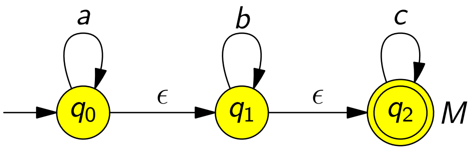
Übersicht

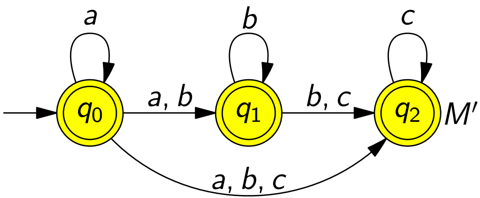
2 Reguläre Sprachen
- 2.1 Reguläre Ausdrücke
- 2.2 Endliche Automaten
- 2.3 Nichtdeterministische endliche Automaten
- 2.4 Die Potenzmengenkonstruktion
- 2.5 NFAs mit ϵ-Übergängen
- 2.6 Minimale DFAs und der Satz von Myhill-Nerode
- 2.7 Berechnung des minimalen DFA
- 2.8 Umwandlung eines Automaten in einen regulären Ausdruck II
- 2.9 Das Pumping-Lemma
- 2.10 Entscheidungsprobleme für reguläre Sprachen

NFAs mit ϵ-Übergängen



Dies ist kein NFA!

Ziel: Erkenne die Sprache $a^* b^* c^*$.



NFA ist komplizierter!

Definition 2.5.1

Ein NFA mit ϵ-Übergängen ist ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$ mit

1. $\delta \colon Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q$,
2. Q, Σ, q_0, F *wie bei NFAs.*

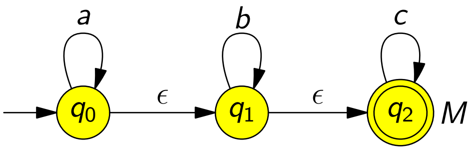
Für $q \in Q$:

$$\epsilon\text{-Hülle}(q) := \{\, q_n \in Q \mid \text{es gibt } q_1, \ldots, q_n \text{ mit } q = q_1 \text{ und } q_{i+1} \in \delta(q_i, \epsilon) \text{ für alle } 1 \le i < n \,\}$$

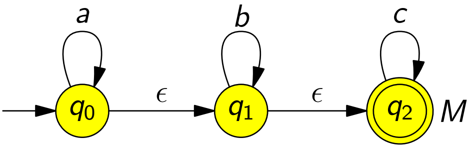
Für $S \subseteq Q$:

$$\epsilon\text{-Hülle}(S) := \bigcup_{q \in S} \epsilon\text{-Hülle}(q)$$

Beispiel



- ϵ-Hülle$(q_0) = \{q_0, q_1, q_2\}$
- ϵ-Hülle$(q_1) = \{q_1, q_2\}$
- ϵ-Hülle$(q_2) = \{q_2\}$
- ϵ-Hülle$(\{q_1, q_2\}) = \{q_1, q_2\}$



Definition 2.5.2

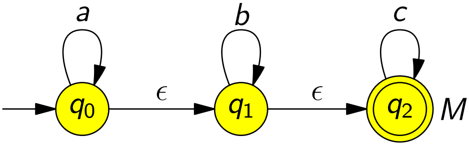
Sei $M = (Q, \Sigma, \delta, q_0, F)$ *ein NFA mit* ϵ*-Übergängen.*
Es sei $q \in Q$, $w \in \Sigma^*$ *und* $a \in \Sigma$.

- $\hat{\delta}(q, \epsilon) = \epsilon\text{-Hülle}(q)$
- $\hat{\delta}(q, w\,a) = \bigcup_{p \in \hat{\delta}(q,w)} \epsilon\text{-Hülle}(\delta(p, a))$

$\hat{\delta}(q, a_1 \ldots a_n)$ sind Zustände, die von q erreichbar sind:

1. Erst über ϵ-Transitionen
2. Dann über a_1-Transition
3. Dann über ϵ-Transitionen
4. Dann über a_2-Transition
5. Dann über ϵ-Transitionen
6. ...

Beispiel



- $\delta(q_0, a) = \{q_0\}$
- $\hat{\delta}(q_0, a) = \{q_0, q_1, q_2\}$
- $\delta(q_0, b) = \varnothing$
- $\hat{\delta}(q_0, b) = \{q_1, q_2\}$
- $\delta(q_0, \epsilon) = \{q_1\}$
- $\hat{\delta}(q_0, \epsilon) = \{q_0, q_1, q_2\}$

- ϵ-Hülle$(q_0) = \{q_0, q_1, q_2\}$
- ϵ-Hülle$(q_1) = \{q_1, q_2\}$

Satz 2.5.3

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein NFA mit ϵ-Übergängen.
Dann gibt es einen NFA M' mit $L(M') = L(M)$.

Beweis.

$M' = (Q, \Sigma, \delta', q_0, F')$ mit

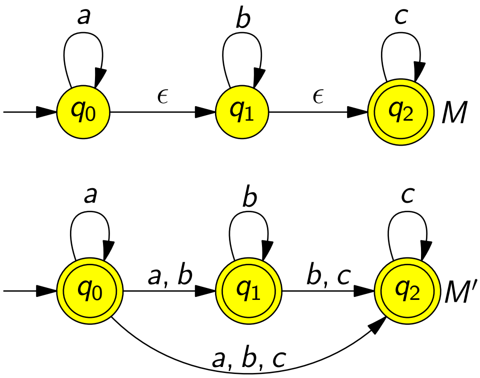
- $F' = \{\, q \in Q \mid \epsilon\text{-Hülle}(q) \cap F \neq \varnothing \,\}$,
- $\delta'(q, a) = \hat{\delta}(q, a)$.

$\square$

Informell:
$p \in \delta'(q, a)$ gdw. in M gibt es Pfad von q nach p, der

1. zunächst mit ϵ beschriftet ist,
2. dann einen a-Übergang hat,
3. dann wieder mit ϵ beschriftet ist.

Beispiel

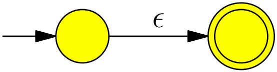


Die Thompson-Konstruktion

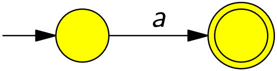
Gegeben regulärer Ausdruck r.
Konstruktion eines NFA M mit ϵ-Übergängen, so dass $L(M) = L(r)$.
Vorgehen: Induktiv über Aufbau von r.

- $r = \emptyset$:

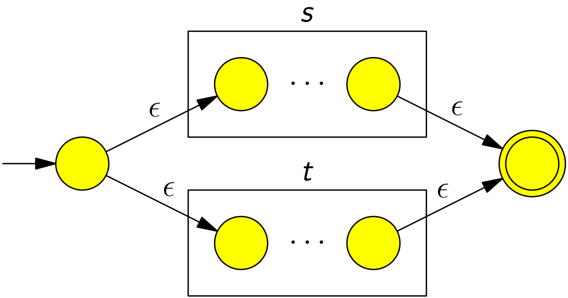
- $r = \epsilon$:



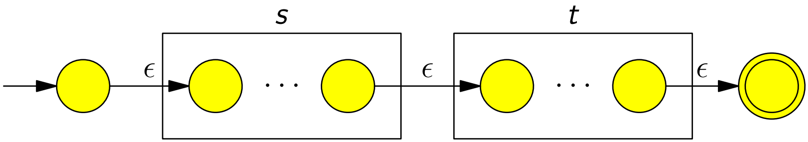
- $r = a$:



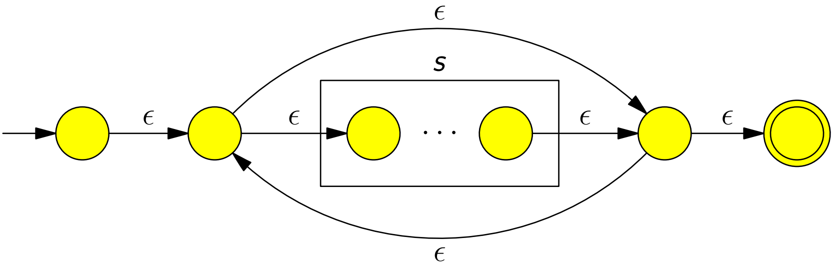
- $r = s + t$:



- $r = s\,t$:



- $r = s^*$:



- $r = (s)$: Automat für s

Satz 2.5.4

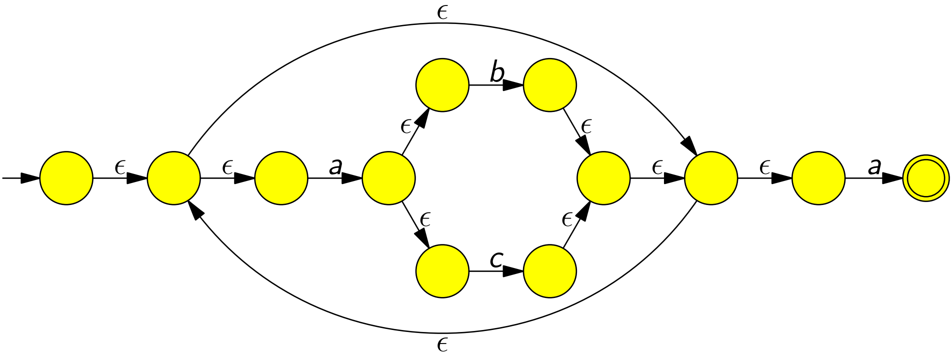
Zu jedem regulären Ausdruck r gibt es einen NFA M mit ϵ-Übergängen, so dass $L(M) = L(r)$.

Beweis.

Thompson-Konstruktion.

Korrektheit:
Strukturelle Induktion über den Aufbau regulärer Ausdrücke. □

Beispiel



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

Größe des NFA linear in der Länge des regulären Ausdrucks!

Robustheit regulärer Sprachen

Satz 2.5.5

DFAs, NFAs, NFAs mit ϵ-Übergängen und reguläre Ausdrücke charakterisieren jeweils die regulären Sprachen.

Beweis.

1. regulärer Ausdruck $\longrightarrow$ ϵ-NFA:
   Thompson-Konstruktion (Satz 2.5.4)
2. ϵ-NFA $\longrightarrow$ NFA: Eliminierung von ϵ-Kanten (Satz 2.5.3)
3. NFA $\longrightarrow$ DFA: Potenzautomat (Satz 2.4.2)
4. DFA $\longrightarrow$ regulärer Ausdruck: L_{ij}^{k}-Konstruktion (Satz 2.2.3)

$\square$

Robustheit regulärer Sprachen

Satz 2.5.6

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

- Vereinigung: Reguläre Ausdrücke (Satz 2.1.6)
- Schnitt: DFAs – Produktautomat (Satz 2.2.6)
- Konkatenation: Reguläre Ausdrücke (Satz 2.1.6)
- Kleene'sche Hülle: Reguläre Ausdrücke (Satz 2.1.6)
- Komplement: DFAs – Komplementärautomat (Satz 2.2.4)
- Differenz: Komplement und Schnitt
- Homomorphismen: Reguläre Ausdrücke (Satz 2.1.6)

Simulation eines NFA

$S := \{q_0\}$;
while (es gibt noch ein Zeichen im Wort w) {
  $c :=$ nächstes Zeichen;
  $H := \varnothing$;
  for (q in S) $\{H := H \cup \delta(q, c);\}$
  $S := H$;
}
if($S \cap F$!= $\varnothing$) **return** true;
else **return** false;

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

Bei DFA: $O(|w|)$

Einige Zwischenfragen

Welche Konstruktionen funktionieren auch für NFAs?

1. Komplementärautomat Nein
2. Produktautomat Ja
3. L_{ij}^k-Konstruktion Ja

Wer ist besser? 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