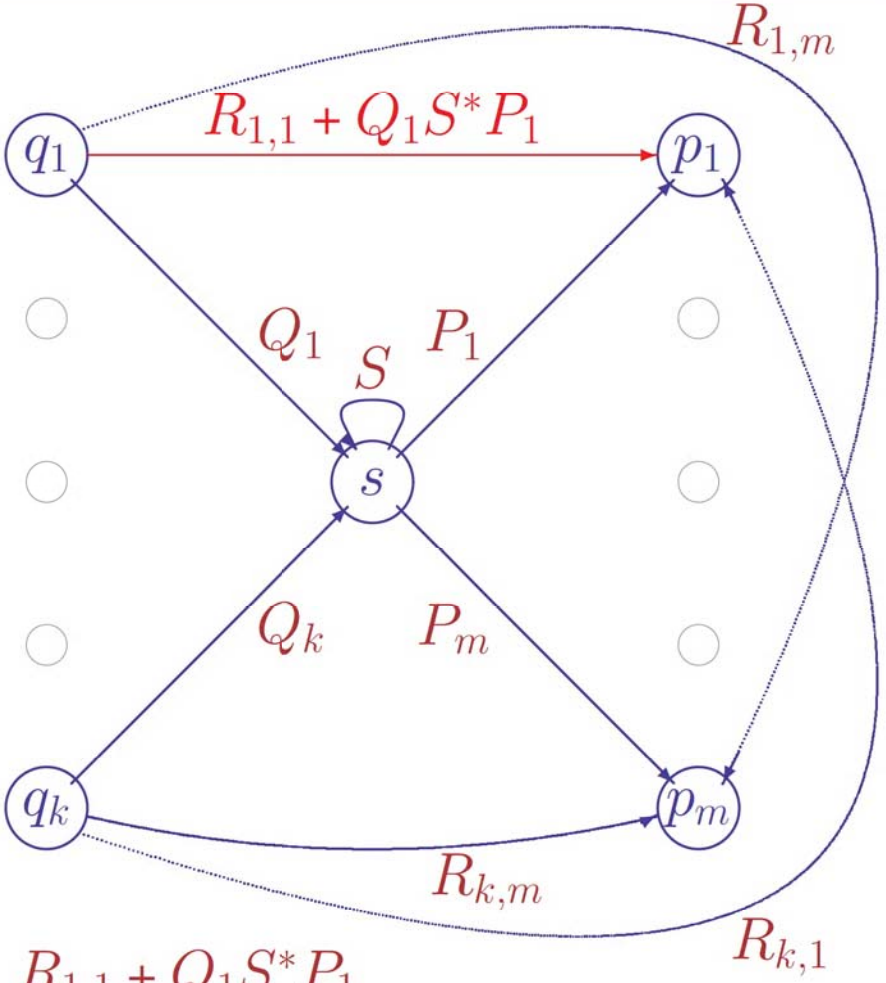


Eliminiere Zustand s
mit Vorgängern $q_1, .., q_k$
und Nachfolgern $p_1, .., p_m$

– Eliminiere Pfad von q_1 nach p_1 über s: $R_{1,1} + \underline{Q_1 S}^* P_1$

Allgemeine Zustandselimination in VNEAs





Eliminiere Zustand s

mit Vorgängern $q_1, .., q_k$

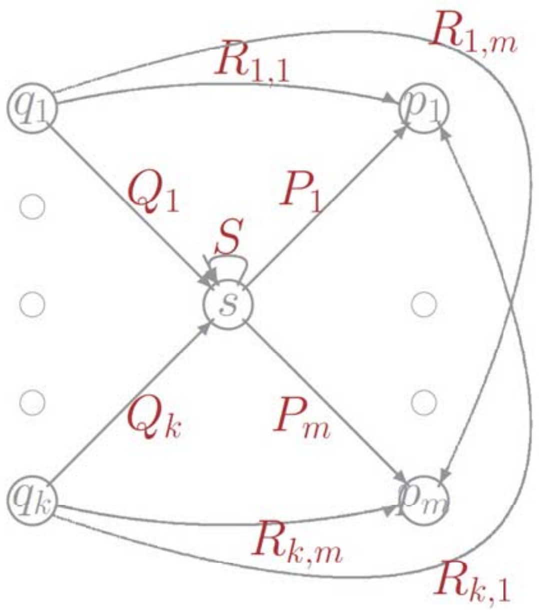
und Nachfolgern $p_1, .., p_m$

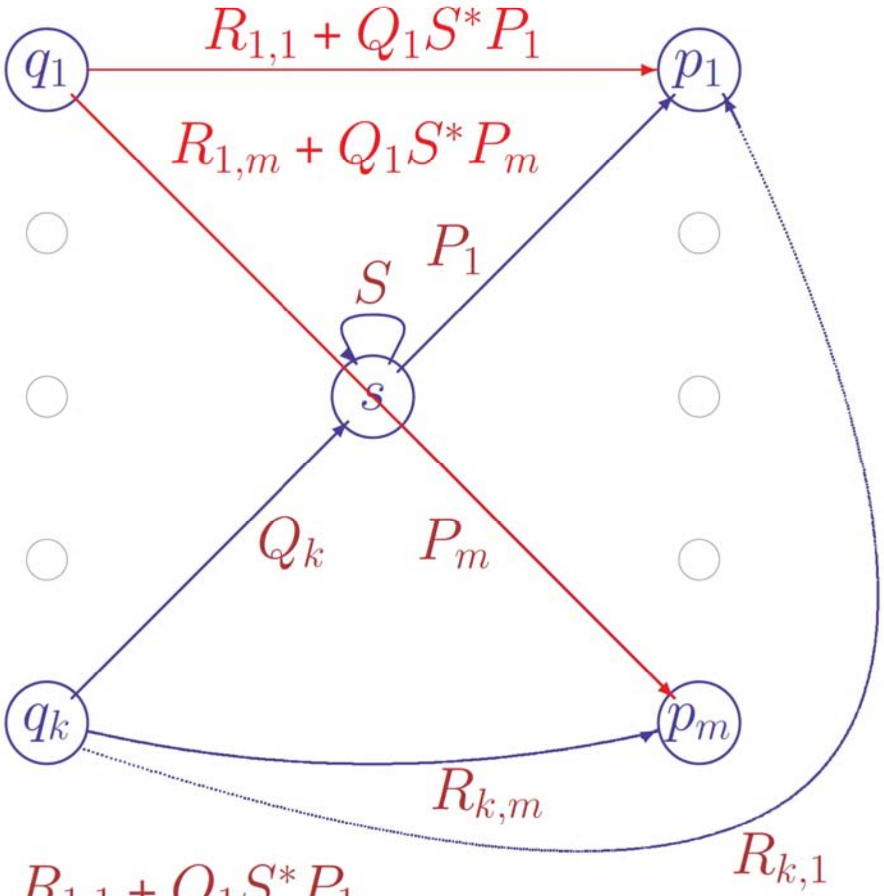
– Eliminiere Pfad von q_1 nach p_1 über s: $R_{1,1} + Q_1S^*P_1$

⋮

– Eliminiere Pfad von q_1 nach p_m über s: $R_{1,m} + Q_1S^*P_m$

ALLGEMEINE ZUSTANDSELIMINATION IN VNEAS





Eliminiere Zustand s

mit Vorgängern $q_1, .., q_k$

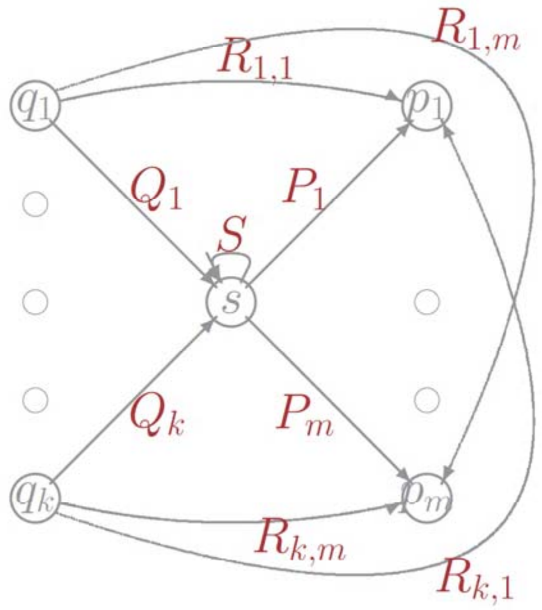
und Nachfolgern $p_1, .., p_m$

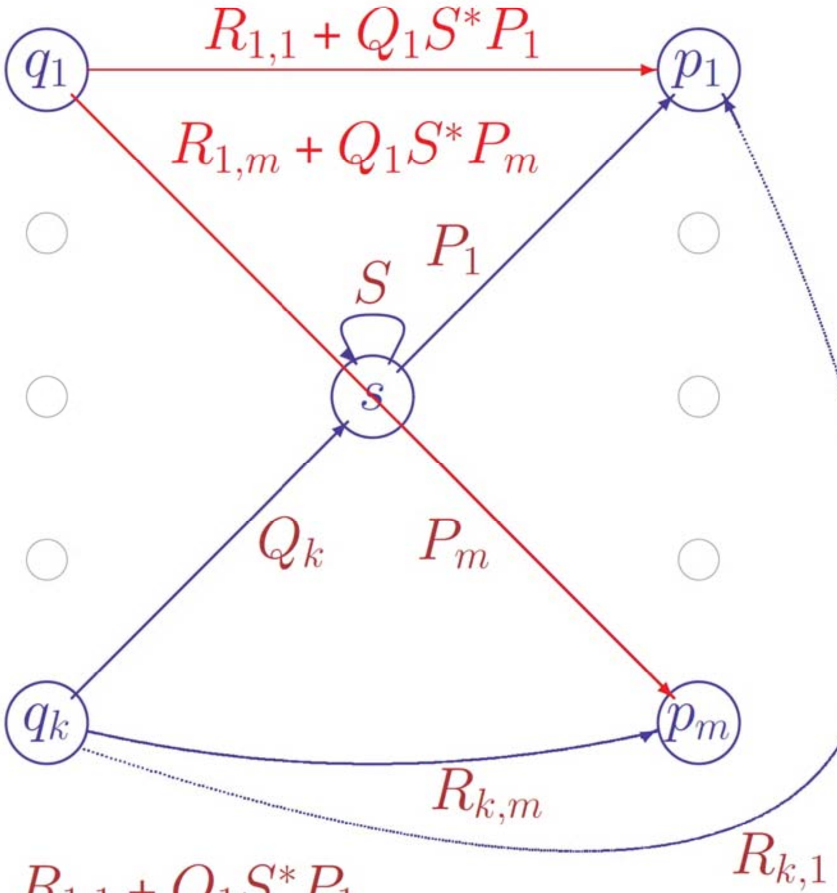
– Eliminiere Pfad von q_1 nach p_1 über s: $R_{1,1} + Q_1S^*P_1$

⋮

– Eliminiere Pfad von q_1 nach p_m über s: $R_{1,m} + Q_1S^*P_m$

ALLGEMEINE ZUSTANDSELIMINATION IN VNEAS





Eliminiere Zustand s

mit Vorgängern $q_1, .., q_k$

und Nachfolgern $p_1, .., p_m$

– Eliminiere Pfad von q_1 nach p_1 über s: $R_{1,1} + Q_1S^*P_1$

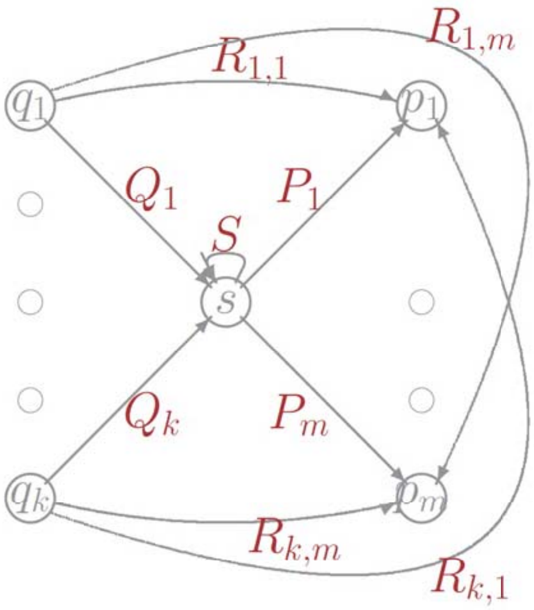
⋮

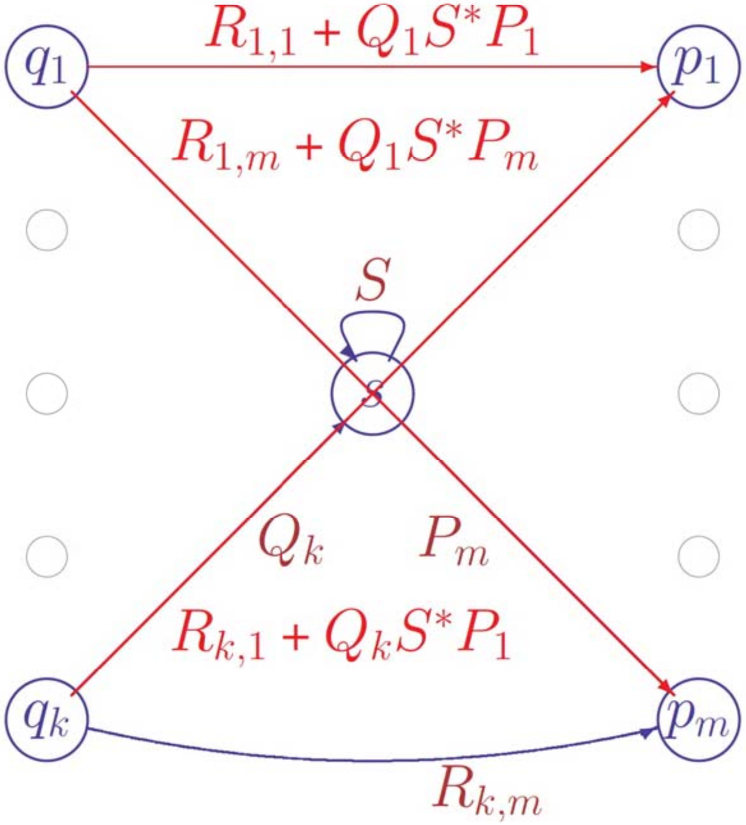
– Eliminiere Pfad von q_1 nach p_m über s: $R_{1,m} + Q_1S^*P_m$

⋮

– Eliminiere Pfad von q_k nach p_1 über s: $R_{k,1} + Q_kS^*P_1$

ALLGEMEINE ZUSTANDSELIMINATION IN VNEAs





Eliminiere Zustand s
mit Vorgängern $q_1, .., q_k$
und Nachfolgern $p_1, .., p_m$

– Eliminiere Pfad von q_1 nach p_1 über s: $R_{1,1} + Q_1 S^* P_1$

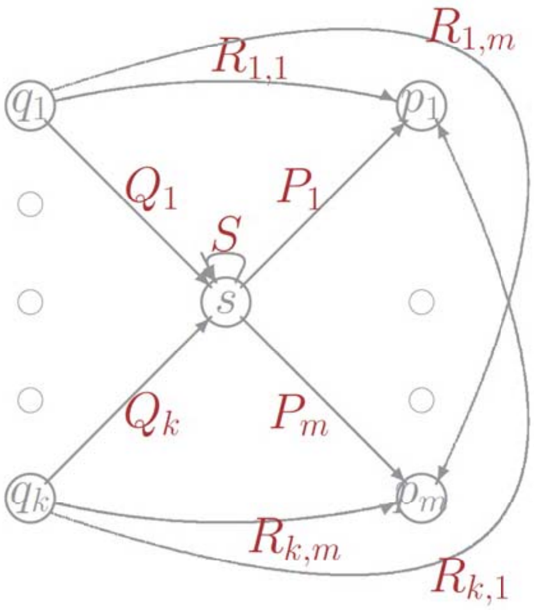
⋮

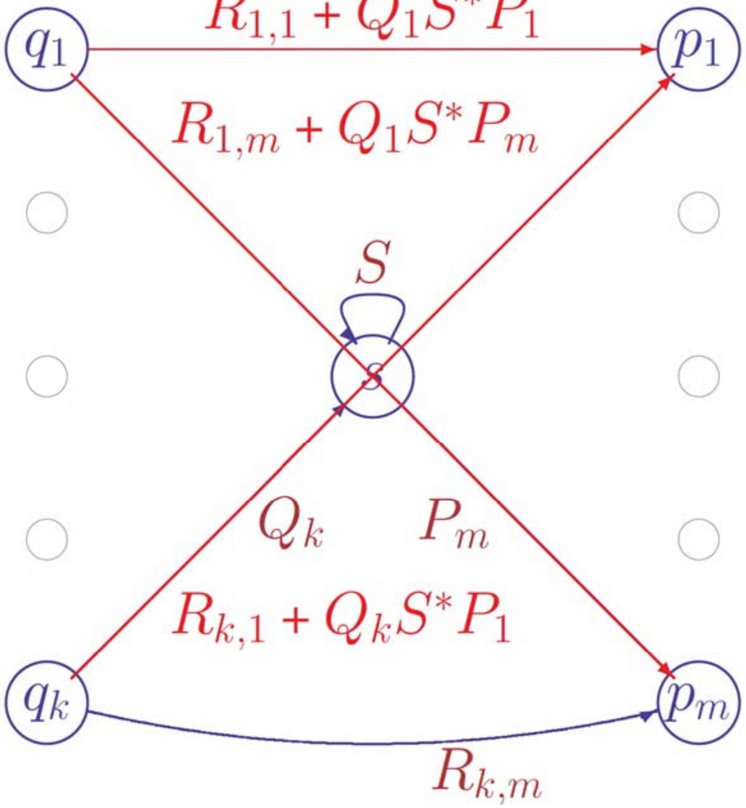
– Eliminiere Pfad von q_1 nach p_m über s: $R_{1,m} + Q_1 S^* P_m$

⋮

– Eliminiere Pfad von q_k nach p_1 über s: $R_{k,1} + Q_k S^* P_1$

ALLGEMEINE ZUSTANDSELIMINATION IN VNEAS





Eliminiere Zustand s
mit Vorgängern $q_1, .., q_k$
und Nachfolgern $p_1, .., p_m$

- Eliminiere Pfad von q_1 nach p_1 über s: $R_{1,1} + Q_1S^*P_1$

⋮

- Eliminiere Pfad von q_1 nach p_m über s: $R_{1,m} + Q_1S^*P_m$

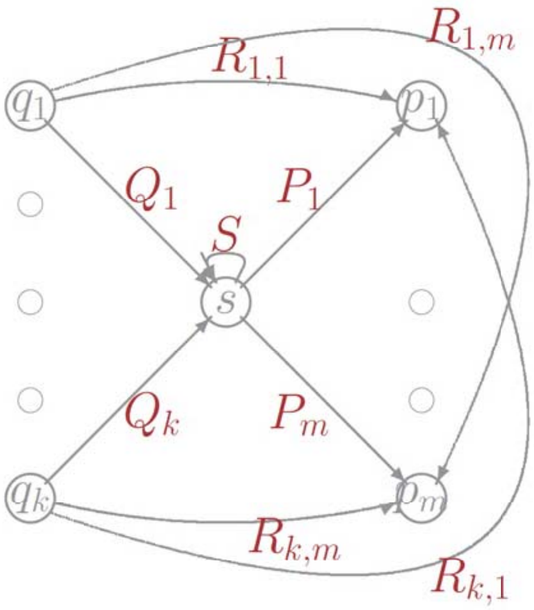
⋮

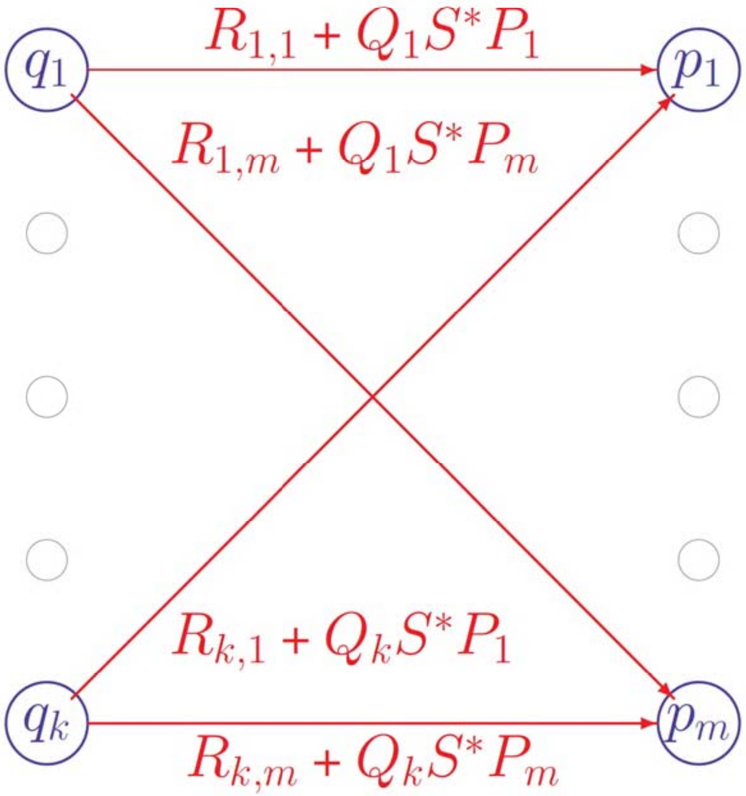
- Eliminiere Pfad von q_k nach p_1 über s: $R_{k,1} + Q_kS^*P_1$

⋮

- Eliminiere Pfad von q_k nach p_m über s: $R_{k,m} + Q_kS^*P_m$

ALLGEMEINE ZUSTANDSELIMINATION IN VNEAS





Eliminiere Zustand s
mit Vorgängern $q_1, .., q_k$
und Nachfolgern $p_1, .., p_m$

– Eliminiere Pfad von q_1 nach p_1 über s: $R_{1,1} + Q_1 S^* P_1$

⋮

– Eliminiere Pfad von q_1 nach p_m über s: $R_{1,m} + Q_1 S^* P_m$

⋮

– Eliminiere Pfad von q_k nach p_1 über s: $R_{k,1} + Q_k S^* P_1$

⋮

– Eliminiere Pfad von q_k nach p_m über s: $R_{k,m} + Q_k S^* P_m$

UMWANDLUNG DURCH ZUSTANDSELIMINATION

1. **Ergänze neuen Anfangszustand p_0 und Endzustand p_f**
   - ϵ-Übergang von neuem zu altem Anfangszustand
   - ϵ-Übergang von allen alten Endzuständen zum neuen Endzustand

   ↦ Anfangs- und Endzustand haben keine ein- bzw. ausgehenden Kanten

UMWANDLUNG DURCH ZUSTANDSELIMINATION

1. **Ergänze neuen Anfangszustand p_0 und Endzustand p_f**
   - ϵ-Übergang von neuem zu altem Anfangszustand
   - ϵ-Übergang von allen alten Endzuständen zum neuen Endzustand

   $\mapsto$ Anfangs- und Endzustand haben keine ein- bzw. ausgehenden Kanten

2. **Transformiere endlichen Automaten in VNEA**
   - Ersetze Beschriftungen mit Symbolen $a \in \Sigma$ durch reguläre Ausdrücke

UMWANDLUNG DURCH ZUSTANDSELIMINATION

1. **Ergänze neuen Anfangszustand p_0 und Endzustand p_f**
   - ϵ-Übergang von neuem zu altem Anfangszustand
   - ϵ-Übergang von allen alten Endzuständen zum neuen Endzustand

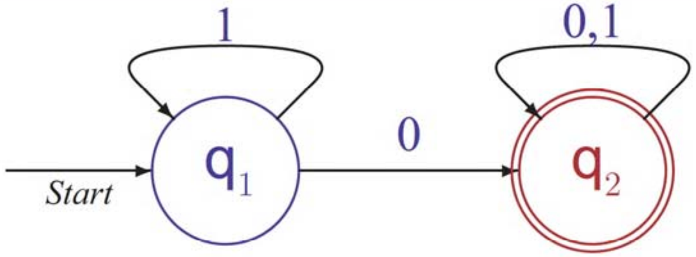
   $\mapsto$ Anfangs- und Endzustand haben keine ein- bzw. ausgehenden Kanten
2. **Transformiere endlichen Automaten in VNEA**
   - Ersetze Beschriftungen mit Symbolen $a \in \Sigma$ durch reguläre Ausdrücke
3. **Eliminiere alle Zustände außer p_0 und p_f**
   - Iterative Anwendung des Eliminationsverfahrens
   - Vereinfache entstehende Ausdrücke nach jedem Schritt

UMWANDLUNG DURCH ZUSTANDSELIMINATION

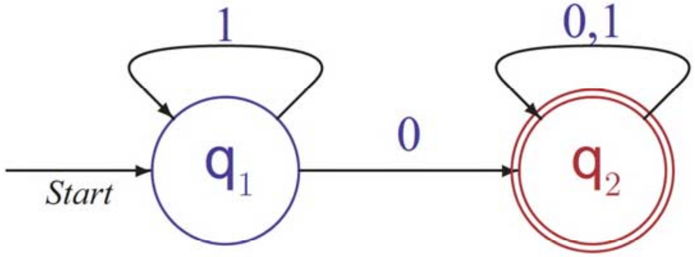
1. **Ergänze neuen Anfangszustand p_0 und Endzustand p_f**
   - ϵ-Übergang von neuem zu altem Anfangszustand
   - ϵ-Übergang von allen alten Endzuständen zum neuen Endzustand

   $\mapsto$ Anfangs- und Endzustand haben keine ein- bzw. ausgehenden Kanten
2. **Transformiere endlichen Automaten in VNEA**
   - Ersetze Beschriftungen mit Symbolen $a \in \Sigma$ durch reguläre Ausdrücke
3. **Eliminiere alle Zustände außer p_0 und p_f**
   - Iterative Anwendung des Eliminationsverfahrens
   - Vereinfache entstehende Ausdrücke nach jedem Schritt
4. **Extrahierte regulären Ausdruck aus VNEA**
   - Zielausdruck ist Beschriftung der Kante von p_0 nach p_f

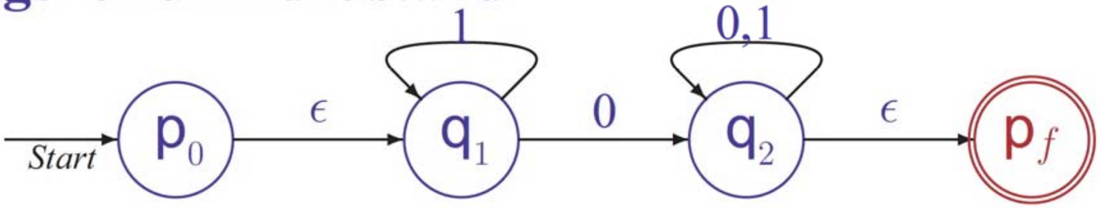
UMWANDLUNG NEA → RA: EINFACHES BEISPIEL



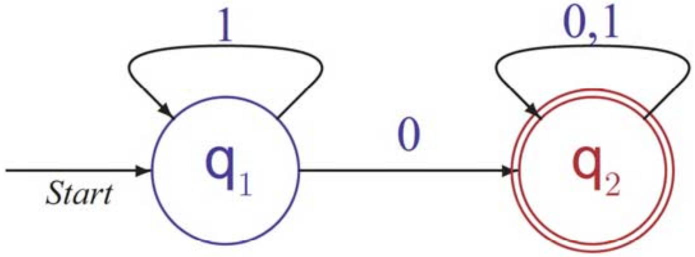
UMWANDLUNG NEA → RA: EINFACHES BEISPIEL



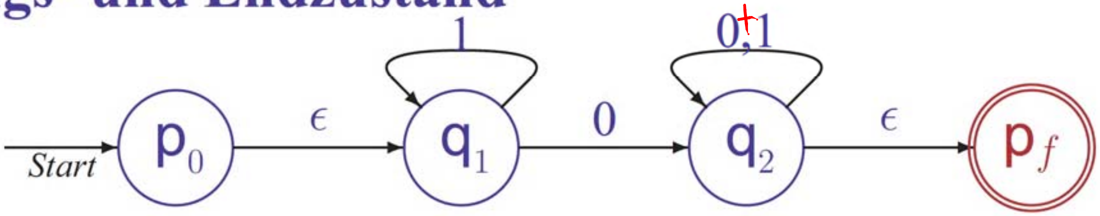
1. Ergänze neuen Anfangs- und Endzustand



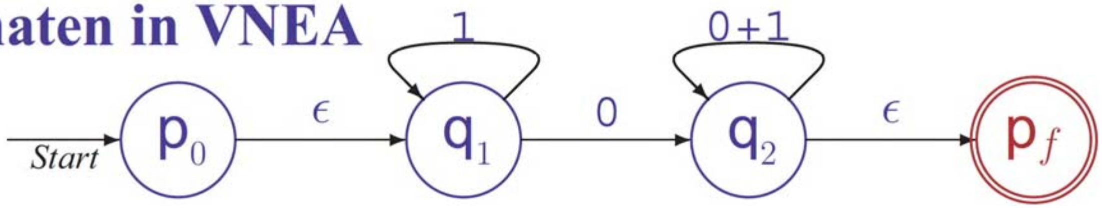
Umwandlung NEA → RA: einfaches Beispiel



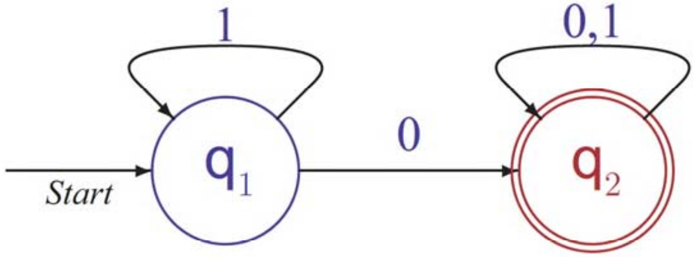
1. Ergänze neuen Anfangs- und Endzustand



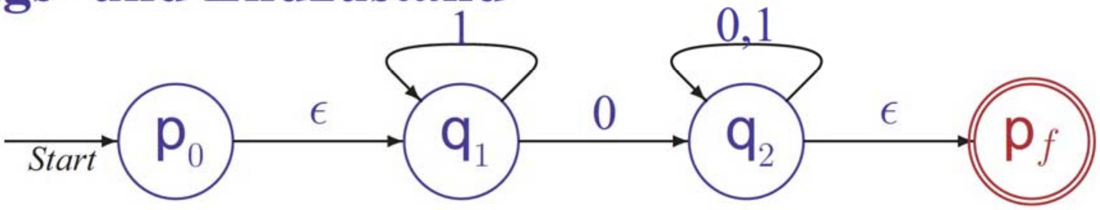
2. Transformiere Automaten in VNEA



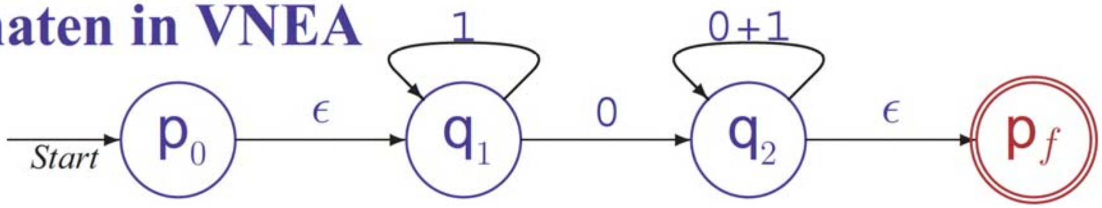
Umwandlung NEA → RA: einfaches Beispiel



1. Ergänze neuen Anfangs- und Endzustand

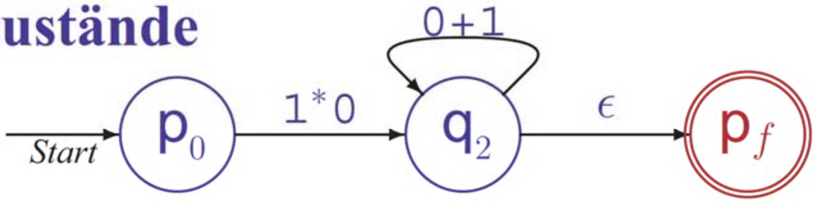


2. Transformiere Automaten in VNEA

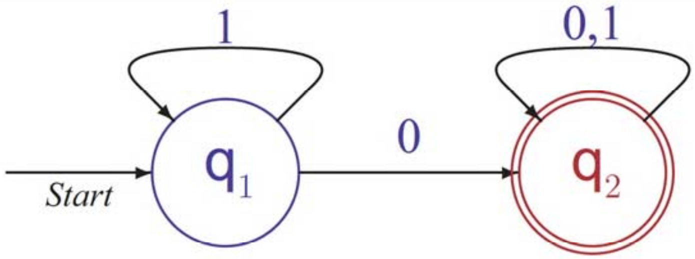


3. Eliminiere Zwischenzustände

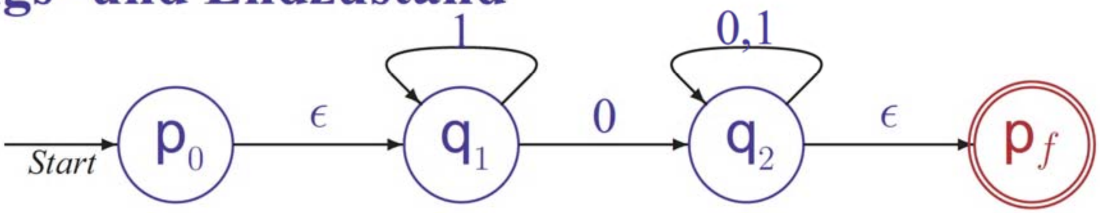
- Eliminiere q_1



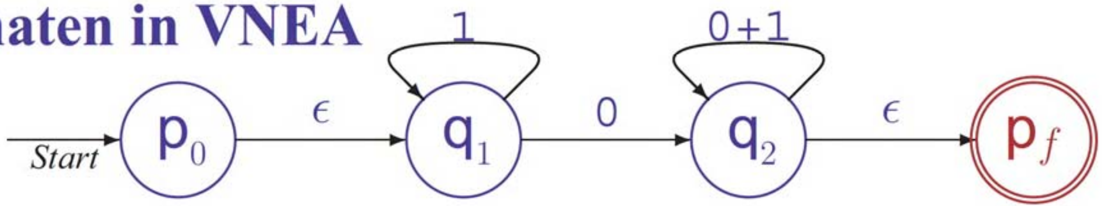
UMWANDLUNG NEA → RA: EINFACHES BEISPIEL



1. Ergänze neuen Anfangs- und Endzustand

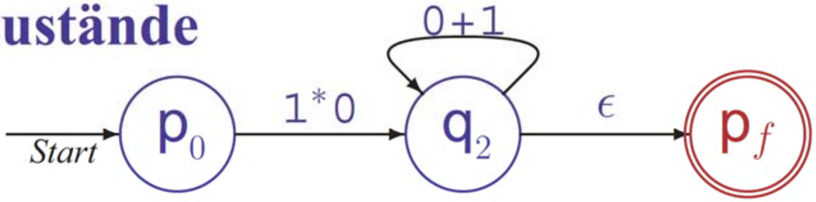


2. Transformiere Automaten in VNEA

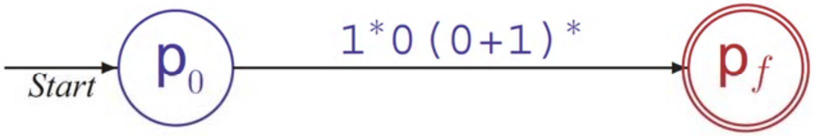


3. Eliminiere Zwischenzustände

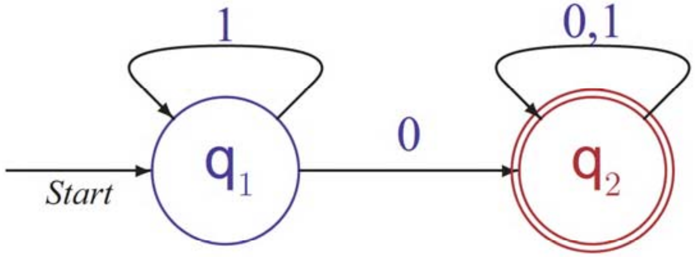
– Eliminiere q_1



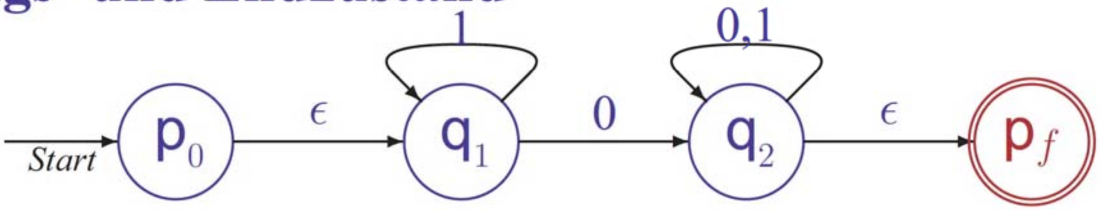
– Eliminiere q_2



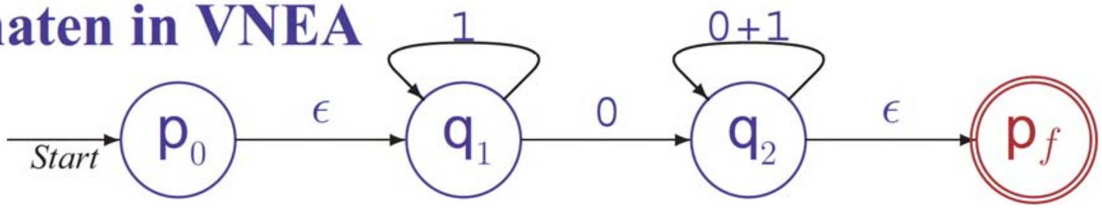
UMWANDLUNG NEA → RA: EINFACHES BEISPIEL



1. Ergänze neuen Anfangs- und Endzustand

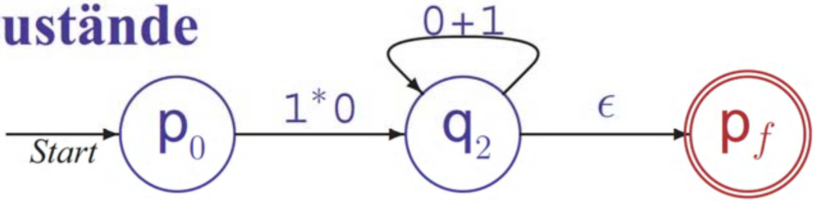


2. Transformiere Automaten in VNEA

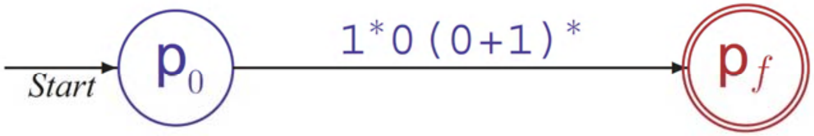


3. Eliminiere Zwischenzustände

- Eliminiere q_1



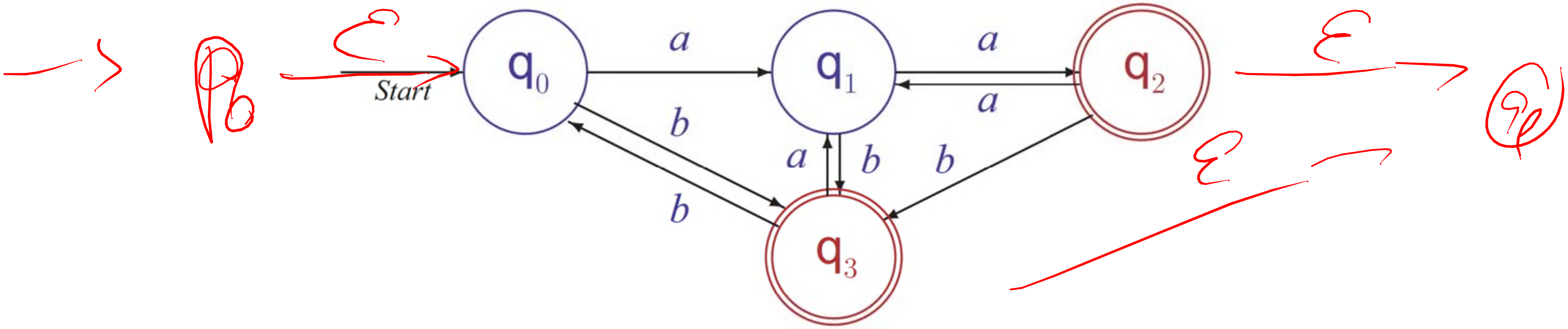
- Eliminiere q_2



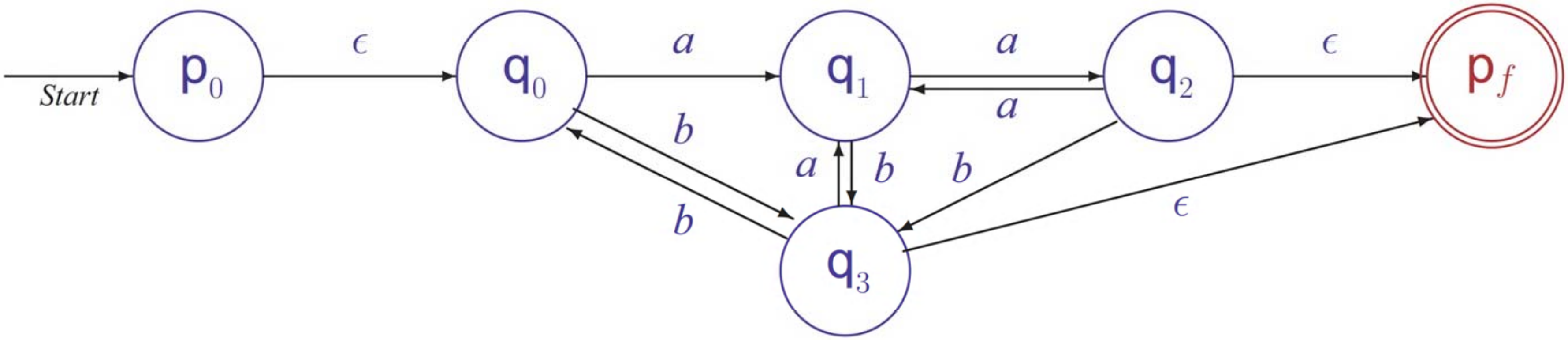
4. Extrahierter Ausdruck ist $1^*0(0+1)^*$

Umwandlung mit Pfadanalyseverfahren würde 12 aufwendige Schritte erfordern

UMWANDLUNG NEA → RA AM BEISPIEL

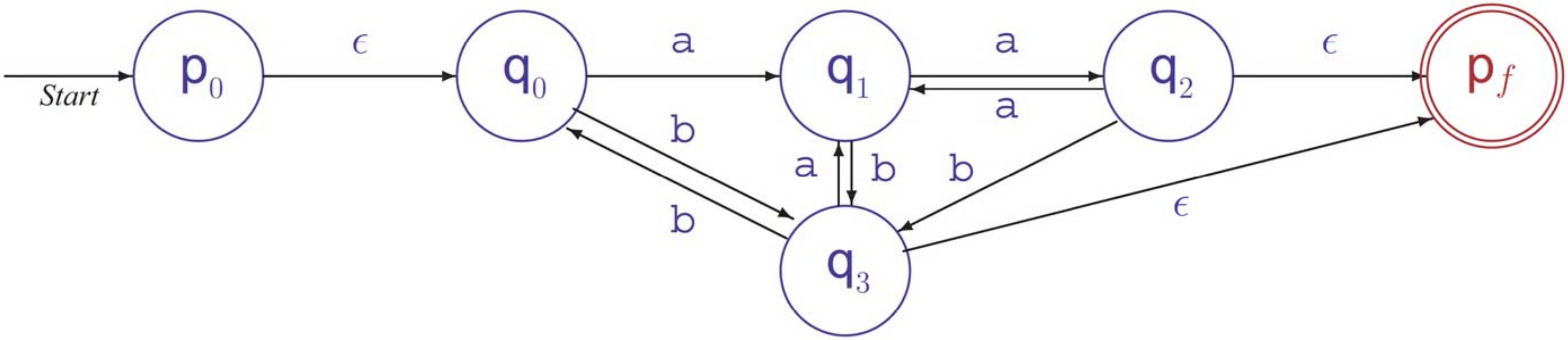


UMWANDLUNG NEA → RA AM BEISPIEL



1. **Ergänze neuen Anfangs- und Endzustand**

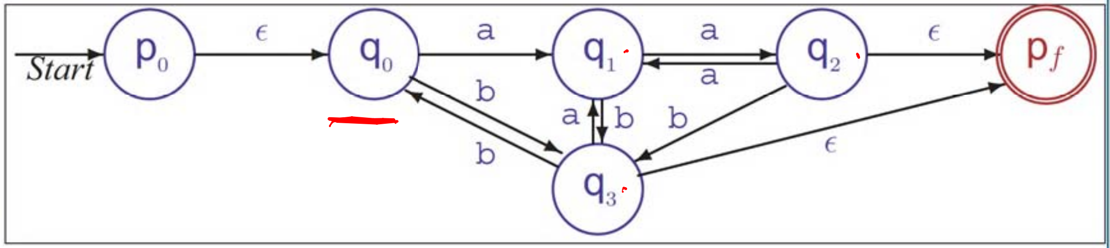
UMWANDLUNG NEA → RA AM BEISPIEL



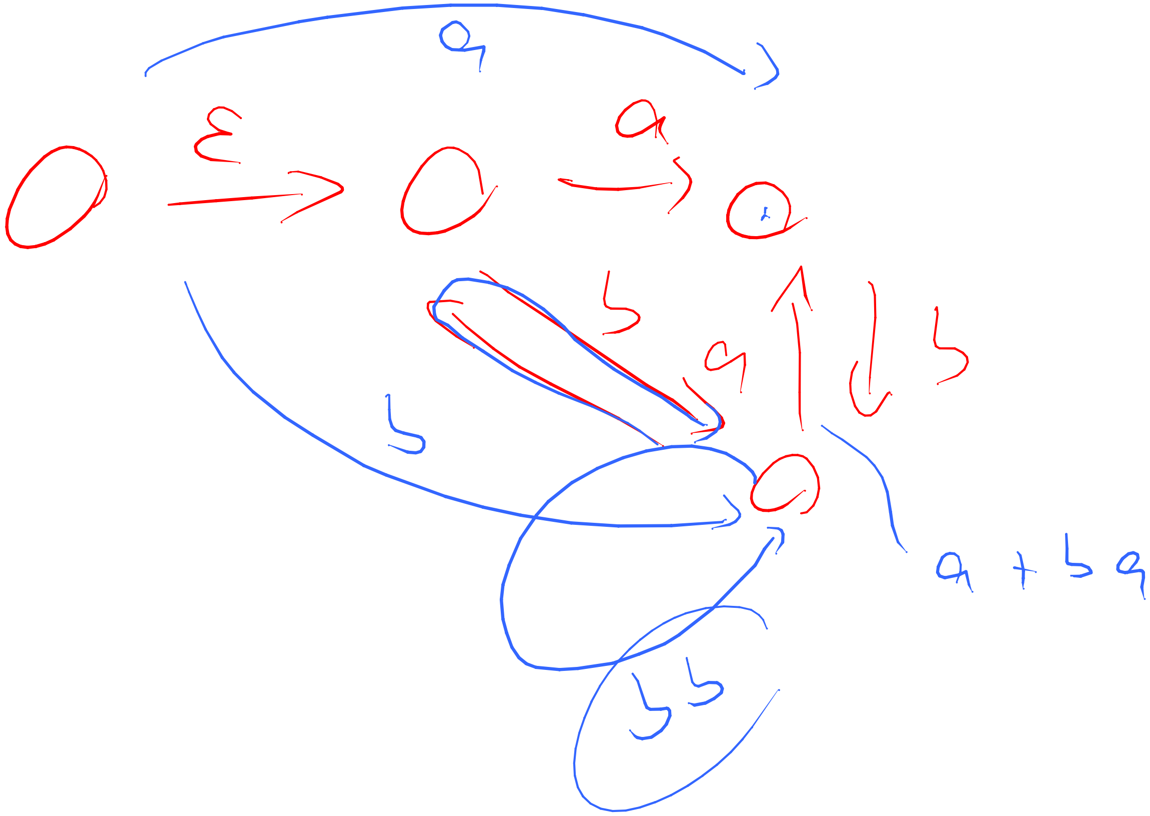
1. **Ergänze neuen Anfangs- und Endzustand**

2. **Transformiere Automaten in VNEA**

Umwandlung NEA → RA am Beispiel



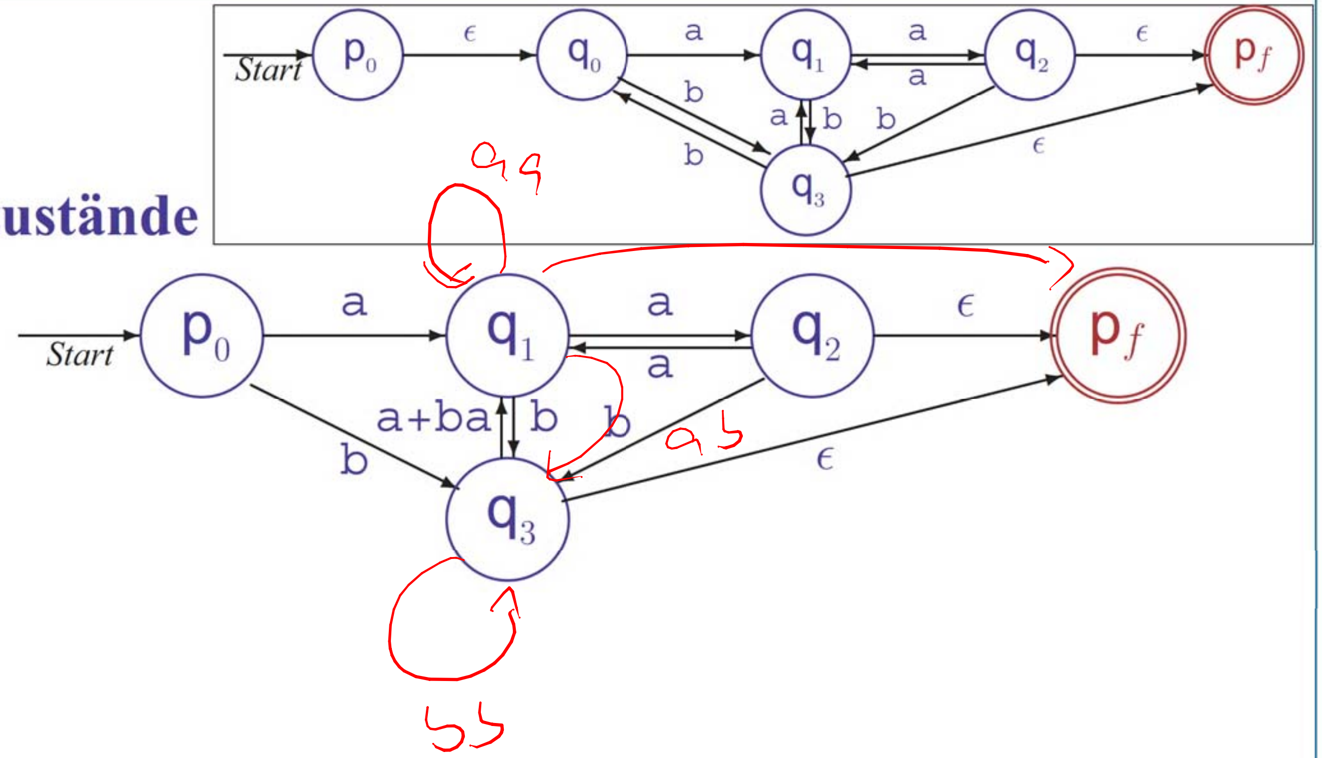
3. Eliminiere Zwischenzustände



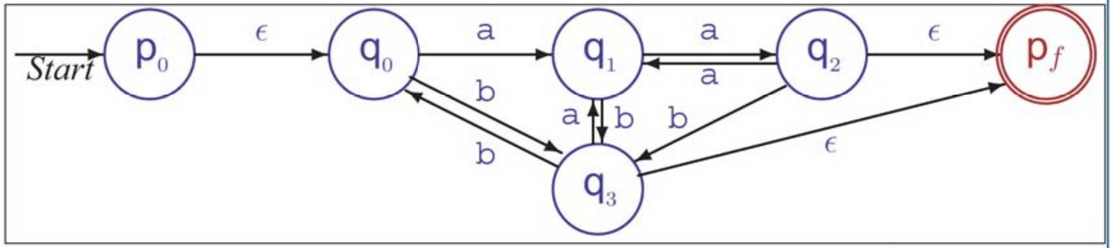
UMWANDLUNG NEA → RA AM BEISPIEL

3. Eliminiere Zwischenzustände

– Eliminiere q_0

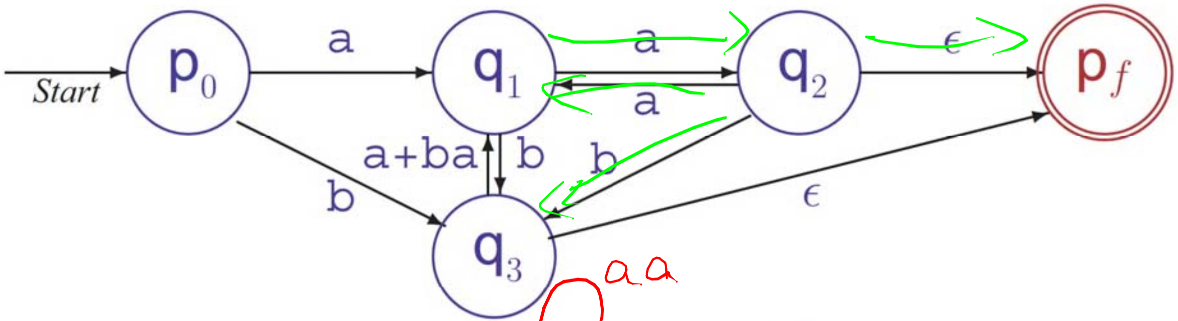


UMWANDLUNG NEA → RA AM BEISPIEL

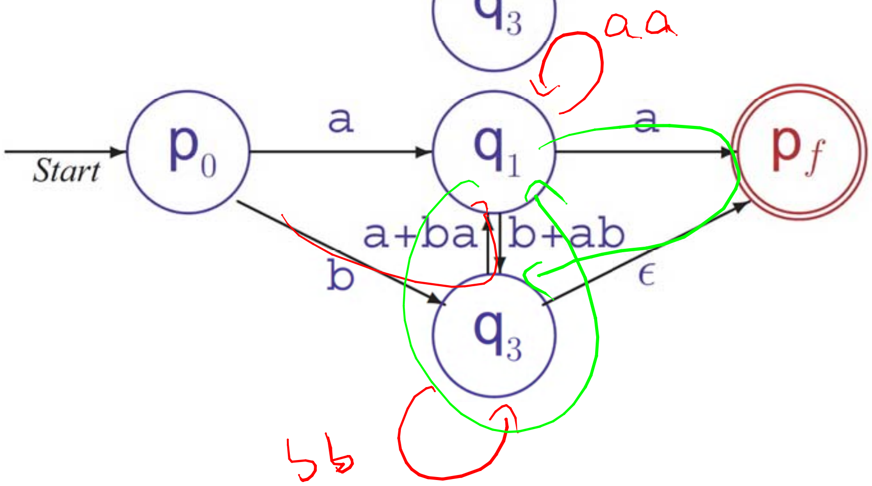


3. Eliminiere Zwischenzustände

- Eliminiere q_0



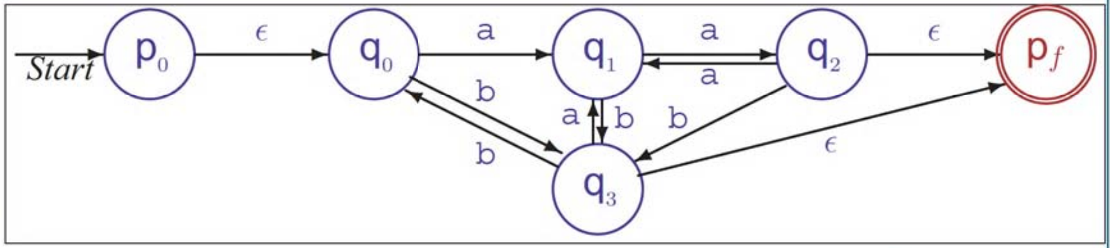
- Eliminiere q_2



$$p_a\,[a+b(a+ba)] \longrightarrow q_1 \xrightarrow{[a+(b+ab)]} p_f$$

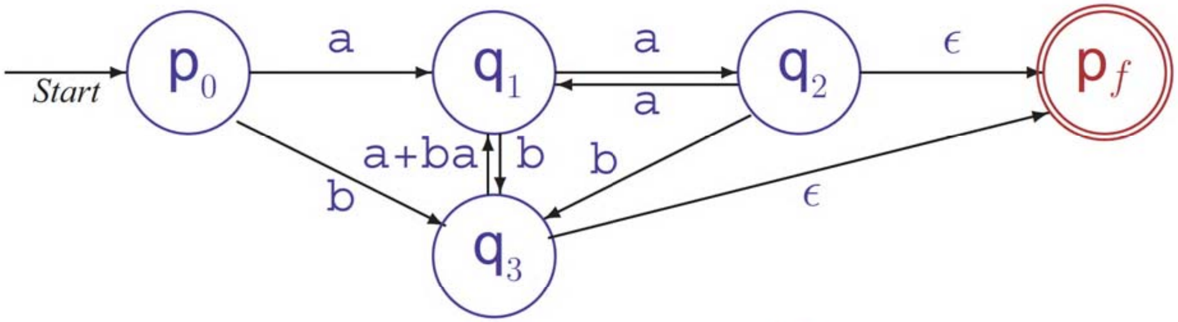
$$[aa+(b+ab)(bb)^*(a+ba)]^*$$

Umwandlung NEA → RA am Beispiel

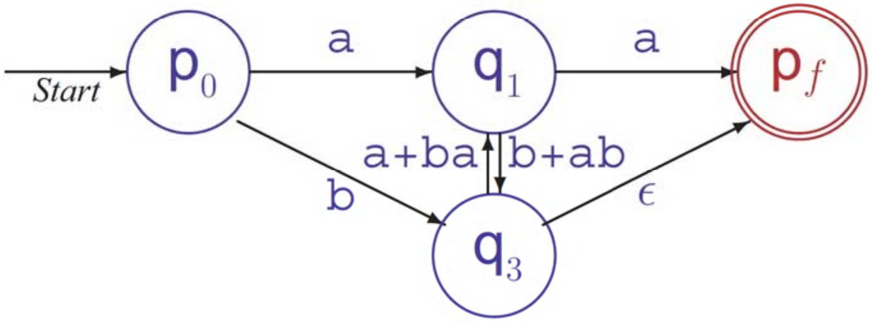


3. Eliminiere Zwischenzustände

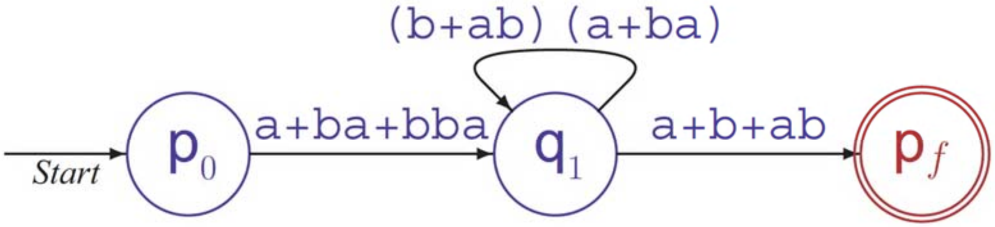
- Eliminiere q_0



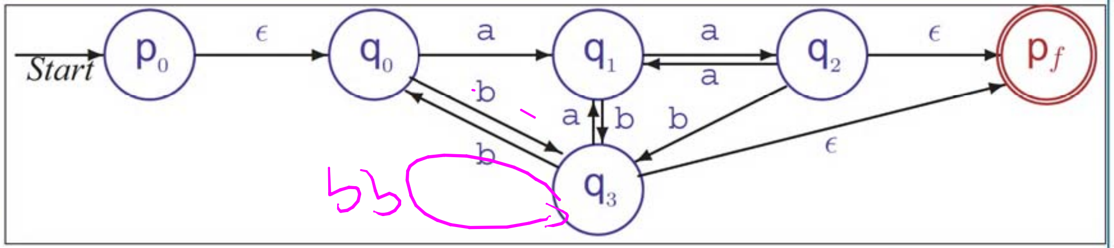
- Eliminiere q_2



- Eliminiere q_3

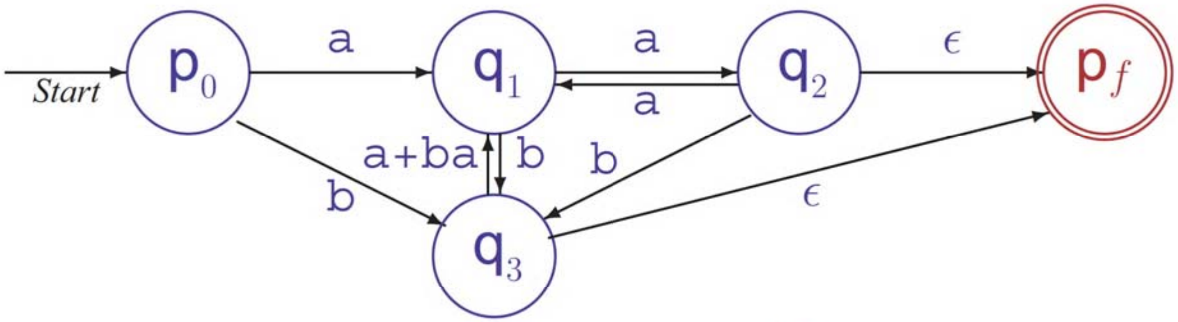


UMWANDLUNG NEA → RA AM BEISPIEL

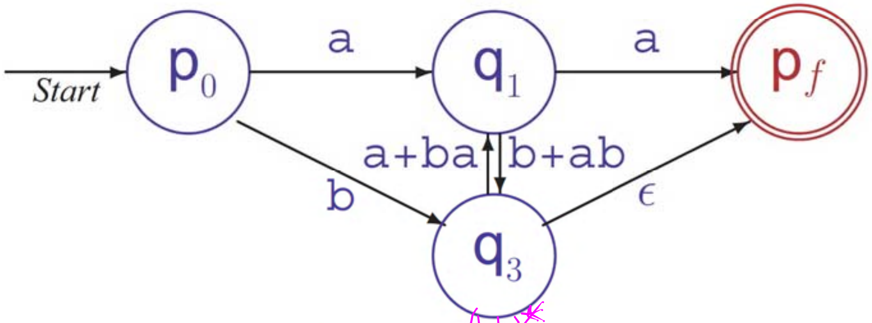


3. Eliminiere Zwischenzustände

– Eliminiere q_0

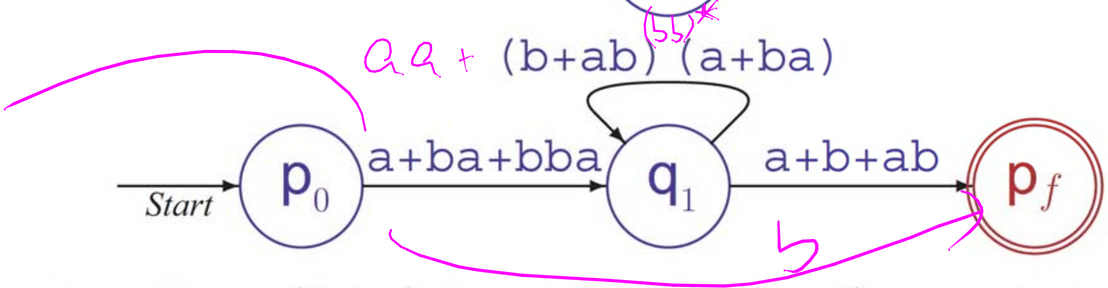


– Eliminiere q_2



– Eliminiere q_3

$a + b\underline{(bb)^*}(a+ba)$



4. Extrahierter Ausdruck ergibt sich nach Elimination von q_1

– $(a+ba+bba)\,(\overset{aa+}{}(b+ab)\overset{(bb)^*}{}(a+ba))^*(a+b+ab)$

REGULÄRE AUSDRÜCKE – ZUSAMMENFASSUNG

- **Algebraische Notation für Sprachen**
  - ϵ, $\emptyset$, Symbole des Alphabets, Vereinigung, Verkettung, Sternoperator
  - Äquivalent zu endlichen Automaten
  - Gut zum Nachweis algebraischer Gesetze von Sprachen
  - Anwendung in Programmiersprachen und Suchmaschinen
- **Transformation in endliche Automaten**
  - Iterative Konstruktion von ϵ-NEAs
  - Nachträgliche Optimierung durch Elimination von ϵ-Übergängen
- **Transformation von Automaten in Ausdrücke**
  - Konstruktion durch Elimination von Zuständen in VNEAs
  - Historisch: Konstruktion von Ausdrücken für Abarbeitungspfade
  - Nachträgliche Optimierungen durch Anwendung algebraischer Gesetze

ANHANG

UMWANDLUNG VON AUTOMATEN IN REGULÄRE AUSDRÜCKE

Originalmethode: allgemeines Graphanalyseverfahren

UMWANDLUNG VON AUTOMATEN IN REGULÄRE AUSDRÜCKE

Originalmethode: allgemeines Graphanalyseverfahren

- **Gegeben DEA** $A = (\{q_1, ..q_n\}, \Sigma, \delta, q_1, \{q_{f_1}, ..q_{f_m}\})$

UMWANDLUNG VON AUTOMATEN IN REGULÄRE AUSDRÜCKE

Originalmethode: allgemeines Graphanalyseverfahren

- **Gegeben DEA A =** $(\{q_1, ..q_n\}, \Sigma, \delta, q_1, \{q_{f_1}, ..q_{f_m}\})$
- **Definiere Ausdrücke für Pfade durch A**
  - R_{ij}^k : Regulärer Ausdruck für Menge der Wörter w mit $\hat{\delta}(q_i, w) = q_j$,
    so dass für alle $\epsilon{\neq}v{\sqsubseteq}w$ $(v{\neq}w)$ gilt: $\hat{\delta}(q_i, v){=}q_m \;\Rightarrow\; m{\leq}k$
    (Abarbeitung von w berührt keinen Zustand größer als k)

UMWANDLUNG VON AUTOMATEN IN REGULÄRE AUSDRÜCKE

Originalmethode: allgemeines Graphanalyseverfahren

- **Gegeben DEA $A = (\{q_1, ..q_n\}, \Sigma, \delta, q_1, \{q_{f_1}, ..q_{f_m}\})$**

- **Definiere Ausdrücke für Pfade durch A**
  - R_{ij}^k : Regulärer Ausdruck für Menge der Wörter w mit $\hat{\delta}(q_i, w) = q_j$,
    so dass für alle $\epsilon{\neq}v{\sqsubseteq}w$ $(v{\neq}w)$ gilt: $\hat{\delta}(q_i, v){=}q_m \;\Rightarrow\; m{\leq}k$
    (Abarbeitung von w berührt keinen Zustand größer als k)

- **Setze die R_{ij}^k zu Ausdruck für $L(A)$ zusammen**
  - Per Definition ist R_{ij}^n ein Ausdruck für Wörter w mit $\hat{\delta}(q_i, w) = q_j$

UMWANDLUNG VON AUTOMATEN IN REGULÄRE AUSDRÜCKE

Originalmethode: allgemeines Graphanalyseverfahren

- **Gegeben DEA $A = (\{q_1, ..q_n\}, \Sigma, \delta, q_1, \{q_{f_1}, ..q_{f_m}\})$**

- **Definiere Ausdrücke für Pfade durch A**
  - R^k_{ij} : Regulärer Ausdruck für Menge der Wörter w mit $\hat{\delta}(q_i, w) = q_j$,
    so dass für alle $\epsilon{\neq}v{\sqsubseteq}w$ $(v{\neq}w)$ gilt: $\hat{\delta}(q_i, v){=}q_m \Rightarrow m{\leq}k$
    (Abarbeitung von w berührt keinen Zustand größer als k)

- **Setze die R^k_{ij} zu Ausdruck für $L(A)$ zusammen**
  - Per Definition ist R^n_{ij} ein Ausdruck für Wörter w mit $\hat{\delta}(q_i, w) = q_j$
  - Setze $R = R^n_{1f_1} + ... + R^n_{1f_m}$

Originalmethode: allgemeines Graphanalyseverfahren

- **Gegeben DEA $A = (\{q_1, ..q_n\}, \Sigma, \delta, q_1, \{q_{f_1}, ..q_{f_m}\})$**
- **Definiere Ausdrücke für Pfade durch A**
  - R^k_{ij} : Regulärer Ausdruck für Menge der Wörter w mit $\hat{\delta}(q_i, w) = q_j$,
    so dass für alle $\epsilon{\neq}v{\sqsubseteq}w$ $(v{\neq}w)$ gilt: $\hat{\delta}(q_i, v){=}q_m \;\Rightarrow\; m{\leq}k$
    (Abarbeitung von w berührt keinen Zustand größer als k)
- **Setze die R^k_{ij} zu Ausdruck für $L(A)$ zusammen**
  - Per Definition ist R^n_{ij} ein Ausdruck für Wörter w mit $\hat{\delta}(q_i, w) = q_j$
  - Setze $R = R^n_{1f_1} + ... + R^n_{1f_m}$
  - Dann gilt $L(R) = \bigcup_{j=1}^m \{w \in \Sigma^* \mid \hat{\delta}(q_1, w) = q_{f_j}\}$

UMWANDLUNG VON AUTOMATEN IN REGULÄRE AUSDRÜCKE

Originalmethode: allgemeines Graphanalyseverfahren

- **Gegeben DEA $A = (\{q_1, ..q_n\}, \Sigma, \delta, q_1, \{q_{f_1}, ..q_{f_m}\})$**

- **Definiere Ausdrücke für Pfade durch A**
  - R_{ij}^k : Regulärer Ausdruck für Menge der Wörter w mit $\hat{\delta}(q_i, w) = q_j$,
    so dass für alle $\epsilon{\neq}v{\sqsubseteq}w$ $(v{\neq}w)$ gilt: $\hat{\delta}(q_i, v){=}q_m \;\Rightarrow\; m{\leq}k$
    (Abarbeitung von w berührt keinen Zustand größer als k)

- **Setze die R_{ij}^k zu Ausdruck für $L(A)$ zusammen**
  - Per Definition ist R_{ij}^n ein Ausdruck für Wörter w mit $\hat{\delta}(q_i, w) = q_j$
  - Setze $R = R_{1f_1}^n + ... + R_{1f_m}^n$
  - Dann gilt $L(R) = \bigcup_{j=1}^m \{w \in \Sigma^* \mid \hat{\delta}(q_1, w) = q_{f_j}\}$
    $= \{w \in \Sigma^* \mid \exists q \in \{q_{f_1}, .., q_{f_m}\}.\ \hat{\delta}(q_1, w) = q\}$

UMWANDLUNG VON AUTOMATEN IN REGULÄRE AUSDRÜCKE

Originalmethode: allgemeines Graphanalyseverfahren

- **Gegeben DEA $A = (\{q_1, ..q_n\}, \Sigma, \delta, q_1, \{q_{f_1}, ..q_{f_m}\})$**

- **Definiere Ausdrücke für Pfade durch A**
  - R^k_{ij} : Regulärer Ausdruck für Menge der Wörter w mit $\hat{\delta}(q_i, w) = q_j$,
    so dass für alle $\epsilon \neq v \sqsubseteq w$ $(v \neq w)$ gilt: $\hat{\delta}(q_i, v) = q_m \Rightarrow m \leq k$
    (Abarbeitung von w berührt keinen Zustand größer als k)

- **Setze die R^k_{ij} zu Ausdruck für $L(A)$ zusammen**
  - Per Definition ist R^n_{ij} ein Ausdruck für Wörter w mit $\hat{\delta}(q_i, w) = q_j$
  - Setze $R = R^n_{1f_1} + ... + R^n_{1f_m}$
  - Dann gilt $L(R) = \bigcup_{j=1}^{m} \{w \in \Sigma^* \mid \hat{\delta}(q_1, w) = q_{f_j}\}$
    $= \{w \in \Sigma^* \mid \exists q \in \{q_{f_1}, .., q_{f_m}\}.\ \hat{\delta}(q_1, w) = q\}$ $= L(A)$

ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren

ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^{k}

- **Basisfall $\boldsymbol{R_{ij}^{0}}$:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^{0})$

ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^0)$

ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren
  - – Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - – Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i,a) = q_j\} \subseteq L(R_{ij}^0)$
  - – Ergebnis: $R_{ii}^0 = \epsilon + \mathtt{a}_1 + \mathtt{..} + \mathtt{a}_k$, wobei $\{a_1,..,a_k\} = \{a \in \Sigma \mid \delta(q_i,a){=}q_j\}$

    $R_{ij}^0 = \emptyset + \mathtt{a}_1 + \mathtt{..} + \mathtt{a}_k$ $(i{\neq}j)$

ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^0)$
  - Ergebnis: $R_{ii}^0 = \epsilon + \mathtt{a}_1 + \,..\, + \mathtt{a}_k$, wobei $\{a_1, .., a_k\} = \{a \in \Sigma \mid \delta(q_i, a) {=} q_j\}$

    $R_{ij}^0 = \emptyset + \mathtt{a}_1 + \,..\, + \mathtt{a}_k \;\; (i{\neq}j)$

- **Schrittfall R_{ij}^k** $(0{<}k{\leq}n)$: zwei Alternativen

ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i,a) = q_j\} \subseteq L(R_{ij}^0)$
  - Ergebnis: $R_{ii}^0 = \epsilon + a_1 + .. + a_k$, wobei $\{a_1,..,a_k\} = \{a \in \Sigma \mid \delta(q_i,a){=}q_j\}$
    $R_{ij}^0 = \emptyset + a_1 + .. + a_k$ $(i{\neq}j)$
- **Schrittfall R_{ij}^k** $(0{<}k{\leq}n)$: zwei Alternativen
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k nicht enthält, gehören zu $L(R_{ij}^{k-1})$

ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^0)$
  - Ergebnis: $R_{ii}^0 = \epsilon + \mathtt{a}_1 + \mathtt{..} + \mathtt{a}_k$, wobei $\{a_1, .., a_k\} = \{a \in \Sigma \mid \delta(q_i, a){=}q_j\}$

    $R_{ij}^0 = \emptyset + \mathtt{a}_1 + \mathtt{..} + \mathtt{a}_k$ $(i{\neq}j)$

- **Schrittfall R_{ij}^k** $(0{<}k{\leq}n)$: zwei Alternativen
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k nicht enthält, gehören zu $L(R_{ij}^{k-1})$
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k enthält:

    Zerlege w in $uz_1..z_pv$ mit $\hat\delta(q_i, u){=}q_k \wedge \forall l{\leq}p.\hat\delta(q_k, z_l){=}q_k \wedge \hat\delta(q_k, v){=}q_j$

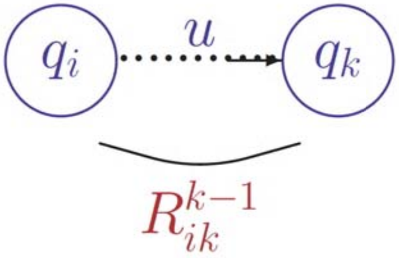
ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall $\boldsymbol{R_{ij}^0}$:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^0)$
  - Ergebnis: $\boldsymbol{R_{ii}^0} = \epsilon + \mathtt{a}_1 + \,.\,.\, + \mathtt{a}_k$, wobei $\{a_1, .., a_k\} = \{a \in \Sigma \mid \delta(q_i, a){=}q_j\}$

    $\boldsymbol{R_{ij}^0} = \emptyset + \mathtt{a}_1 + \,.\,.\, + \mathtt{a}_k$ $(i{\neq}j)$

- **Schrittfall $\boldsymbol{R_{ij}^k}$** $(0{<}k{\leq}n)$: zwei Alternativen
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k nicht enthält, gehören zu $L(\boldsymbol{R_{ij}^{k-1}})$
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k enthält:

    Zerlege w in $uz_1..z_pv$ mit $\hat{\delta}(q_i, u){=}q_k \wedge \forall l{\leq}p.\hat{\delta}(q_k, z_l){=}q_k \wedge \hat{\delta}(q_k, v){=}q_j$



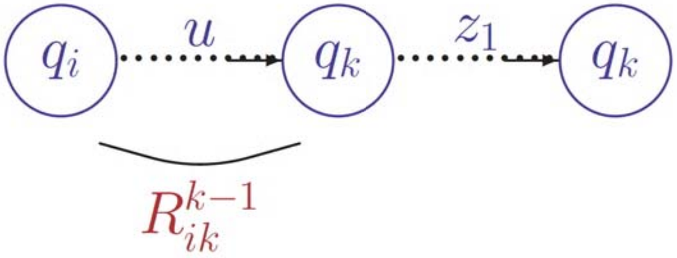
ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^0)$
  - Ergebnis: $R_{ii}^0 = \epsilon + \mathtt{a}_1 + \,..\, + \mathtt{a}_k$, wobei $\{a_1, .., a_k\} = \{a \in \Sigma \mid \delta(q_i, a){=}q_j\}$

    $R_{ij}^0 = \emptyset + \mathtt{a}_1 + \,..\, + \mathtt{a}_k$ $(i{\neq}j)$

- **Schrittfall R_{ij}^k** $(0{<}k{\leq}n)$: zwei Alternativen
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k nicht enthält, gehören zu $L(R_{ij}^{k-1})$
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k enthält:

    Zerlege w in $uz_1..z_pv$ mit $\hat{\delta}(q_i, u){=}q_k \wedge \forall l{\leq}p.\hat{\delta}(q_k, z_l){=}q_k \wedge \hat{\delta}(q_k, v){=}q_j$

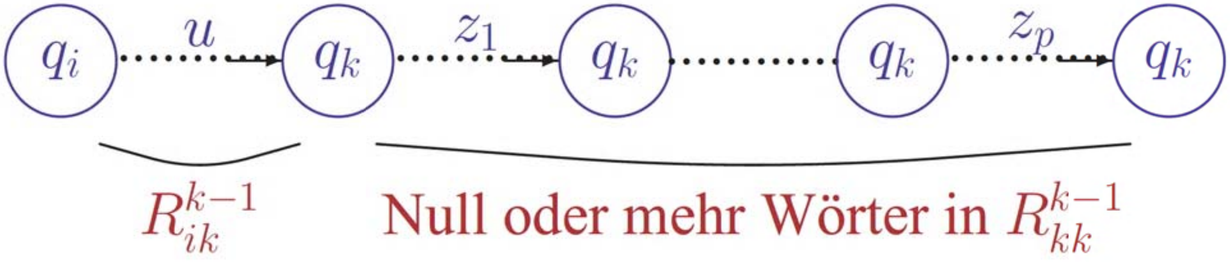


ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^k

- **Basisfall R_{ij}^0:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^0)$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^0)$
  - Ergebnis: $R_{ii}^0 = \epsilon + \mathtt{a}_1 + \,..\, + \mathtt{a}_k$, wobei $\{a_1, .., a_k\} = \{a \in \Sigma \mid \delta(q_i, a){=}q_j\}$

    $R_{ij}^0 = \emptyset + \mathtt{a}_1 + \,..\, + \mathtt{a}_k$ $(i{\neq}j)$
- **Schrittfall R_{ij}^k** $(0{<}k{\leq}n)$: zwei Alternativen
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k nicht enthält, gehören zu $L(R_{ij}^{k-1})$
  - Wörter $w \in L(R_{ij}^k)$, deren Pfad q_k enthält:

    Zerlege w in $uz_1..z_pv$ mit $\hat{\delta}(q_i, u){=}q_k \wedge \forall l{\leq}p.\hat{\delta}(q_k, z_l){=}q_k \wedge \hat{\delta}(q_k, v){=}q_j$



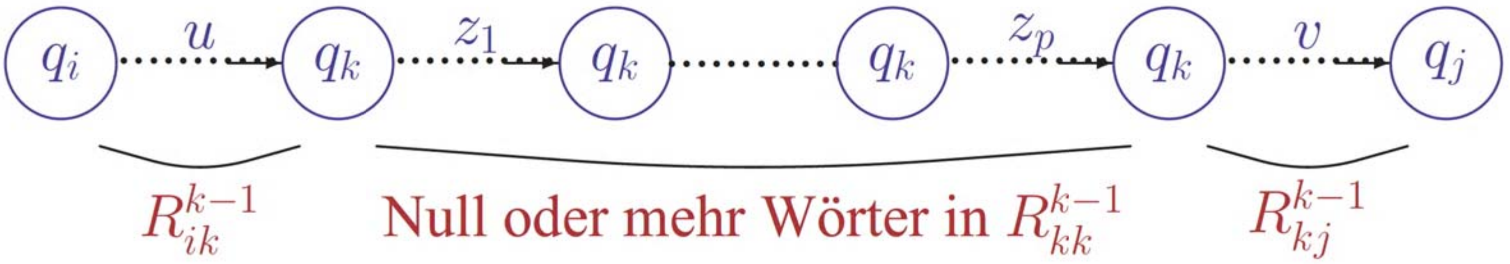
ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^{k}

- **Basisfall $\boldsymbol{R_{ij}^{0}}$:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^{0})$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^{0})$
  - Ergebnis: $\boldsymbol{R_{ii}^{0}} = \epsilon + \mathtt{a}_1 + \ldots + \mathtt{a}_k$, wobei $\{a_1, .., a_k\} = \{a \in \Sigma \mid \delta(q_i, a){=}q_j\}$

    $\boldsymbol{R_{ij}^{0}} = \emptyset + \mathtt{a}_1 + \ldots + \mathtt{a}_k$ $(i{\neq}j)$

- **Schrittfall $\boldsymbol{R_{ij}^{k}}$** $(0{<}k{\leq}n)$: zwei Alternativen
  - Wörter $w \in L(R_{ij}^{k})$, deren Pfad q_k nicht enthält, gehören zu $L(\boldsymbol{R_{ij}^{k-1}})$
  - Wörter $w \in L(R_{ij}^{k})$, deren Pfad q_k enthält:

    Zerlege w in $uz_1..z_pv$ mit $\hat{\delta}(q_i, u){=}q_k \wedge \forall l{\leq}p.\hat{\delta}(q_k, z_l){=}q_k \wedge \hat{\delta}(q_k, v){=}q_j$



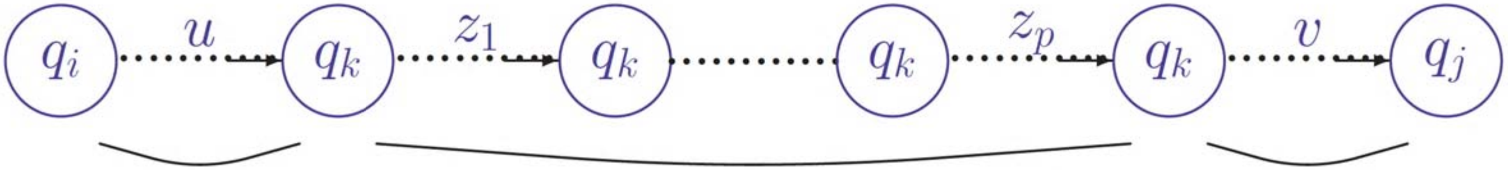
ITERATIVE BESTIMMUNG DER AUSDRÜCKE R_{ij}^{k}

- **Basisfall R_{ij}^{0}:** Pfad darf zwischendurch keine Zustände berühren
  - Pfadlänge 0 (nur für $i{=}j$): $\epsilon \in L(R_{ii}^{0})$
  - Pfadlänge 1: $\{a \in \Sigma \mid \delta(q_i, a) = q_j\} \subseteq L(R_{ij}^{0})$
  - Ergebnis: $R_{ii}^{0} = \epsilon + \mathtt{a}_1 + \,.\,.\, + \mathtt{a}_k$, wobei $\{a_1, .., a_k\} = \{a \in \Sigma \mid \delta(q_i, a){=}q_j\}$

    $R_{ij}^{0} = \emptyset + \mathtt{a}_1 + \,.\,.\, + \mathtt{a}_k$ $(i{\neq}j)$

- **Schrittfall R_{ij}^{k}** $(0{<}k{\leq}n)$: zwei Alternativen
  - Wörter $w \in L(R_{ij}^{k})$, deren Pfad q_k nicht enthält, gehören zu $L(R_{ij}^{k-1})$
  - Wörter $w \in L(R_{ij}^{k})$, deren Pfad q_k enthält:

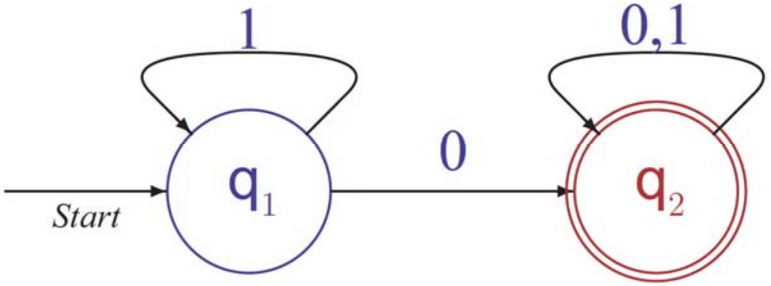
    Zerlege w in $uz_1..z_pv$ mit $\hat{\delta}(q_i, u){=}q_k \land \forall l{\leq}p.\hat{\delta}(q_k, z_l){=}q_k \land \hat{\delta}(q_k, v){=}q_j$



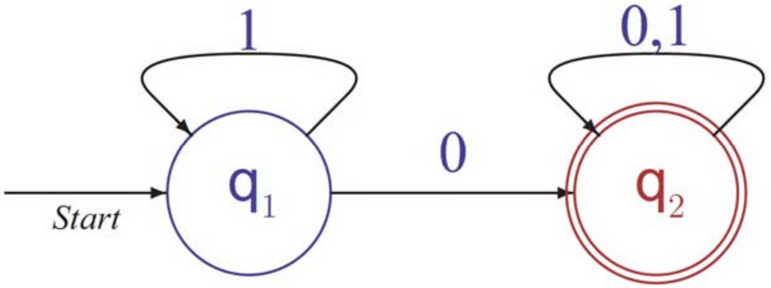
R_{ik}^{k-1} Null oder mehr Wörter in R_{kk}^{k-1} R_{kj}^{k-1}

  - Ergebnis: $R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{k-1} \circ (R_{kk}^{k-1})^{*} \circ R_{kj}^{k-1}$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL



UMWANDLUNG VON AUTOMATEN AM BEISPIEL



- **Basisfall**

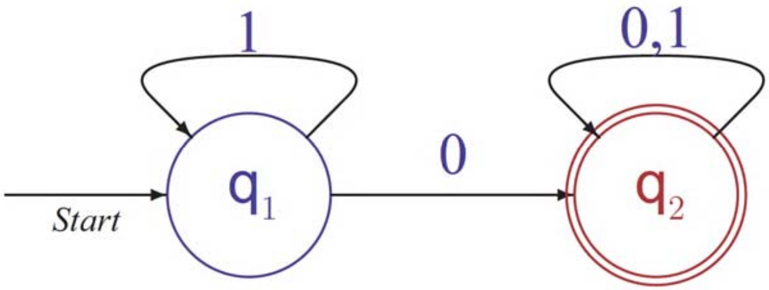
  $R_{11}^{0} = \epsilon$ `+ 1`

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

- **Basisfall**

  $R_{11}^{0} = \epsilon\ +\ \mathtt{1}$

  $R_{12}^{0} = \mathtt{0}$



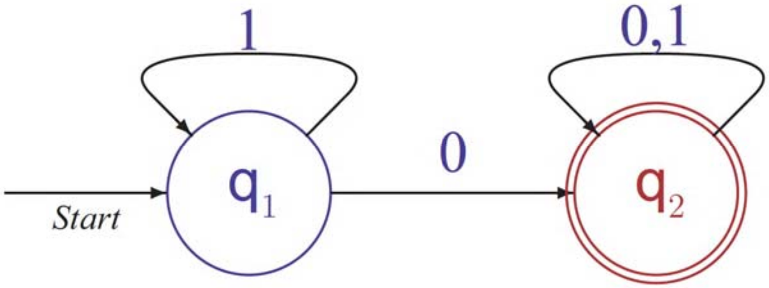
UMWANDLUNG VON AUTOMATEN AM BEISPIEL

- **Basisfall**

  $R_{11}^{0} = \epsilon$ `+ 1`

  $R_{12}^{0} =$ `0`

  $R_{21}^{0} = \emptyset$



UMWANDLUNG VON AUTOMATEN AM BEISPIEL

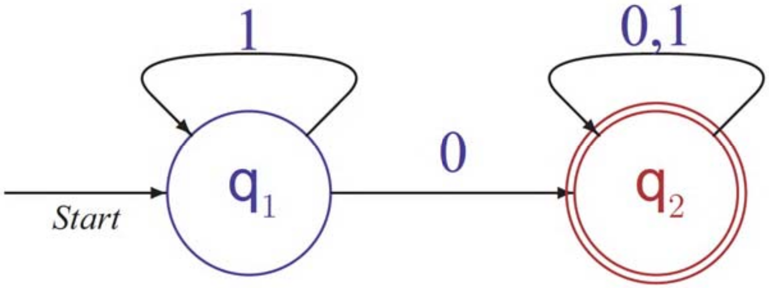
- **Basisfall**

  $R_{11}^{0} = \epsilon\ +\ 1$

  $R_{12}^{0} = 0$

  $R_{21}^{0} = \emptyset$

  $R_{22}^{0} = \epsilon\ +\ 0\ +\ 1$



UMWANDLUNG VON AUTOMATEN AM BEISPIEL

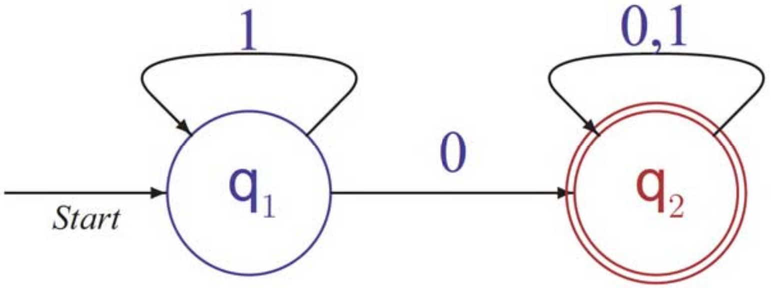
- **Basisfall**

  $R^0_{11} = \epsilon + 1$

  $R^0_{12} = 0$

  $R^0_{21} = \emptyset$

  $R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

  $R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1)$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

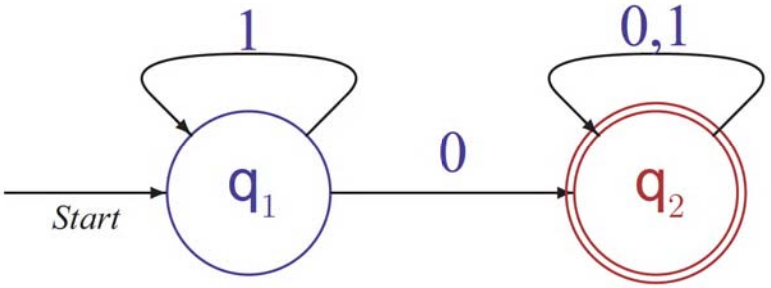
- **Basisfall**

  $R^0_{11} = \epsilon + 1$

  $R^0_{12} = 0$

  $R^0_{21} = \emptyset$

  $R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

  $R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1)$

  $R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)(\epsilon + 1)^* 0$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

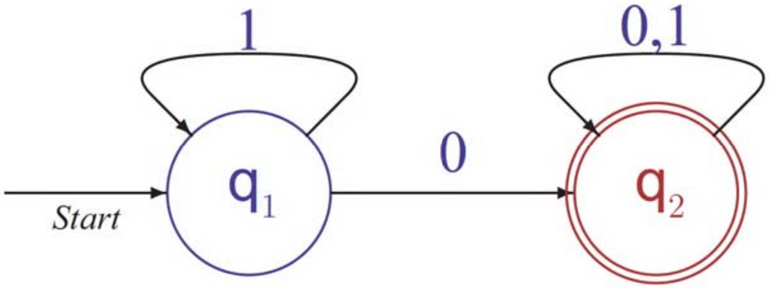
- **Basisfall**

  $R_{11}^{0} = \epsilon + 1$

  $R_{12}^{0} = 0$

  $R_{21}^{0} = \emptyset$

  $R_{22}^{0} = \epsilon + 0 + 1$



- **Stufe 1**

  $R_{11}^{1} = R_{11}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^{*}(\epsilon + 1)$

  $R_{12}^{1} = R_{12}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{12}^{0} = 0 + (\epsilon + 1)(\epsilon + 1)^{*}0$

  $R_{21}^{1} = R_{21}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \emptyset + \emptyset(\epsilon + 1)^{*}(\epsilon + 1)$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

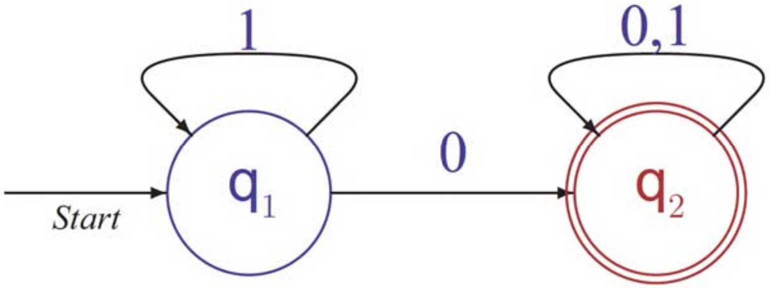
- **Basisfall**

  $R_{11}^{0} = \epsilon + 1$

  $R_{12}^{0} = 0$

  $R_{21}^{0} = \emptyset$

  $R_{22}^{0} = \epsilon + 0 + 1$



- **Stufe 1**

  $R_{11}^{1} = R_{11}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^{*}(\epsilon + 1)$

  $R_{12}^{1} = R_{12}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{12}^{0} = 0 + (\epsilon + 1)(\epsilon + 1)^{*}0$

  $R_{21}^{1} = R_{21}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \emptyset + \emptyset(\epsilon + 1)^{*}(\epsilon + 1)$

  $R_{22}^{1} = R_{22}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{12}^{0} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^{*}0$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

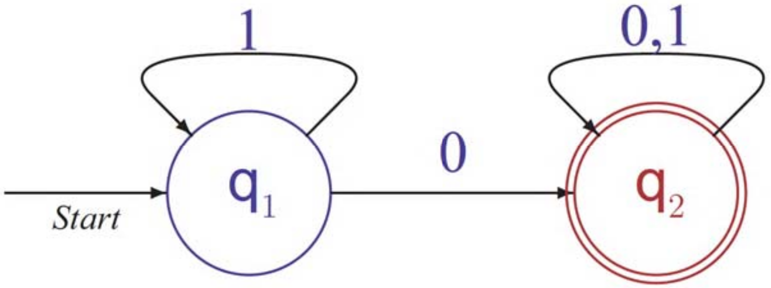
- **Basisfall**

  $R^0_{11} = \epsilon + 1$

  $R^0_{12} = 0$

  $R^0_{21} = \emptyset$

  $R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

  $R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto \quad 1^*$

  $R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)(\epsilon + 1)^* 0$

  $R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1)$

  $R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

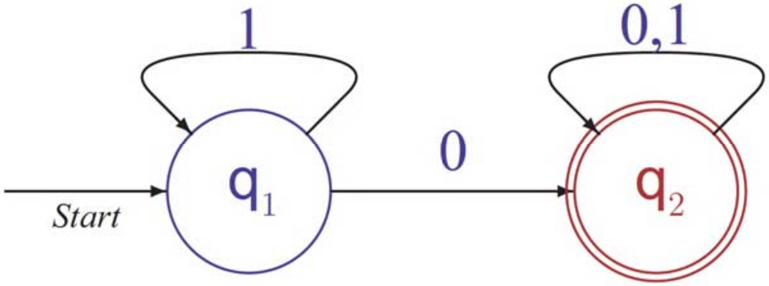
- **Basisfall**

  $R^0_{11} = \epsilon + 1$

  $R^0_{12} = 0$

  $R^0_{21} = \emptyset$

  $R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

  $R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto 1^*$

  $R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)(\epsilon + 1)^* 0 \quad \mapsto 1^* 0$

  $R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1)$

  $R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0$

Umwandlung von Automaten am Beispiel

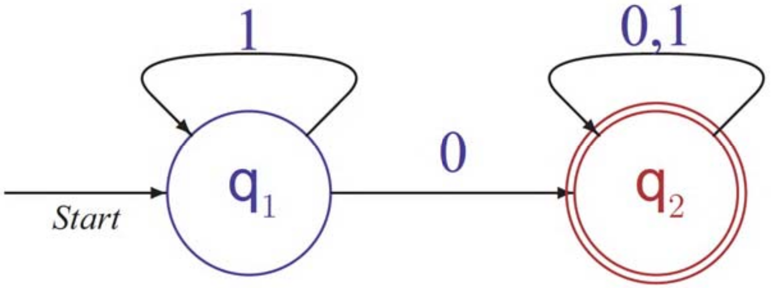
- **Basisfall**

$R^0_{11} = \epsilon + 1$

$R^0_{12} = 0$

$R^0_{21} = \emptyset$

$R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

$R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto \quad 1^*$

$R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)(\epsilon + 1)^* 0 \quad \mapsto \quad 1^* 0$

$R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto \quad \emptyset$

$R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

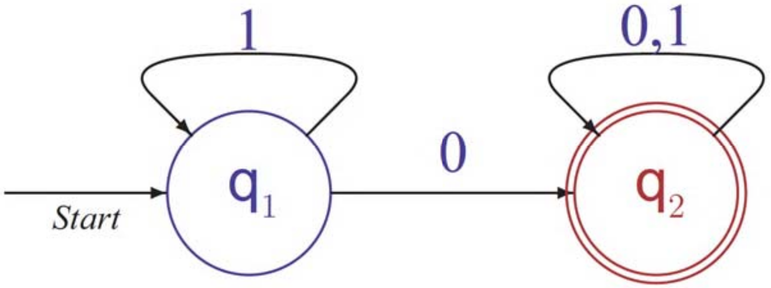
- **Basisfall**

  $R_{11}^0 = \epsilon + 1$

  $R_{12}^0 = 0$

  $R_{21}^0 = \emptyset$

  $R_{22}^0 = \epsilon + 0 + 1$



- **Stufe 1**

  $R_{11}^1 = R_{11}^0 + R_{11}^0(R_{11}^0)^* R_{11}^0 = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto \quad 1^*$

  $R_{12}^1 = R_{12}^0 + R_{11}^0(R_{11}^0)^* R_{12}^0 = 0 + (\epsilon + 1)(\epsilon + 1)^* 0 \quad \mapsto \quad 1^* 0$

  $R_{21}^1 = R_{21}^0 + R_{21}^0(R_{11}^0)^* R_{11}^0 = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto \quad \emptyset$

  $R_{22}^1 = R_{22}^0 + R_{21}^0(R_{11}^0)^* R_{12}^0 = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0 \quad \mapsto \quad \epsilon + 0 + 1$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

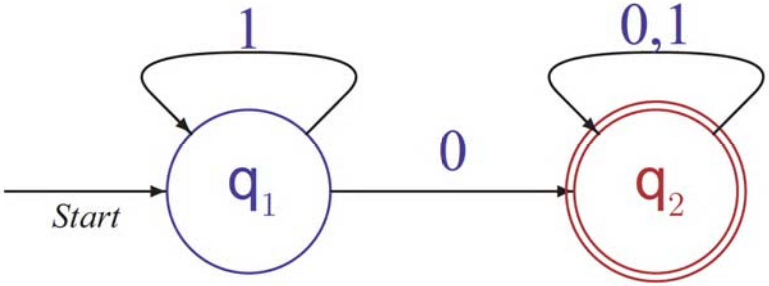
- **Basisfall**

$R^0_{11} = \epsilon + 1$

$R^0_{12} = 0$

$R^0_{21} = \emptyset$

$R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

$$R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)\,(\epsilon + 1)^*(\epsilon + 1) \qquad \mapsto \; 1^*$$

$$R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)\,(\epsilon + 1)^* 0 \qquad \mapsto \; 1^* 0$$

$$R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset\,(\epsilon + 1)^*(\epsilon + 1) \qquad \mapsto \; \emptyset$$

$$R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset\,(\epsilon + 1)^* 0 \qquad \mapsto \; \epsilon + 0 + 1$$

- **Stufe 2**

$$R^2_{11} = R^1_{11} + R^1_{12}(R^1_{22})^* R^1_{21} = 1^* + 1^* 0\,(\epsilon + 0 + 1)^* \emptyset$$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

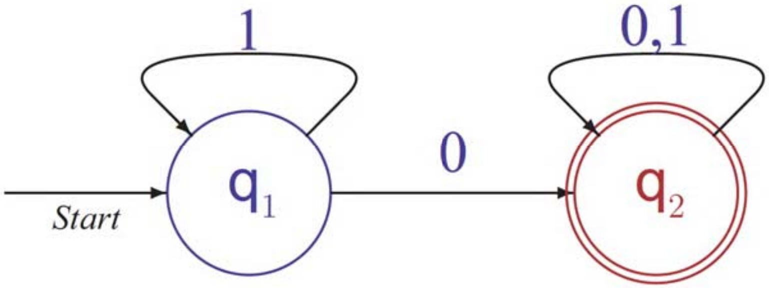
- **Basisfall**

$R_{11}^{0} = \epsilon + 1$

$R_{12}^{0} = 0$

$R_{21}^{0} = \emptyset$

$R_{22}^{0} = \epsilon + 0 + 1$



- **Stufe 1**

$$R_{11}^{1} = R_{11}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^{*}(\epsilon + 1) \quad \mapsto \quad 1^{*}$$

$$R_{12}^{1} = R_{12}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{12}^{0} = 0 + (\epsilon + 1)(\epsilon + 1)^{*}0 \quad \mapsto \quad 1^{*}0$$

$$R_{21}^{1} = R_{21}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \emptyset + \emptyset(\epsilon + 1)^{*}(\epsilon + 1) \quad \mapsto \quad \emptyset$$

$$R_{22}^{1} = R_{22}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{12}^{0} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^{*}0 \quad \mapsto \quad \epsilon + 0 + 1$$

- **Stufe 2**

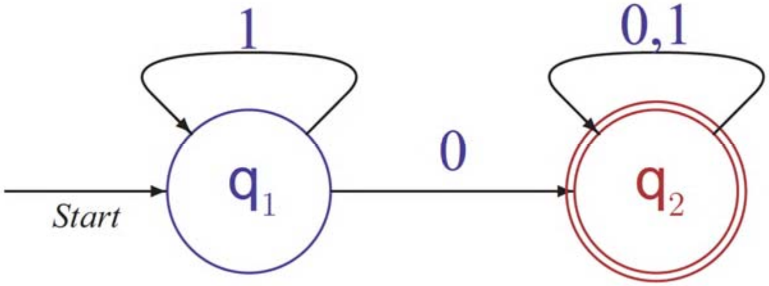
$$R_{11}^{2} = R_{11}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{21}^{1} = 1^{*} + 1^{*}0(\epsilon+0+1)^{*}\emptyset$$

$$R_{12}^{2} = R_{12}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{22}^{1} = 1^{*}0 + 1^{*}0(\epsilon+0+1)^{*}(\epsilon+0+1)$$

Umwandlung von Automaten am Beispiel

- **Basisfall**

$$R_{11}^{0} = \epsilon \;+\; 1$$
$$R_{12}^{0} = 0$$
$$R_{21}^{0} = \emptyset$$
$$R_{22}^{0} = \epsilon \;+\; 0 \;+\; 1$$



- **Stufe 1**

$$R_{11}^{1} = R_{11}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^{*}(\epsilon + 1) \quad\mapsto\quad 1^{*}$$
$$R_{12}^{1} = R_{12}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{12}^{0} = 0 + (\epsilon + 1)(\epsilon + 1)^{*}0 \quad\mapsto\quad 1^{*}0$$
$$R_{21}^{1} = R_{21}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \emptyset + \emptyset(\epsilon + 1)^{*}(\epsilon + 1) \quad\mapsto\quad \emptyset$$
$$R_{22}^{1} = R_{22}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{12}^{0} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^{*}0 \quad\mapsto\quad \epsilon + 0 + 1$$

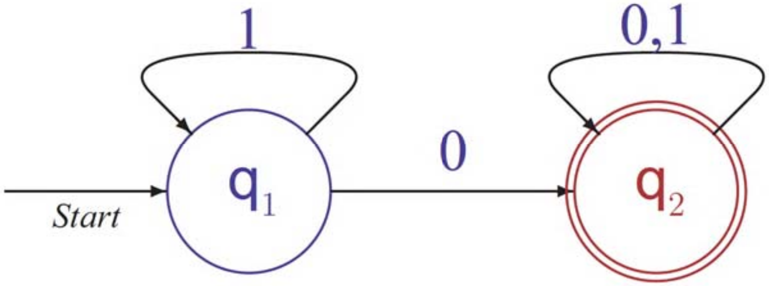
- **Stufe 2**

$$R_{11}^{2} = R_{11}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{21}^{1} = 1^{*} + 1^{*}0(\epsilon+0+1)^{*}\emptyset$$
$$R_{12}^{2} = R_{12}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{22}^{1} = 1^{*}0 + 1^{*}0(\epsilon+0+1)^{*}(\epsilon+0+1)$$
$$R_{21}^{2} = R_{21}^{1} + R_{22}^{1}(R_{22}^{1})^{*}R_{21}^{1} = \emptyset + (\epsilon+0+1)(\epsilon+0+1)^{*}\emptyset$$

Umwandlung von Automaten am Beispiel

- **Basisfall**

$$R^0_{11} = \epsilon + 1$$
$$R^0_{12} = 0$$
$$R^0_{21} = \emptyset$$
$$R^0_{22} = \epsilon + 0 + 1$$



- **Stufe 1**

$$R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto 1^*$$
$$R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)(\epsilon + 1)^* 0 \quad \mapsto 1^* 0$$
$$R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1) \quad \mapsto \emptyset$$
$$R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0 \quad \mapsto \epsilon + 0 + 1$$

- **Stufe 2**

$$R^2_{11} = R^1_{11} + R^1_{12}(R^1_{22})^* R^1_{21} = 1^* + 1^* 0 (\epsilon+0+1)^* \emptyset$$
$$R^2_{12} = R^1_{12} + R^1_{12}(R^1_{22})^* R^1_{22} = 1^* 0 + 1^* 0 (\epsilon+0+1)^* (\epsilon+0+1)$$
$$R^2_{21} = R^1_{21} + R^1_{22}(R^1_{22})^* R^1_{21} = \emptyset + (\epsilon+0+1)(\epsilon+0+1)^* \emptyset$$
$$R^2_{22} = R^1_{22} + R^1_{22}(R^1_{22})^* R^1_{22} = (\epsilon+0+1) + (\epsilon+0+1)(\epsilon+0+1)^*(\epsilon+0+1)$$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

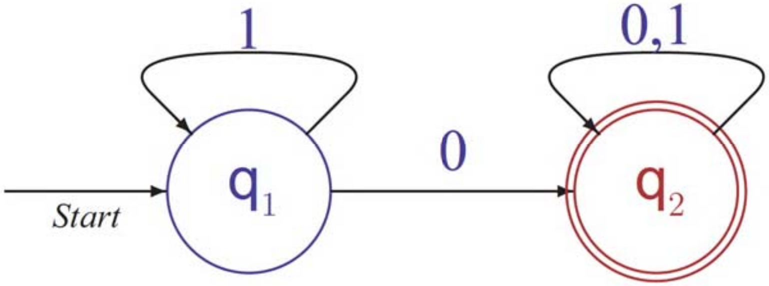
- **Basisfall**

$R_{11}^0 = \epsilon + 1$

$R_{12}^0 = 0$

$R_{21}^0 = \emptyset$

$R_{22}^0 = \epsilon + 0 + 1$



- **Stufe 1**

$$R_{11}^1 = R_{11}^0 + R_{11}^0(R_{11}^0)^* R_{11}^0 = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1) \qquad \mapsto 1^*$$

$$R_{12}^1 = R_{12}^0 + R_{11}^0(R_{11}^0)^* R_{12}^0 = 0 + (\epsilon + 1)(\epsilon + 1)^* 0 \qquad \mapsto 1^* 0$$

$$R_{21}^1 = R_{21}^0 + R_{21}^0(R_{11}^0)^* R_{11}^0 = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1) \qquad \mapsto \emptyset$$

$$R_{22}^1 = R_{22}^0 + R_{21}^0(R_{11}^0)^* R_{12}^0 = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0 \qquad \mapsto \epsilon + 0 + 1$$

- **Stufe 2**

$$R_{11}^2 = R_{11}^1 + R_{12}^1(R_{22}^1)^* R_{21}^1 = 1^* + 1^* 0(\epsilon+0+1)^* \emptyset \qquad \mapsto 1^*$$

$$R_{12}^2 = R_{12}^1 + R_{12}^1(R_{22}^1)^* R_{22}^1 = 1^* 0 + 1^* 0(\epsilon+0+1)^*(\epsilon+0+1)$$

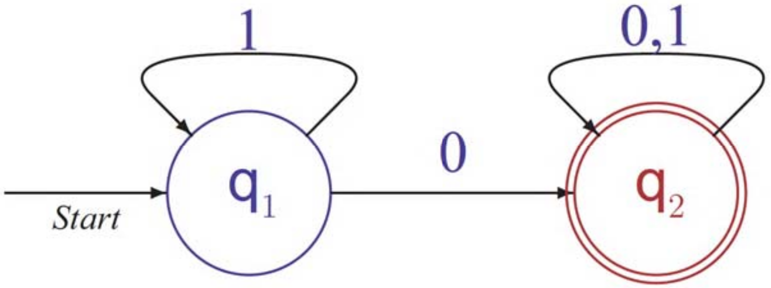
$$R_{21}^2 = R_{21}^1 + R_{22}^1(R_{22}^1)^* R_{21}^1 = \emptyset + (\epsilon+0+1)(\epsilon+0+1)^* \emptyset$$

$$R_{22}^2 = R_{22}^1 + R_{22}^1(R_{22}^1)^* R_{22}^1 = (\epsilon+0+1) + (\epsilon+0+1)(\epsilon+0+1)^*(\epsilon+0+1)$$

Umwandlung von Automaten am Beispiel

- **Basisfall**

$$R^0_{11} = \epsilon + 1$$
$$R^0_{12} = 0$$
$$R^0_{21} = \emptyset$$
$$R^0_{22} = \epsilon + 0 + 1$$



- **Stufe 1**

$$R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)\,(\epsilon + 1)^*\,(\epsilon + 1) \mapsto 1^*$$
$$R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)\,(\epsilon + 1)^* 0 \mapsto 1^* 0$$
$$R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset\,(\epsilon + 1)^*\,(\epsilon + 1) \mapsto \emptyset$$
$$R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset\,(\epsilon + 1)^* 0 \mapsto \epsilon + 0 + 1$$

- **Stufe 2**

$$R^2_{11} = R^1_{11} + R^1_{12}(R^1_{22})^* R^1_{21} = 1^* + 1^* 0\,(\epsilon+0+1)^* \emptyset \mapsto 1^*$$
$$R^2_{12} = R^1_{12} + R^1_{12}(R^1_{22})^* R^1_{22} = 1^* 0 + 1^* 0\,(\epsilon+0+1)^*\,(\epsilon+0+1) \mapsto 1^* 0\,(0+1)^*$$
$$R^2_{21} = R^1_{21} + R^1_{22}(R^1_{22})^* R^1_{21} = \emptyset + (\epsilon+0+1)\,(\epsilon+0+1)^* \emptyset$$
$$R^2_{22} = R^1_{22} + R^1_{22}(R^1_{22})^* R^1_{22} = (\epsilon+0+1) + (\epsilon+0+1)\,(\epsilon+0+1)^*(\epsilon+0+1)$$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

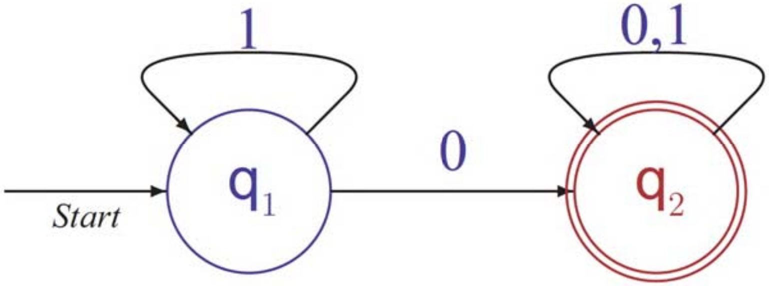
- **Basisfall**

$R_{11}^{0} = \epsilon + 1$

$R_{12}^{0} = 0$

$R_{21}^{0} = \emptyset$

$R_{22}^{0} = \epsilon + 0 + 1$



- **Stufe 1**

$R_{11}^{1} = R_{11}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^{*}(\epsilon + 1) \qquad \mapsto 1^{*}$

$R_{12}^{1} = R_{12}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{12}^{0} = 0 + (\epsilon + 1)(\epsilon + 1)^{*}0 \qquad \mapsto 1^{*}0$

$R_{21}^{1} = R_{21}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \emptyset + \emptyset(\epsilon + 1)^{*}(\epsilon + 1) \qquad \mapsto \emptyset$

$R_{22}^{1} = R_{22}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{12}^{0} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^{*}0 \qquad \mapsto \epsilon + 0 + 1$

- **Stufe 2**

$R_{11}^{2} = R_{11}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{21}^{1} = 1^{*} + 1^{*}0(\epsilon+0+1)^{*}\emptyset \qquad \mapsto 1^{*}$

$R_{12}^{2} = R_{12}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{22}^{1} = 1^{*}0 + 1^{*}0(\epsilon+0+1)^{*}(\epsilon+0+1) \qquad \mapsto 1^{*}0(0+1)^{*}$

$R_{21}^{2} = R_{21}^{1} + R_{22}^{1}(R_{22}^{1})^{*}R_{21}^{1} = \emptyset + (\epsilon+0+1)(\epsilon+0+1)^{*}\emptyset \qquad \mapsto \emptyset$

$R_{22}^{2} = R_{22}^{1} + R_{22}^{1}(R_{22}^{1})^{*}R_{22}^{1} = (\epsilon+0+1) + (\epsilon+0+1)(\epsilon+0+1)^{*}(\epsilon+0+1)$

Umwandlung von Automaten am Beispiel

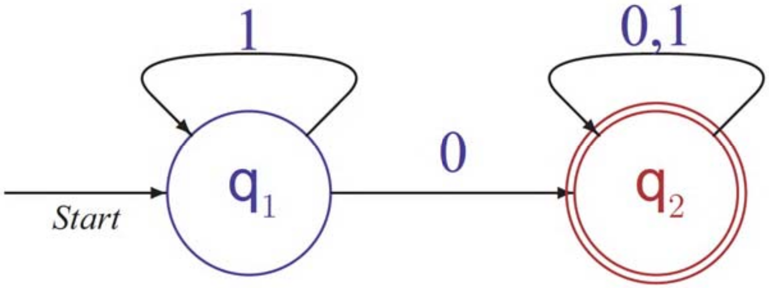
- **Basisfall**

$R^0_{11} = \epsilon + 1$

$R^0_{12} = 0$

$R^0_{21} = \emptyset$

$R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

$R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1) \mapsto 1^*$

$R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)(\epsilon + 1)^* 0 \mapsto 1^* 0$

$R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1) \mapsto \emptyset$

$R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset(\epsilon + 1)^* 0 \mapsto \epsilon + 0 + 1$

- **Stufe 2**

$R^2_{11} = R^1_{11} + R^1_{12}(R^1_{22})^* R^1_{21} = 1^* + 1^* 0 (\epsilon+0+1)^* \emptyset \mapsto 1^*$

$R^2_{12} = R^1_{12} + R^1_{12}(R^1_{22})^* R^1_{22} = 1^* 0 + 1^* 0 (\epsilon+0+1)^*(\epsilon+0+1) \mapsto 1^* 0 (0+1)^*$

$R^2_{21} = R^1_{21} + R^1_{22}(R^1_{22})^* R^1_{21} = \emptyset + (\epsilon+0+1)(\epsilon+0+1)^* \emptyset \mapsto \emptyset$

$R^2_{22} = R^1_{22} + R^1_{22}(R^1_{22})^* R^1_{22} = (\epsilon+0+1) + (\epsilon+0+1)(\epsilon+0+1)^*(\epsilon+0+1) \mapsto (0+1)^*$

Umwandlung von Automaten am Beispiel

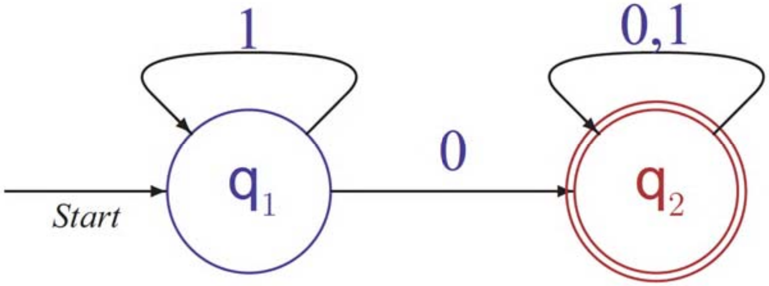
- **Basisfall**

$R^0_{11} = \epsilon + 1$

$R^0_{12} = 0$

$R^0_{21} = \emptyset$

$R^0_{22} = \epsilon + 0 + 1$



- **Stufe 1**

$R^1_{11} = R^0_{11} + R^0_{11}(R^0_{11})^* R^0_{11} = \epsilon + 1 + (\epsilon + 1)\,(\epsilon + 1)^*(\epsilon + 1) \quad\mapsto\; 1^*$

$R^1_{12} = R^0_{12} + R^0_{11}(R^0_{11})^* R^0_{12} = 0 + (\epsilon + 1)\,(\epsilon + 1)^* 0 \quad\mapsto\; 1^* 0$

$R^1_{21} = R^0_{21} + R^0_{21}(R^0_{11})^* R^0_{11} = \emptyset + \emptyset\,(\epsilon + 1)^*(\epsilon + 1) \quad\mapsto\; \emptyset$

$R^1_{22} = R^0_{22} + R^0_{21}(R^0_{11})^* R^0_{12} = \epsilon + 0 + 1 + \emptyset\,(\epsilon + 1)^* 0 \quad\mapsto\; \epsilon + 0 + 1$

- **Stufe 2**

Gebraucht wird nur R^2_{12}

$R^2_{11} = R^1_{11} + R^1_{12}(R^1_{22})^* R^1_{21} = 1^* + 1^* 0\,(\epsilon+0+1)^* \emptyset \quad\mapsto\; 1^*$

$R^2_{12} = R^1_{12} + R^1_{12}(R^1_{22})^* R^1_{22} = 1^* 0 + 1^* 0\,(\epsilon+0+1)^*(\epsilon+0+1) \quad\mapsto\; 1^* 0\,(0+1)^*$

$R^2_{21} = R^1_{21} + R^1_{22}(R^1_{22})^* R^1_{21} = \emptyset + (\epsilon+0+1)\,(\epsilon+0+1)^* \emptyset \quad\mapsto\; \emptyset$

$R^2_{22} = R^1_{22} + R^1_{22}(R^1_{22})^* R^1_{22} = (\epsilon+0+1) + (\epsilon+0+1)\,(\epsilon+0+1)^*(\epsilon+0+1) \;\mapsto\; (0+1)^*$

UMWANDLUNG VON AUTOMATEN AM BEISPIEL

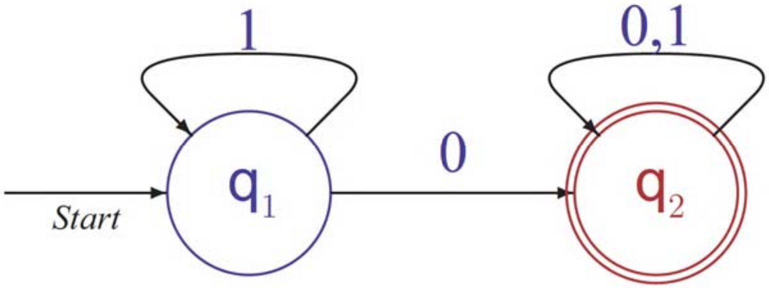
- **Basisfall**

$R_{11}^{0} = \epsilon\ +\ 1$

$R_{12}^{0} = 0$

$R_{21}^{0} = \emptyset$

$R_{22}^{0} = \epsilon\ +\ 0\ +\ 1$



- **Stufe 1**

$R_{11}^{1} = R_{11}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \epsilon + 1 + (\epsilon + 1)\,(\epsilon + 1)^{*}(\epsilon + 1) \quad\mapsto\ 1^{*}$

$R_{12}^{1} = R_{12}^{0} + R_{11}^{0}(R_{11}^{0})^{*}R_{12}^{0} = 0 + (\epsilon + 1)\,(\epsilon + 1)^{*}0 \quad\mapsto\ 1^{*}0$

$R_{21}^{1} = R_{21}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{11}^{0} = \emptyset + \emptyset\,(\epsilon + 1)^{*}(\epsilon + 1) \quad\mapsto\ \emptyset$

$R_{22}^{1} = R_{22}^{0} + R_{21}^{0}(R_{11}^{0})^{*}R_{12}^{0} = \epsilon + 0 + 1 + \emptyset\,(\epsilon + 1)^{*}0 \quad\mapsto\ \epsilon + 0 + 1$

- **Stufe 2**

Gebraucht wird nur R_{12}^{2}

$R_{11}^{2} = R_{11}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{21}^{1} = 1^{*} + 1^{*}0\,(\epsilon{+}0{+}1)^{*}\emptyset \quad\mapsto\ 1^{*}$

$R_{12}^{2} = R_{12}^{1} + R_{12}^{1}(R_{22}^{1})^{*}R_{22}^{1} = 1^{*}0 + 1^{*}0\,(\epsilon{+}0{+}1)^{*}(\epsilon{+}0{+}1) \quad\mapsto\ 1^{*}0\,(0{+}1)^{*}$

$R_{21}^{2} = R_{21}^{1} + R_{22}^{1}(R_{22}^{1})^{*}R_{21}^{1} = \emptyset + (\epsilon{+}0{+}1)\,(\epsilon{+}0{+}1)^{*}\emptyset \quad\mapsto\ \emptyset$

$R_{22}^{2} = R_{22}^{1} + R_{22}^{1}(R_{22}^{1})^{*}R_{22}^{1} = (\epsilon{+}0{+}1) + (\epsilon{+}0{+}1)(\epsilon{+}0{+}1)^{*}(\epsilon{+}0{+}1) \mapsto (0{+}1)^{*}$

Regulärer Ausdruck des Automaten ist: $1^{*}0\,(0{+}1)^{*}$

DAS PFADANALYSEVERFAHREN IST ZU KOMPLIZIERT

- **Konstruktion aller R_{ij}^k ist aufwendig**
  - Es müssen mehr als n^3 Ausdrücke R_{ij}^k erzeugt werden
  - Ausdrücke R_{ij}^k können viermal so groß wie die R_{ij}^{k-1} werden

DAS PFADANALYSEVERFAHREN IST ZU KOMPLIZIERT

- **Konstruktion aller R_{ij}^{k} ist aufwendig**
  - Es müssen mehr als n^3 Ausdrücke R_{ij}^{k} erzeugt werden
  - Ausdrücke R_{ij}^{k} können viermal so groß wie die R_{ij}^{k-1} werden
  - Ohne Vereinfachung der R_{ij}^{k} sind bis zu $n^3 * 4^n$ Symbole zu erzeugen

DAS PFADANALYSEVERFAHREN IST ZU KOMPLIZIERT

- **Konstruktion aller R_{ij}^k ist aufwendig**
  - Es müssen mehr als n^3 Ausdrücke R_{ij}^k erzeugt werden
  - Ausdrücke R_{ij}^k können viermal so groß wie die R_{ij}^{k-1} werden
  - Ohne Vereinfachung der R_{ij}^k sind bis zu $n^3 * 4^n$ Symbole zu erzeugen
- **Optimierungen des Verfahrens sind möglich**
  - Vermeide Vielfachkopien der R_{ij}^{k-1}
  - Vereinfache Ausdrücke R_{ij}^k direkt nach Erzeugung
  - Liefert keine grundsätzliche Verbesserung

DAS PFADANALYSEVERFAHREN IST ZU KOMPLIZIERT

- **Konstruktion aller R_{ij}^k ist aufwendig**
  - Es müssen mehr als n^3 Ausdrücke R_{ij}^k erzeugt werden
  - Ausdrücke R_{ij}^k können viermal so groß wie die R_{ij}^{k-1} werden
  - Ohne Vereinfachung der R_{ij}^k sind bis zu $n^3 * 4^n$ Symbole zu erzeugen
- **Optimierungen des Verfahrens sind möglich**
  - Vermeide Vielfachkopien der R_{ij}^{k-1}
  - Vereinfache Ausdrücke R_{ij}^k direkt nach Erzeugung
  - Liefert keine grundsätzliche Verbesserung

Zustandselimination ist erheblich effizienter