Formale Grundlagen der Informatik

3 Reguläre Ausdrücke und Sprachen

ε-NEA

Reguläre Ausdrücke in der Praxis

- **Suchmuster**
  - in Texteditoren und anderen Tools (z.B. in UNIX: **`grep`**)
  - Suche nach allen Zeichenketten, die auf einen regulären Ausdruck passen
  - *Beispiele*: `FGI.*` *oder* `FGI + GdP`

- **Tokenisierung**
  - von Interpretern, Compilern, bei der Analyse von Sprache benötigt
  - Tokenisierung: Umwandlung von Character-Folgen in zugehörige **Token**
  - *Beispiele:* Schlüsselwörter, Identifier, …
    `while` *oder* `('_'+[A-Z]+[a-z])('_'+[0-9]+[a-z]+[A-Z])*`

Reguläre Ausdrücke – Definition

Definition: Sei Σ ein Alphabet.
Reguläre Ausdrücke über Σ sind rekursiv wie folgt definiert:

1. $\emptyset$, ε, a sind reguläre Ausdrücke für alle $a \in \Sigma$
2. Wenn r und s reguläre Ausdrücke über Σ sind, dann auch $(r + s)$, (rs) und r^*

Beispiele für $\Sigma = \{0,1\}$

`0` `1` `(0+1)` `(0+1)*` `(1(0+1))`

`(0+1)*(1(0+1))`

Verzicht auf überflüssige Klammern

- `0+1` statt `(0+1)` und `10` statt `(10)`
- Legen Prioritäten fest:
  r^* vor (rs) vor $(r + s)$

➢ `(0+1)*` ist verschieden von `0+1*`

Reguläre Ausdrücke - Interpretation

- Jeder reguläre Ausdruck definiert eine formale Sprache:
  die Menge aller Wörter, die „auf den Ausdruck passen“.
- ***Intuitiv:***

  $(r + s)$ steht für Vereinigung von Mengen matchender Wörter
  (rs) steht für Konkatenation der matchenden Wörter
  r^* steht für beliebige häufige Wiederholung matchender Wörter
- Eigentlich werden Operationen auf die zugehörigen formalen Sprachen angewendet

Operationen auf Sprachen

- Seien L, L_1 und L_2 formale Sprachen.
- Dann sind $\boldsymbol{L_1 \cup L_2}$, $\boldsymbol{L_1 \cap L_2}$, $\boldsymbol{L_1 \setminus L_2}$, $\boldsymbol{\bar{L}}$ ebenfalls formale Sprachen.
- **Konkatenation:**

$$\boldsymbol{L_1 \cdot L_2 = \{\, uv \mid u \in L_1, v \in L_2 \,\}}$$

- *Beispiel:* $\{a, aaa\} \cdot \{aa, b\} = \{aaa, aaaaa, ab, aaab\}$

$$= \{a^3, a^5, ab, a^3b\}$$

➢ $\boldsymbol{L_1 \cdot L_2 \neq L_2 \cdot L_1}$ (nicht kommutativ)

Operationen auf Sprachen

- **Potenzen**

  $\boldsymbol{L^0 = \{\varepsilon\}}$

  $\boldsymbol{L^n = L^{n-1} \cdot L}$ für alle $n \geq 1$

➢ $L^1 = L^0 \cdot L = \{\varepsilon\} \cdot L = L$

- *Beispiel:* $\{aa, b\}^2 = \{aaaa, aab, baa, bb\} = \{a^4, a^2b, ba^2, b^2\}$

$$\{aa, b\}^3 = \{aa, b\}^2 \cdot \{aa, b\}$$
$$= \{a^6, a^2ba^2, ba^4, b^2a^2, a^4b, a^2b^2, ba^2b, b^3\}$$

Operationen auf Sprachen

- **Kleene-Hülle** $$L^* = \bigcup_{i \geq 0} L^i$$
- $\{aa, b\}^* = \{\varepsilon, a^2, b, a^4, a^2b, ba^2, b^2, a^6, a^2ba^2, ba^4, b^2a^2, a^4b, a^2b^2, ba^2b, b^3, \dots\}$
- **positive Kleene-Hülle** $$L^+ = \bigcup_{i > 0} L^i$$
  - $L^* = L^+ \cup \{\varepsilon\}$
  - $L^+ = L^* \setminus \{\varepsilon\}$ gilt nur, falls $\varepsilon \notin L$
  - definiert exakt Σ^* und Σ^+ für Alphabete Σ

Reguläre Ausdrücke – Interpretation formal

- **Definition:** Sei Σ ein Alphabet.
  Die **formalen Sprachen regulärer Ausdrücke** über Σ sind rekursiv wie folgt definiert:

1. $L(\emptyset) = \emptyset, \qquad L(\boldsymbol{\varepsilon}) = \{\varepsilon\}, \qquad L(\boldsymbol{a}) = \{a\}$ (für alle $a \in \Sigma$)
2. Sind $\boldsymbol{r}$ und $\boldsymbol{s}$ reguläre Ausdrücke über Σ, dann gilt

$$L(\boldsymbol{r} + \boldsymbol{s}) = L(r) \cup L(s)$$

$$L(\boldsymbol{rs}) = L(r) \cdot L(s)$$

$$L(\boldsymbol{r}^*) = L(r)^*$$

Sprache eines regulären Ausdrucks (*Bsp.*)

$r = (0+1)^*(1(1+0))$

$L(0) = \{0\},\ L(1) = \{1\} \ \rightarrow \ L(0+1) = \{0,1\}$

$L((0+1)^*) = \{0,1\}^*$

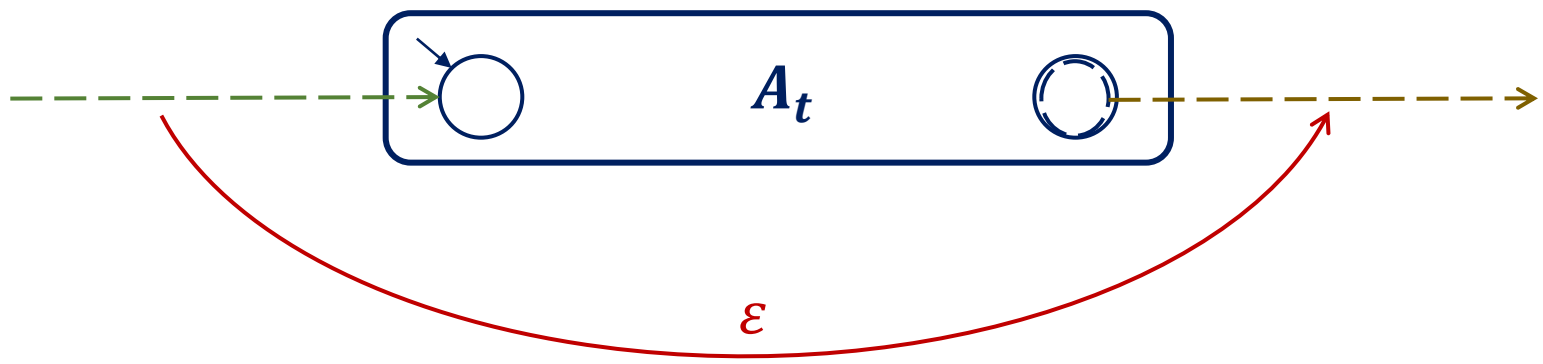
$L(1(0+1)) = \{1\} \cdot \{0,1\} = \{10,11\}$

$L(r) = \{0,1\}^* \cdot \{10,11\}$

$= \{\, w1x \mid w \in \{0,1\}^*, x \in \{0,1\} \,\}$

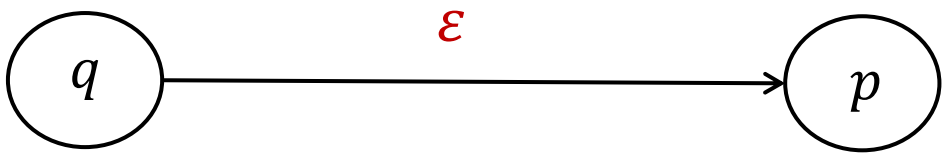
Recab: Reguläre Ausdrücke in der Praxis

- Suchmuster, Tokenisierung
  - Erkennen von Zeichenketten, die auf einen Ausdruck r passen
  - Erkennen von Zeichenketten aus $L(r)$
- *Idee:* Konstruktion eines DEA A_r aus r mit $L(A_r) = L(r)$
- *Herausforderung:* Teilausdrücke t mit *



benötigen z.B. nichtdeterministische Entscheidung, ob A_t durchlaufen oder umgangen wird

NEA mit ε-Übergängen



$$p \in \delta(q, \varepsilon)$$

entspricht „spontanem Zustandswechsel“ von q nach p

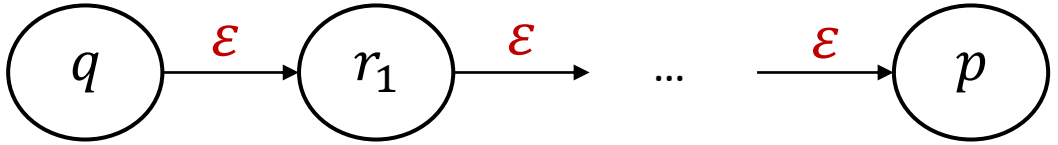
ε-NEA - Definition

- **Definition:**
  Ein **nichtdeterministischer endlicher Automat mit ε-Übergängen (ε-NEA)** ist ein NEA $A = (Q, \Sigma, \delta, q_0, F)$, dessen Überführungsfunktion zu $\boldsymbol{\delta}: \left(\boldsymbol{Q} \times (\boldsymbol{\Sigma} \cup \{\boldsymbol{\varepsilon}\})\right) \rightarrow \boldsymbol{2^Q}$ erweitert ist.

- $\hat{\delta}(q, \varepsilon)$ kann andere Zustände als q enthalten
  - Welche Zustände sind von einem Zustand (nur) mit Hilfe von ε-Übergängen erreichbar?!
  - **ε-Hülle** eines Zustands

ε-Hülle

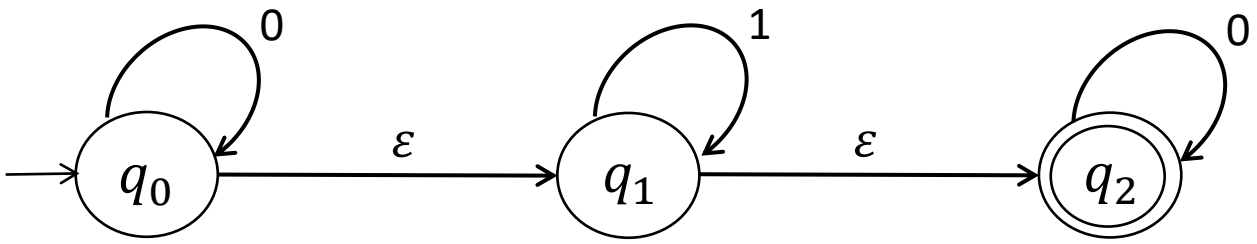
- Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein ε-NEA und $q \in Q$. Die **ε-Hülle von q** ist die Menge aller Zustände p, für die es Zustände $r_0, r_1, \ldots, r_k$ gibt, $k \geq 0$, so dass
  1. $r_0 = q$,
  2. $r_k = p$,
  3. $r_{i+1} \in \delta(r_i, \varepsilon)$ für alle i, $0 \leq i < k$.



- Sie wird mit $\varepsilon\mathrm{H}(q)$ bezeichnet.

➢ Wegen der Option $k = 0$ gilt für jeden Zustand q, dass $q \in \varepsilon\mathrm{H}(q)$.

ε-NEA – Beispiel



$$\varepsilon\mathrm{H}(q_0) = \{q_0, q_1, q_2\}$$
$$\varepsilon\mathrm{H}(q_1) = \{q_1, q_2\}$$
$$\varepsilon\mathrm{H}(q_2) = \{q_2\}$$

$$\delta(q_0, 0) = \{q_0\}$$
$$\delta(q_0, 1) = \emptyset$$
$$\delta(q_0, \varepsilon) = \{q_1\}$$

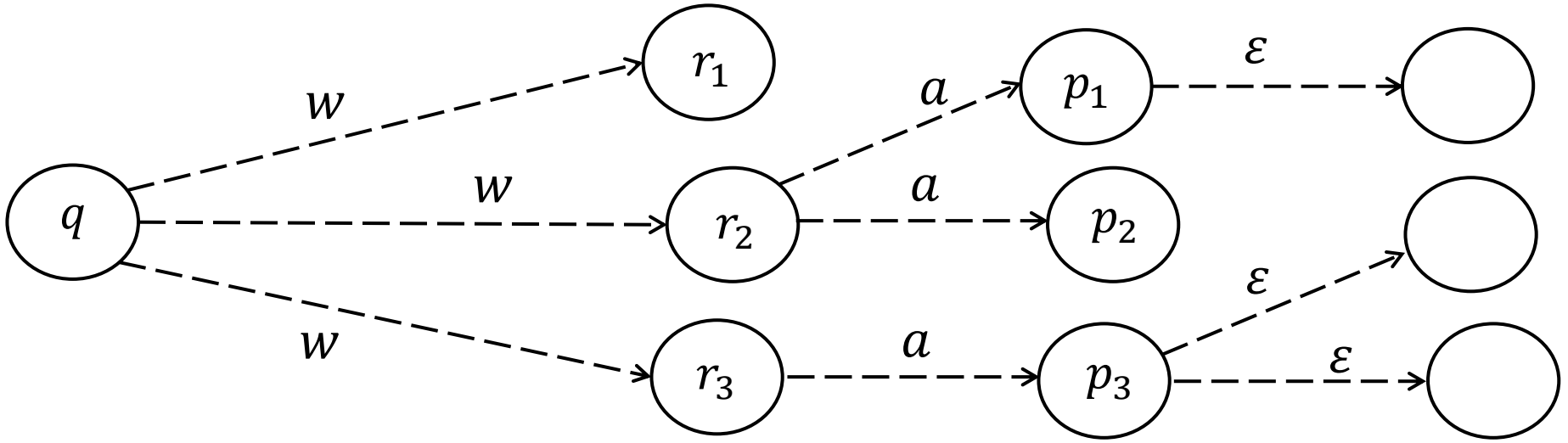
$$\delta(q_1, 0) = \emptyset$$
$$\delta(q_1, 1) = \{q_1\}$$
$$\delta(q_1, \varepsilon) = \{q_2\}$$

$$\delta(q_2, 0) = \{q_2\}$$
$$\delta(q_2, 1) = \emptyset$$
$$\delta(q_2, \varepsilon) = \emptyset$$

ε-NEA – Erweiterte Überführungsfunktion

- Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein ε-NEA und $q \in Q, w \in \Sigma^*, a \in \Sigma$.

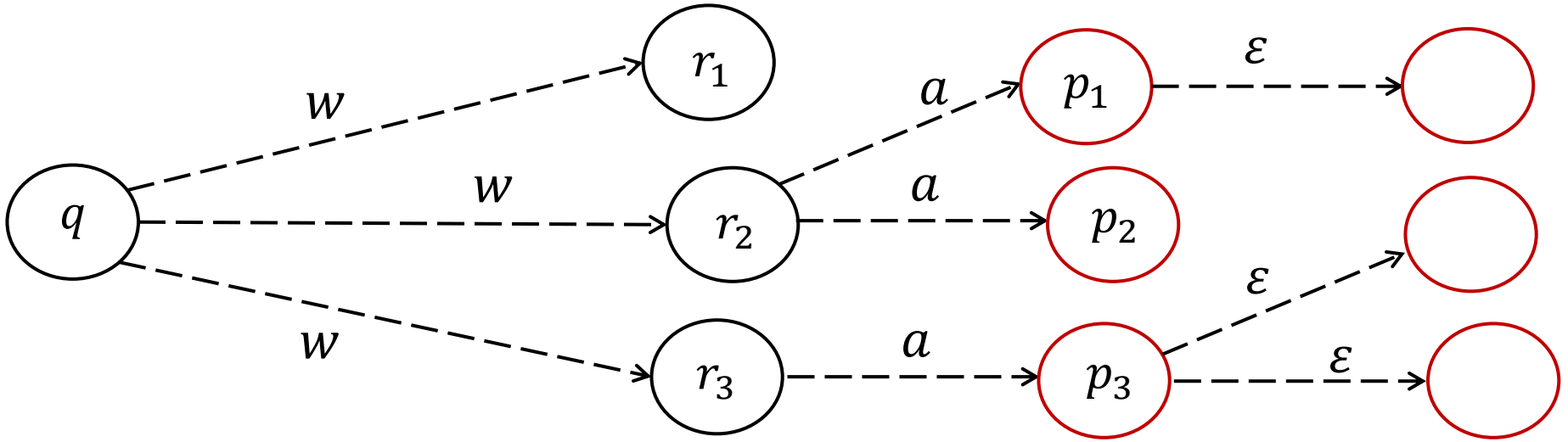
1. $\hat{\delta}(q, \varepsilon) = \varepsilon\mathrm{H}(q)$
2. $\hat{\delta}(q, wa) = ?$



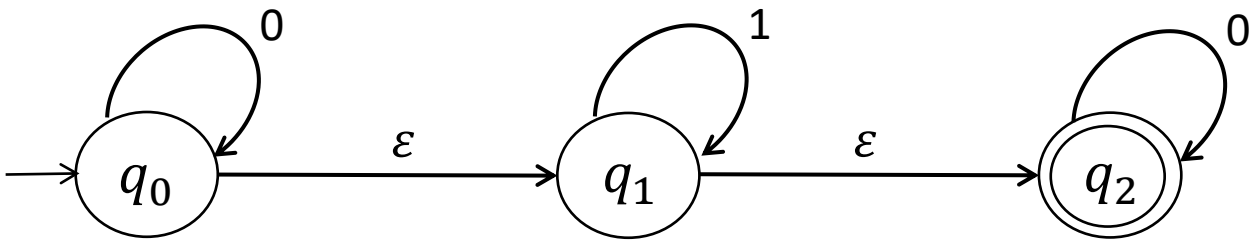
ε-NEA – Erweiterte Überführungsfunktion

- Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein ε-NEA und $q \in Q, w \in \Sigma^*, a \in \Sigma$.

1. $\hat{\delta}(q, \varepsilon) = \varepsilon\mathrm{H}(q)$
2. $\hat{\delta}(q, wa) = \bigcup_{r \in \hat{\delta}(q,w)} \bigcup_{p \in \delta(r,a)} \varepsilon\mathrm{H}(p)$



ε-NEA – Beispiel



$\varepsilon H(q_0) = \{q_0, q_1, q_2\}$
$\varepsilon H(q_1) = \{q_1, q_2\}$
$\varepsilon H(q_2) = \{q_2\}$

$\hat{\delta}(q_0, \varepsilon) = \{q_0, q_1, q_2\}$

$\hat{\delta}(q_0, 0)$: benötigen $\delta(q_0, 0) = \{q_0\}$, $\delta(q_1, 0) = \emptyset$, $\delta(q_2, 0) = \{q_2\}$

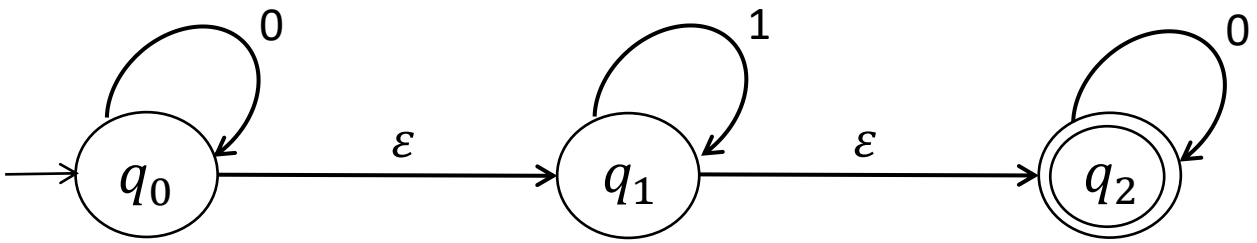
$\hat{\delta}(q_0, 0) = \varepsilon H(q_0) \cup \emptyset \cup \varepsilon H(q_2)$
$= \{q_0, q_1, q_2\}$

$\hat{\delta}(q_0, 01)$: benötigen $\delta(q_0, 1) = \emptyset$, $\delta(q_1, 1) = \{q_1\}$, $\delta(q_2, 1) = \emptyset$

$\hat{\delta}(q_0, 01) = \varepsilon H(q_1) = \{q_1, q_2\}$

ε-NEA – Sprache

Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein ε-NEA. Die von A akzeptierte Sprache ist die Menge $L(A) = \{ w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset \}$.

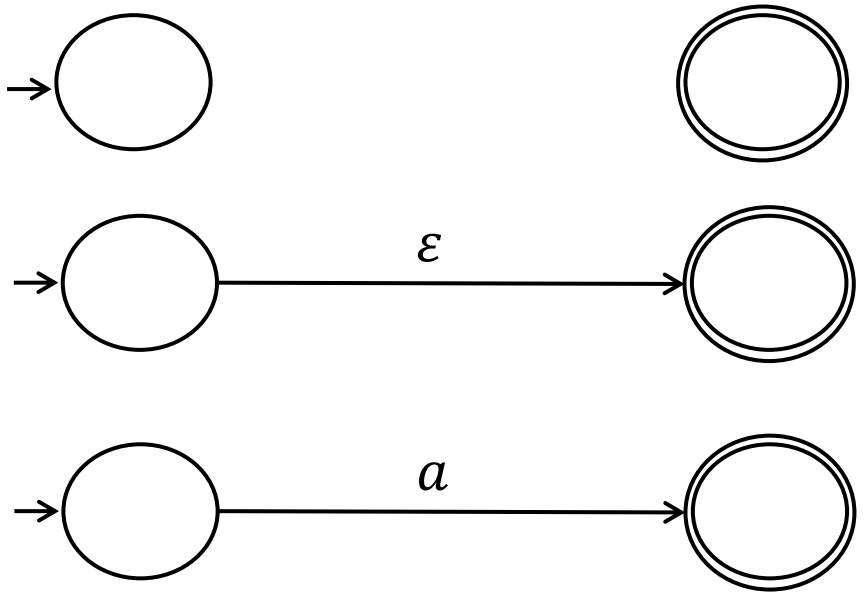


$$L(A) = L(0^*1^*0^*)$$

$$L(A) = \{0^k 1^m 0^n \mid k, m, n \geq 0 \}$$

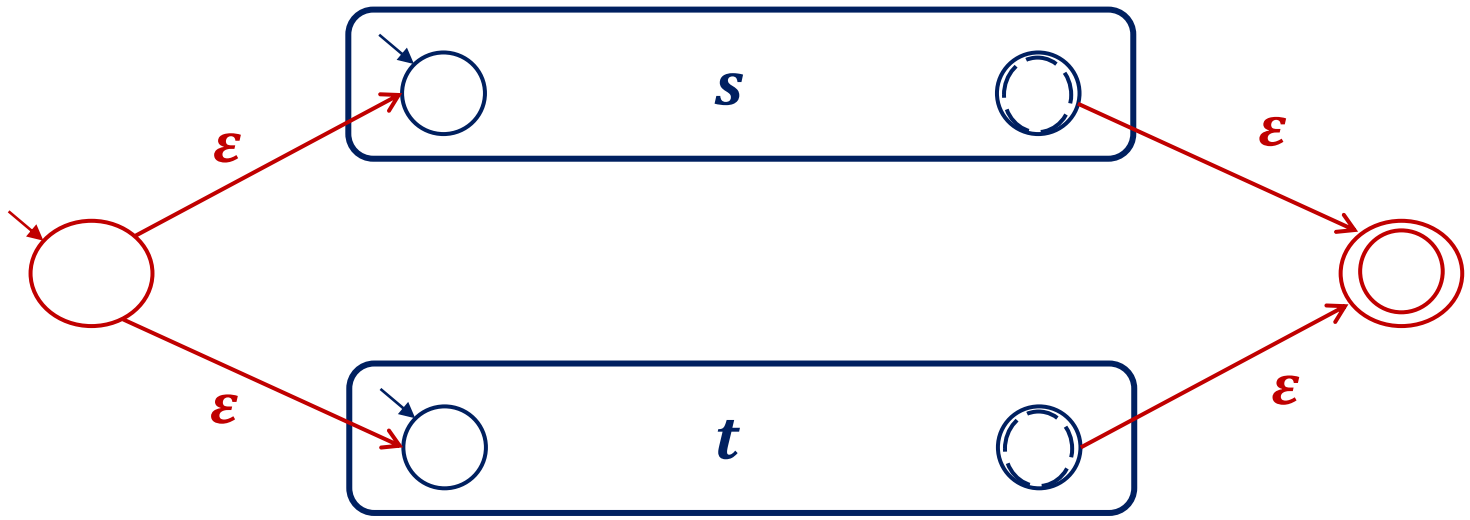
Vom regulären Ausdruck zum ε-NEA

- Gegeben sei ein regulärer Ausdruck r über Σ.
- Konstruieren der rekursiven Definition folgend einen ε-NEA A mit $L(A) = L(r)$, der genau einen akzeptierenden Zustand hat:
  - Falls $r = \emptyset$:
  - Falls $r = \varepsilon$:
  - Falls $r = a \ \ (a \in \Sigma)$:



Vom regulären Ausdruck zum ε-NEA

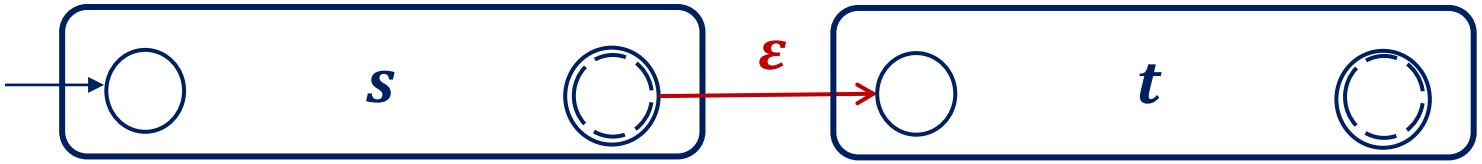
- Falls $r = (s + t)$:



Durch Umbenennung von Zuständen erreicht man, dass die Zustandsmengen der Automaten für s und t disjunkt sind.

Vom regulären Ausdruck zum ε-NEA

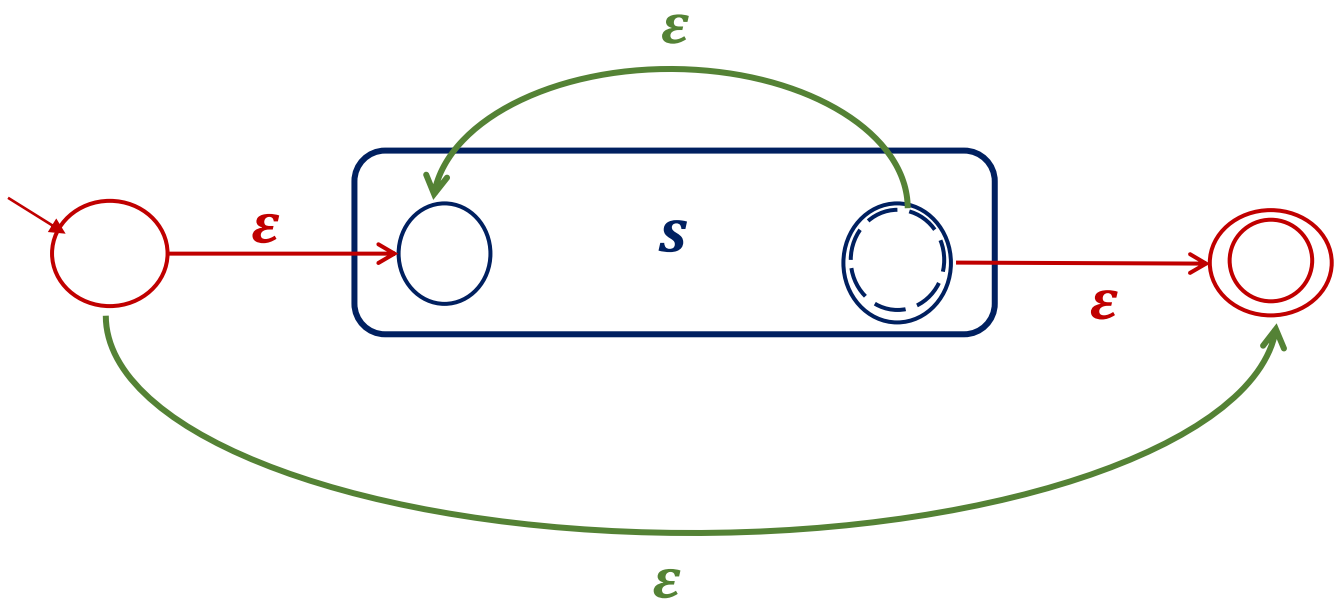
- Falls $r = (st)$:



Durch Umbenennung von Zuständen erreicht man, dass die Zustandsmengen der Automaten für s und t disjunkt sind.

Vom regulären Ausdruck zum ε-NEA

- Falls $r = s^*$:

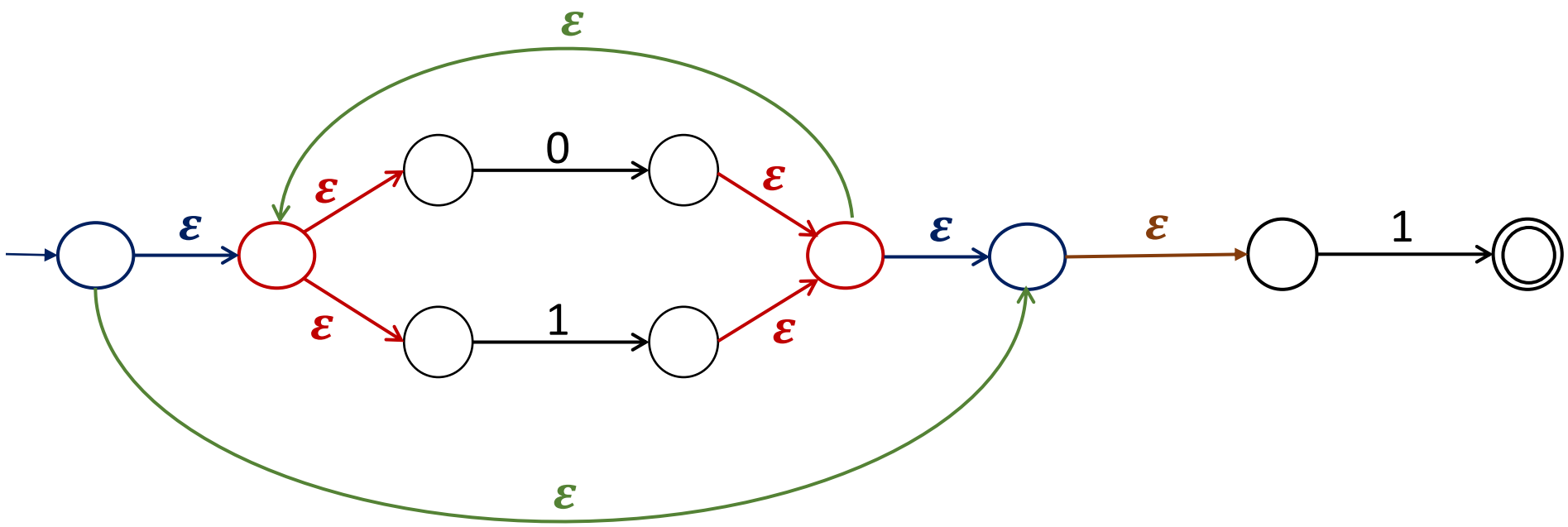


Vom regulären Ausdruck zum ε-NEA

Satz 3.1: Zu jedem regulären Ausdruck r kann ein ε-NEA A konstruiert werden, so dass $L(A) = L(r)$ gilt.

Beweis: Induktion über den strukturellen Aufbau des regulären Ausdrucks entsprechend der Konstruktion ■

Beispiel $(0 + 1)^*1$



Wie geht es weiter?

- *bisher:*
  regulärer Ausdruck ⟶ ε-NEA
- *nächste Vorlesung:*

1. ε-NEA ⟶ NEA
   ⟶ DEA
2. DEA ⟶ regulärer Ausdruck

➢ Alle vier Mechanismen definieren die gleiche Sprachfamilie!