

# Die Thompson-Konstruktion

Gegeben regulärer Ausdruck  $r$ .

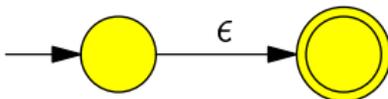
Konstruktion eines NFA  $M$  mit  $L(M) = L(r)$ .

Vorgehen: Induktiv über Aufbau von  $r$ .

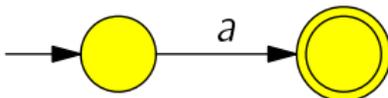
- $r = \emptyset$ :



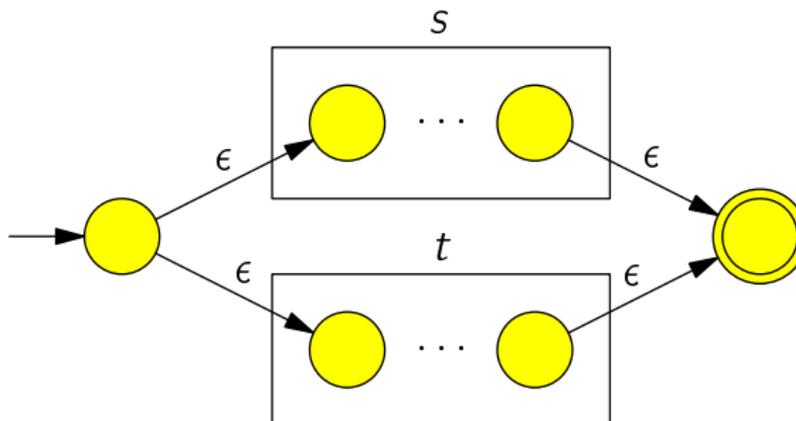
- $r = \epsilon$ :



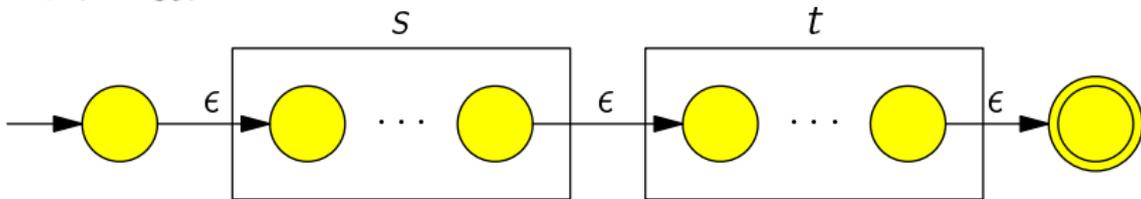
- $r = a$ :



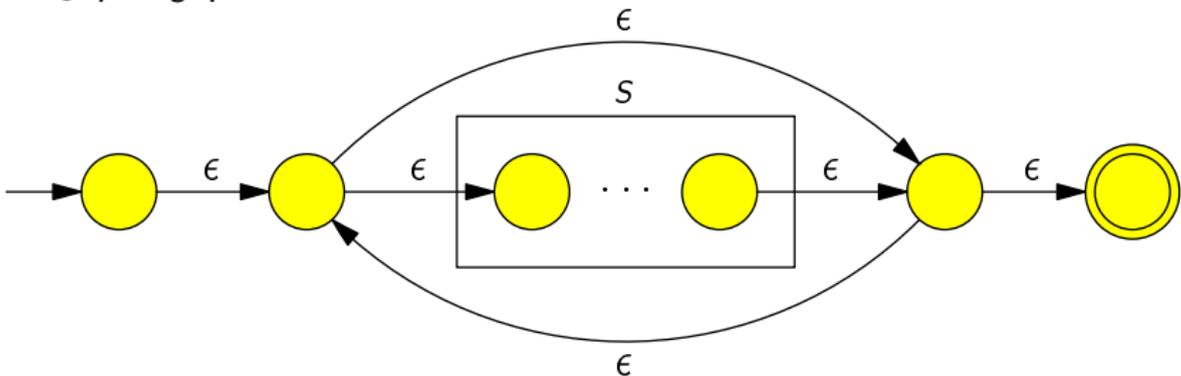
- $r = s + t$ :



- $r = st$ :



- $r = s^*$ :



## Theorem

*Zu jedem regulären Ausdruck  $r$  gibt es einen NFA mit  $\epsilon$ -Kanten  $M$ , so daß  $L(M) = L(r)$ .*

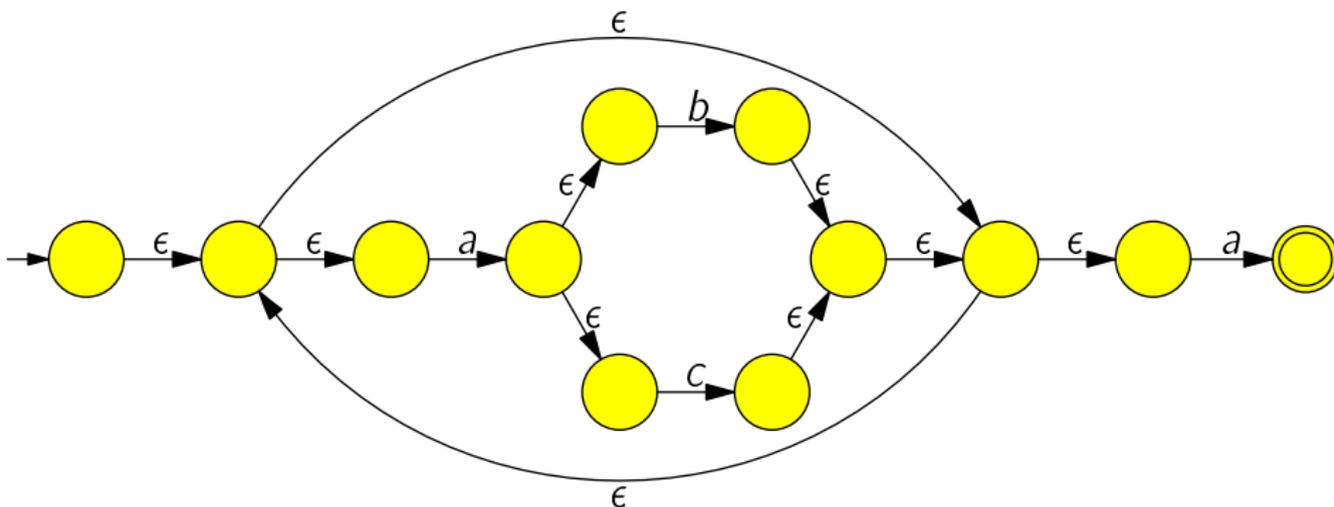
## Beweis.

Thompson-Konstruktion.

Korrektheit:

Strukturelle Induktion über den Aufbau regulärer Ausdrücke.  $\square$

# Beispiel



$$(a(b+c))^*a$$

Größe des NFA linear in der Länge des regulären Ausdrucks! **RWTHAACHEN**

# Robustheit regulärer Sprachen

## Theorem

*DFAs, NFAs, NFAs mit  $\epsilon$ -Übergängen und reguläre Ausdrücke charakterisieren jeweils die regulären Sprachen.*

## Beweis.

- 1 regulärer Ausdruck  $\rightarrow$   $\epsilon$ -NFA: Thompson-Konstruktion
- 2  $\epsilon$ -NFA  $\rightarrow$  NFA: Elimination von  $\epsilon$ -Kanten
- 3 NFA  $\rightarrow$  DFA: Potenzautomat
- 4 DFA  $\rightarrow$  regulärer Ausdruck:  $L_{ij}^k$ -Konstruktion



# Robustheit regulärer Sprachen

## Theorem

*Die Reguläre Sprachen sind abgeschlossen unter Vereinigung, Schnitt, Konkatenation, Kleene'scher Hülle, Komplement, Differenz und Homomorphismen.*

- Vereinigung: Reguläre Ausdrücke
- Schnitt: DFAs, Produktautomat
- Konkatenation: Reguläre Ausdrücke
- Kleene'sche Hülle: Reguläre Ausdrücke
- Komplement: DFAs
- Differenz: Komplement und Schnitt
- Homomorphismen: Reguläre Ausdrücke

## Simulation eines NFA

```
S := { q0 };  
while(es gibt noch ein Zeichen) {  
  c := lese Zeichen;  
  H :=  $\emptyset$ ;  
  for(q in S) { H := H  $\cup$  delta(q, c); }  
  S := H;  
}  
if(S  $\cap$  F  $\neq \emptyset$ ) return 1;  
return 0;
```

Datenstruktur für  $H$ :

- Stack (FIFO-Queue) und
- Bitfeld

Laufzeit:  $O(|Q| \cdot |w|)$ , falls  $|\Sigma|$  konstant.

# Einige Zwischenfragen

Welche Konstruktionen funktionieren auch für NFAs?

- 1 Komplementäutomat **Nein**
- 2 Produktautomat **Ja**
- 3  $L_{ij}^k$ -Konstruktion **Ja**

Wer hat die Nase vorne? NFA oder DFA?

- 1 Vereinigung zweier Sprachen **NFA**
- 2 Schnitt zweier Sprachen **DFA**
- 3 Konstruktion aus einem regulären Ausdruck **NFA**
- 4 Verwandeln in einen regulären Ausdruck **egal**
- 5 Komplementieren **DFA**
- 6 Simulieren **DFA**
- 7 Größe **NFA**