4a
Reguläre Ausdrücke

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 21. Juli 2025
Basierend auf Folien von PD Dr. David Sabel und Dr. Jan Johannsen

Reguläre Ausdrücke

Informelle Kurzfassung:

- Reguläre Ausdrücke sind (wie Automaten und Grammatiken) ein Formalismus zur Repräsentation von Sprachen.
- Sie bieten eine kompakte Notation für reguläre Sprachen an.
- In der Praxis werden sie verwendet, um Text zu erkennen oder zu suchen.


WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.
OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!
BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!
IT'S HOPELESS!
EVERYBODY STAND BACK.
I KNOW REGULAR EXPRESSIONS.
GRU
TAP TAP

xkcd.com/208/

Reguläre Ausdrücke

Definition

Sei Σ ein Alphabet. Die regulären Ausdrücke über Σ sind definiert durch folgende Regeln (und nur diese):

- $\emptyset$ ist ein regulärer Ausdruck.
- ε ist ein regulärer Ausdruck.
- a mit $a \in \Sigma$ ist ein regulärer Ausdruck.
- Wenn α und β reguläre Ausdrücke sind, dann auch $\alpha\beta$ (alternativ $\alpha \cdot \beta$).
- Wenn α und β reguläre Ausdrücke sind, dann auch $(\alpha|\beta)$.
- Wenn α ein regulärer Ausdruck ist, dann auch $(\alpha)^*$.

Erzeugte Sprache eines regulären Ausdrucks

Definition

Die von einem regulären Ausdruck α erzeugte Sprache $L(\alpha)$ ist rekursiv definiert:

$$
\begin{aligned}
L(\emptyset) &:= \emptyset \\
L(\varepsilon) &:= \{\varepsilon\} \\
L(a) &:= \{a\} \quad \text{für } a \in \Sigma \\
L(\alpha\beta) &:= L(\alpha)L(\beta) = \{uv \mid u \in L(\alpha), v \in L(\beta)\} \\
L(\alpha|\beta) &:= L(\alpha) \cup L(\beta) \\
L((\alpha)^*) &:= L(\alpha)^*
\end{aligned}
$$

Erzeugte Sprache eines regulären Ausdrucks

Definition

Die von einem regulären Ausdruck α erzeugte Sprache $L(\alpha)$ ist rekursiv definiert:

$$
\begin{aligned}
L(\emptyset) &:= \emptyset \\
L(\varepsilon) &:= \{\varepsilon\} \\
L(a) &:= \{a\} \quad \text{für } a \in \Sigma \\
L(\alpha\beta) &:= L(\alpha)L(\beta) = \{uv \mid u \in L(\alpha), v \in L(\beta)\} \\
L(\alpha|\beta) &:= L(\alpha) \cup L(\beta) \\
L((\alpha)^*) &:= L(\alpha)^*
\end{aligned}
$$

Für alle regulären Ausdrücke α, β, γ gilt $L((\alpha|\beta)|\gamma) = L(\alpha|(\beta|\gamma))$.
Daher lassen wir Klammern weg und schreiben $(\alpha_1|\alpha_2|\cdots|\alpha_n)$.

Beispiele für reguläre Ausdrücke

1. $(a|b)^*aa(a|b)^*$
   erzeugt ?
2. $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$
   erzeugt ?
3. $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3))):$
   $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$
   erzeugt ?

Beispiele für reguläre Ausdrücke

1. $(a|b)^*aa(a|b)^*$
   erzeugt alle Wörter über $\{a, b\}$, die zwei aufeinanderfolgende a's enthalten.
2. $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$
   erzeugt ?
3. $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3))):$
   $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$
   erzeugt ?

Beispiele für reguläre Ausdrücke

1. $(a|b)^*aa(a|b)^*$
   erzeugt alle Wörter über $\{a, b\}$, die zwei aufeinanderfolgende a's enthalten.

2. $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$
   erzeugt alle Wörter über $\{a, b, c\}$, die an viertletzter Stelle ein a haben sowie das leere Wort.

3. $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3)))$:
   $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$
   erzeugt ?

Beispiele für reguläre Ausdrücke

1. $(a|b)^*aa(a|b)^*$
   erzeugt alle Wörter über $\{a, b\}$, die zwei aufeinanderfolgende a's enthalten.

2. $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$
   erzeugt alle Wörter über $\{a, b, c\}$, die an viertletzter Stelle ein a haben sowie das leere Wort.

3. $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3))):$
   $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$
   erzeugt alle Wörter über $\{:, 0, 1, \ldots, 9\}$, die Uhrzeiten im 24-Stunden-Format entsprechen.

Beispiele für reguläre Ausdrücke

Geben Sie reguläre Ausdrücke an, welche die folgenden Sprachen erzeugen.

1. alle Wörter über $\{a, b\}$, die das Teilwort *abba* enthalten:
   ?
2. alle Wörter über $\{a, b\}$, die das Teilwort *aba* mindestens zweimal enthalten:
   ?
3. alle Wörter über $\{a, b\}$, die das Teilwort *aaa* nicht enthalten:
   ?
4. alle Wörter einer endlichen Sprache $L = \{w_1, \ldots, w_n\}$ mit $n \geq 1$:
   ?

Beispiele für reguläre Ausdrücke

Geben Sie reguläre Ausdrücke an, welche die folgenden Sprachen erzeugen.

1. alle Wörter über $\{a, b\}$, die das Teilwort *abba* enthalten:
   $(a|b)^* abba (a|b)^*$
2. alle Wörter über $\{a, b\}$, die das Teilwort *aba* mindestens zweimal enthalten:
   ?
3. alle Wörter über $\{a, b\}$, die das Teilwort *aaa* nicht enthalten:
   ?
4. alle Wörter einer endlichen Sprache $L = \{w_1, \ldots, w_n\}$ mit $n \geq 1$:
   ?

Beispiele für reguläre Ausdrücke

Geben Sie reguläre Ausdrücke an, welche die folgenden Sprachen erzeugen.

1. alle Wörter über $\{a, b\}$, die das Teilwort $abba$ enthalten:
   $(a|b)^*abba(a|b)^*$
2. alle Wörter über $\{a, b\}$, die das Teilwort aba mindestens zweimal enthalten:
   $\big((a|b)^*aba(a|b)^*aba(a|b)^* \mid (a|b)^*ababa(a|b)^*\big)$
3. alle Wörter über $\{a, b\}$, die das Teilwort aaa nicht enthalten:
   ?
4. alle Wörter einer endlichen Sprache $L = \{w_1, \ldots, w_n\}$ mit $n \geq 1$:
   ?

Beispiele für reguläre Ausdrücke

Geben Sie reguläre Ausdrücke an, welche die folgenden Sprachen erzeugen.

1. alle Wörter über $\{a, b\}$, die das Teilwort $abba$ enthalten:
   $(a|b)^*abba(a|b)^*$
2. alle Wörter über $\{a, b\}$, die das Teilwort aba mindestens zweimal enthalten:
   $\big((a|b)^*aba(a|b)^*aba(a|b)^* \mid (a|b)^*ababa(a|b)^*\big)$
3. alle Wörter über $\{a, b\}$, die das Teilwort aaa nicht enthalten:
   $(b|ab|aab)^*(\varepsilon|a|aa)$
4. alle Wörter einer endlichen Sprache $L = \{w_1, \ldots, w_n\}$ mit $n \geq 1$:
   ?

Beispiele für reguläre Ausdrücke

Geben Sie reguläre Ausdrücke an, welche die folgenden Sprachen erzeugen.

1. alle Wörter über $\{a, b\}$, die das Teilwort $abba$ enthalten:
   $(a|b)^*abba(a|b)^*$
2. alle Wörter über $\{a, b\}$, die das Teilwort aba mindestens zweimal enthalten:
   $\big((a|b)^*aba(a|b)^*aba(a|b)^* \mid (a|b)^*ababa(a|b)^*\big)$
3. alle Wörter über $\{a, b\}$, die das Teilwort aaa nicht enthalten:
   $(b|ab|aab)^*(\varepsilon|a|aa)$
4. alle Wörter einer endlichen Sprache $L = \{w_1, \ldots, w_n\}$ mit $n \geq 1$:
   $(w_1|\cdots|w_n)$

Regeln zum Vereinfachen von regulären Ausdrücken

Kommutativität:
$(\alpha|\beta) = (\beta|\alpha)$

Neutrale Elemente:
$(\emptyset|\alpha) = (\alpha|\emptyset) = \alpha$

Absorption:
$\emptyset\alpha = \alpha\emptyset = \emptyset$

Distributivität:
$\alpha(\beta|\gamma) = \alpha\beta|\alpha\gamma$

Gesetze über Kleene-Stern:
$((\alpha)^*)^* = (\alpha)^* \qquad (\emptyset)^* = \varepsilon \qquad (\varepsilon)^* = \varepsilon$

Sprache von regulären Ausdrücken

Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis

- $\subseteq$ Wir zeigen, dass die erzeugte Sprache eines regulären Ausdrucks regulär ist.

Sprache von regulären Ausdrücken

Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis

$\subseteq$ Wir zeigen, dass die erzeugte Sprache eines regulären Ausdrucks regulär ist.

Wir konstruieren für einen regulären Ausdruck α einen NFA M_α mit ε-Übergängen und eindeutigen Start- und Endzuständen, sodass $L(M_\alpha) = L(\alpha)$.

Sprache von regulären Ausdrücken

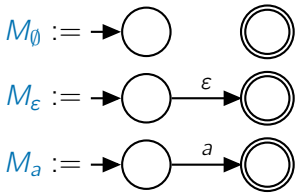
Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis

$\subseteq$ Wir zeigen, dass die erzeugte Sprache eines regulären Ausdrucks regulär ist.

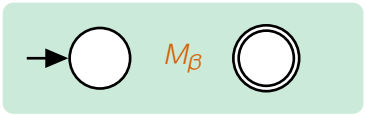
Wir konstruieren für einen regulären Ausdruck α einen NFA M_α mit ε-Übergängen und eindeutigen Start- und Endzuständen, sodass $L(M_\alpha) = L(\alpha)$.



Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

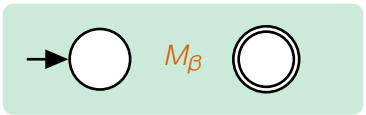
Seien



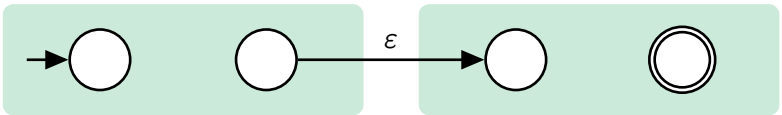
Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Seien



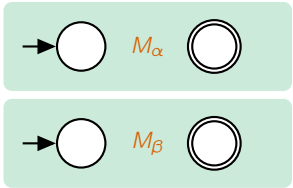
$M_{\alpha\beta} :=$



Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Seien

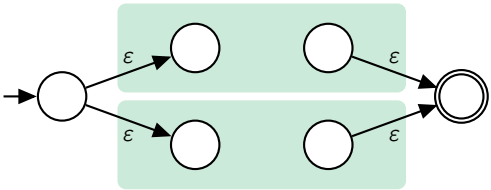


Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Seien

$M_{(\alpha|\beta)} :=$



Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

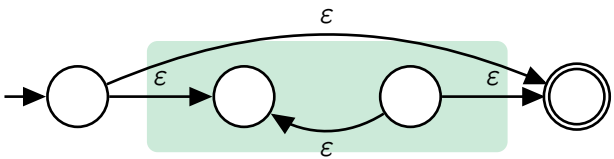
Sei

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei

$M_{(\alpha)^*} :=$



Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

NFA mit ε-Übergängen:

Beispiel für den NFA zu einem regulären Ausdruck

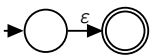
Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

NFA mit ε-Übergängen:

Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

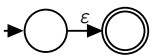
NFA mit ε-Übergängen:

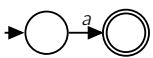


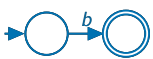
Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

NFA mit ε-Übergängen:



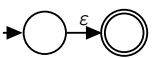


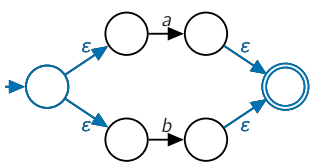


Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

NFA mit ε-Übergängen:

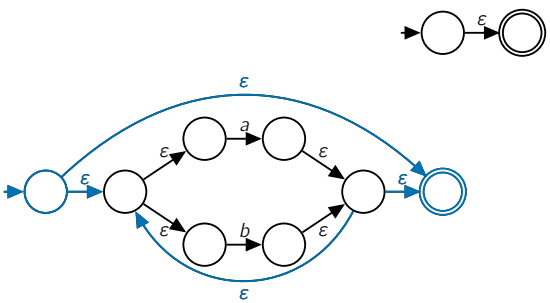




Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

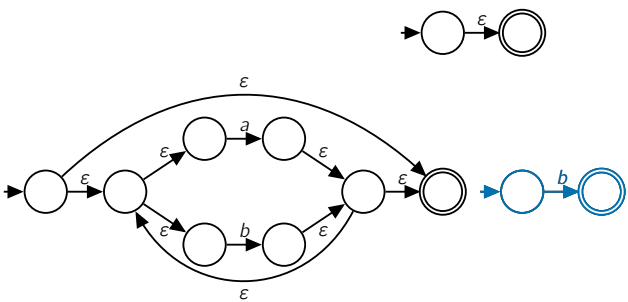
NFA mit ε-Übergängen:



Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

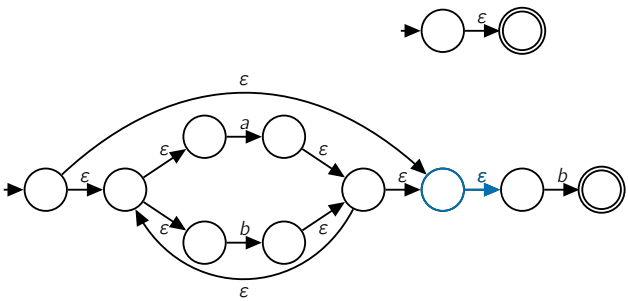
NFA mit ε-Übergängen:



Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

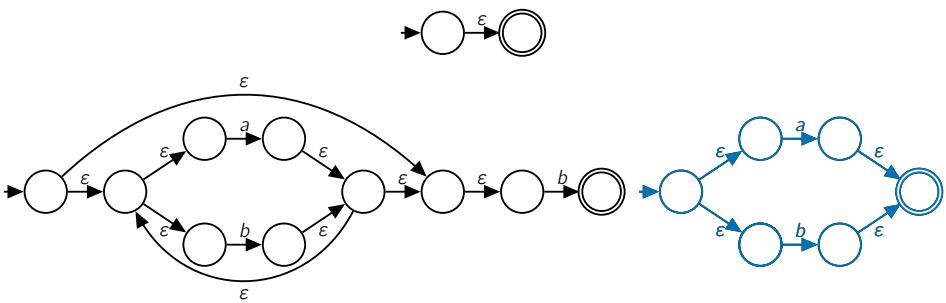
NFA mit ε-Übergängen:



Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

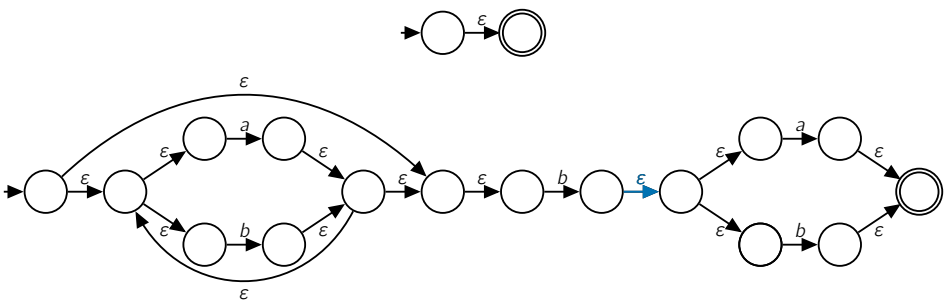
NFA mit ε-Übergängen:



Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

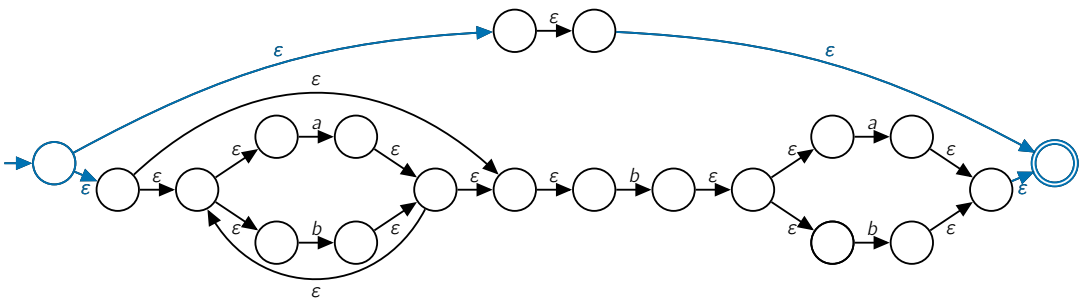
NFA mit ε-Übergängen:



Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

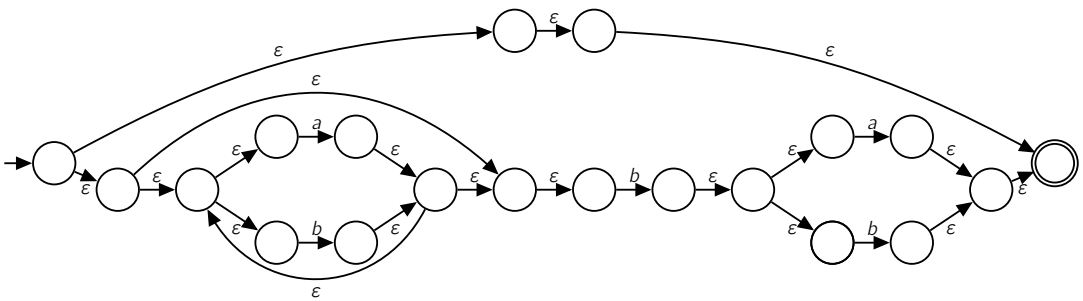
NFA mit ε-Übergängen:



Beispiel für den NFA zu einem regulären Ausdruck

Regulärer Ausdruck:
$(\varepsilon|(a|b)^*b(a|b))$

NFA mit ε-Übergängen:



Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(M_\alpha) = L(\alpha)$ für jeden regulären Ausdruck α gilt. Da $L(M_\alpha)$ regulär ist, heißt es dann, dass $L(\alpha)$ auch regulär ist.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(M_\alpha) = L(\alpha)$ für jeden regulären Ausdruck α gilt. Da $L(M_\alpha)$ regulär ist, heißt es dann, dass $L(\alpha)$ auch regulär ist.

Induktion über die Größe von α.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(M_\alpha) = L(\alpha)$ für jeden regulären Ausdruck α gilt. Da $L(M_\alpha)$ regulär ist, heißt es dann, dass $L(\alpha)$ auch regulär ist.

Induktion über die Größe von α.

- Fall α ist $\emptyset$, ε oder a: Offensichtlich.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(M_\alpha) = L(\alpha)$ für jeden regulären Ausdruck α gilt. Da $L(M_\alpha)$ regulär ist, heißt es dann, dass $L(\alpha)$ auch regulär ist.

Induktion über die Größe von α.

- Fall α ist $\emptyset$, ε oder a: Offensichtlich.
- Fall α ist $\beta\gamma$: Die Induktionshypothese liefert $L(M_\beta) = L(\beta)$ und $L(M_\gamma) = L(\gamma)$. Die Konstruktion von $M_{\beta\gamma}$ sicherstellt, dass $L(M_{\beta\gamma}) = L(M_\beta)L(M_\gamma)$. Daher $L(M_{\beta\gamma}) = L(M_\beta)L(M_\gamma) = L(\beta)L(\gamma) = L(\beta\gamma)$.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(M_\alpha) = L(\alpha)$ für jeden regulären Ausdruck α gilt. Da $L(M_\alpha)$ regulär ist, heißt es dann, dass $L(\alpha)$ auch regulär ist.

Induktion über die Größe von α.

- Fall α ist $\emptyset$, ε oder a: Offensichtlich.
- Fall α ist $\beta\gamma$: Die Induktionshypothese liefert $L(M_\beta) = L(\beta)$ und $L(M_\gamma) = L(\gamma)$.
  Die Konstruktion von $M_{\beta\gamma}$ sicherstellt, dass $L(M_{\beta\gamma}) = L(M_\beta)L(M_\gamma)$.
  Daher $L(M_{\beta\gamma}) = L(M_\beta)L(M_\gamma) = L(\beta)L(\gamma) = L(\beta\gamma)$.
- Fall α ist $(\beta|\gamma)$: Die Induktionshypothese liefert $L(M_\beta) = L(\beta)$ und $L(M_\gamma) = L(\gamma)$.
  Die Konstruktion von $M_{\beta\gamma}$ sicherstellt, dass $L(M_{(\beta|\gamma)}) = L(M_\beta) \cup L(M_\gamma)$.
  Daher $L(M_{(\beta|\gamma)}) = L(M_\beta) \cup L(M_\beta) = L(\beta) \cup L(\gamma) = L(\beta|\gamma)$.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(M_\alpha) = L(\alpha)$ für jeden regulären Ausdruck α gilt. Da $L(M_\alpha)$ regulär ist, heißt es dann, dass $L(\alpha)$ auch regulär ist.

Induktion über die Größe von α.

- Fall α ist $\emptyset$, ε oder a: Offensichtlich.
- Fall α ist $\beta\gamma$: Die Induktionshypothese liefert $L(M_\beta) = L(\beta)$ und $L(M_\gamma) = L(\gamma)$.
  Die Konstruktion von $M_{\beta\gamma}$ sicherstellt, dass $L(M_{\beta\gamma}) = L(M_\beta)L(M_\gamma)$.
  Daher $L(M_{\beta\gamma}) = L(M_\beta)L(M_\gamma) = L(\beta)L(\gamma) = L(\beta\gamma)$.
- Fall α ist $(\beta|\gamma)$: Die Induktionshypothese liefert $L(M_\beta) = L(\beta)$ und $L(M_\gamma) = L(\gamma)$.
  Die Konstruktion von $M_{\beta\gamma}$ sicherstellt, dass $L(M_{(\beta|\gamma)}) = L(M_\beta) \cup L(M_\gamma)$.
  Daher $L(M_{(\beta|\gamma)}) = L(M_\beta) \cup L(M_\beta) = L(\beta) \cup L(\gamma) = L(\beta|\gamma)$.
- Fall α ist $(\beta)^*$: Die Induktionshypothese liefert $L(M_\beta) = L(\beta)$.
  Die Konstruktion von $M_{(\beta)^*}$ sicherstellt, dass $L(M_{(\beta)^*}) = L(M_\beta)^*$.
  Daher $L(M_{(\beta)^*}) = L(M_\beta)^* = L(\beta)^* = L((\beta)^*)$.

□

Sprache von regulären Ausdrücken

Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis (Fortsetzung)

- $\supseteq$ Für jede reguläre Sprache L gibt es einen regulären Ausdruck α mit $L(\alpha) = L$.

Sprache von regulären Ausdrücken

Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis (Fortsetzung)

$\supseteq$ Für jede reguläre Sprache L gibt es einen regulären Ausdruck α mit $L(\alpha) = L$.
Sei DFA $M = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, E)$ mit $L(M) = L$.

Sprache von regulären Ausdrücken

Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis (Fortsetzung)

- $\supseteq$ Für jede reguläre Sprache L gibt es einen regulären Ausdruck α mit $L(\alpha) = L$.
  Sei DFA $M = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, E)$ mit $L(M) = L$.
  Wir konstruieren für M einen regulären Ausdruck α, sodass $L(\alpha) = L(M) = L$.

Sprache von regulären Ausdrücken

Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis (Fortsetzung)

$\supseteq$ Für jede reguläre Sprache L gibt es einen regulären Ausdruck α mit $L(\alpha) = L$.

Sei DFA $M = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, E)$ mit $L(M) = L$.

Wir konstruieren für M einen regulären Ausdruck α, sodass $L(\alpha) = L(M) = L$.

Hilfsmittel: $\alpha_{i,j}^k$ erzeugt die Wörter, die von Zustand z_i zu Zustand z_j führen, ohne dabei Zwischenzustände mit Index größer als k zu nutzen.

Sprache von regulären Ausdrücken

Theorem (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis (Fortsetzung)

$\supseteq$ Für jede reguläre Sprache L gibt es einen regulären Ausdruck α mit $L(\alpha) = L$.

Sei DFA $M = (\{z_1, \ldots, z_n\}, \Sigma, \delta, z_1, E)$ mit $L(M) = L$.

Wir konstruieren für M einen regulären Ausdruck α, sodass $L(\alpha) = L(M) = L$.

Hilfsmittel: $\alpha_{i,j}^k$ erzeugt die Wörter, die von Zustand z_i zu Zustand z_j führen, ohne dabei Zwischenzustände mit Index größer als k zu nutzen.

Daher

$$\alpha := \begin{cases} \emptyset & \text{falls } E = \emptyset \\ (\alpha_{1,j_1}^n | \cdots | \alpha_{1,j_m}^n) & \text{falls } E = \{z_{j_1}, \ldots, z_{j_m}\} \end{cases}$$

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Zur Erinnerung: $\alpha_{i,j}^{k}$ erzeugt die Wörter, die von Zustand z_i zu Zustand z_j führen, ohne dabei Zwischenzustände mit Index größer als k zu nutzen.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Zur Erinnerung: $\alpha_{i,j}^k$ erzeugt die Wörter, die von Zustand z_i zu Zustand z_j führen, ohne dabei Zwischenzustände mit Index größer als k zu nutzen.

Wir müssen $\alpha_{i,j}^k$ konstruieren. Sei $\{a_1, \ldots, a_q\} = \{a \in \Sigma \mid \delta(z_i, a) = z_j\}$.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Zur Erinnerung: $\alpha_{i,j}^k$ erzeugt die Wörter, die von Zustand z_i zu Zustand z_j führen, ohne dabei Zwischenzustände mit Index größer als k zu nutzen.

Wir müssen $\alpha_{i,j}^k$ konstruieren. Sei $\{a_1, \ldots, a_q\} = \{a \in \Sigma \mid \delta(z_i, a) = z_j\}$.

$$\alpha_{i,j}^0 := \begin{cases} \emptyset & \text{falls } i \neq j \text{ und } q = 0 \\ (a_1 | \cdots | a_q) & \text{falls } i \neq j \text{ und } q > 0 \\ (\varepsilon | a_1 | \cdots | a_q) & \text{falls } i = j \end{cases}$$

$$\alpha_{i,j}^{k+1} := (\alpha_{i,j}^k \mid \alpha_{i,k+1}^k (\alpha_{k+1,k+1}^k)^* \alpha_{k+1,j}^k)$$

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Zur Erinnerung: $\alpha_{i,j}^k$ erzeugt die Wörter, die von Zustand z_i zu Zustand z_j führen, ohne dabei Zwischenzustände mit Index größer als k zu nutzen.

Wir müssen $\alpha_{i,j}^k$ konstruieren. Sei $\{a_1, \ldots, a_q\} = \{a \in \Sigma \mid \delta(z_i, a) = z_j\}$.

$$\alpha_{i,j}^0 := \begin{cases} \emptyset & \text{falls } i \neq j \text{ und } q = 0 \\ (a_1 | \cdots | a_q) & \text{falls } i \neq j \text{ und } q > 0 \\ (\varepsilon | a_1 | \cdots | a_q) & \text{falls } i = j \end{cases}$$

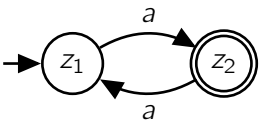
$$\alpha_{i,j}^{k+1} := (\alpha_{i,j}^k \mid \alpha_{i,k+1}^k (\alpha_{k+1,k+1}^k)^* \alpha_{k+1,j}^k)$$

Denn entweder läuft M ohne Zustand z_{k+1} zu besuchen (vgl. $\alpha_{i,j}^k$) oder der „Lauf“ kann in drei Teile gespalten werden:

1. „Lauf“ von z_i bis zum ersten Besuch des Zustands z_{k+1} (vgl. $\alpha_{i,k+1}^k$)
2. mehrmaliges, zyklisches Besuchen von $k+1$ (vgl. $\alpha_{k+1,k+1}^k$)
3. letztmaliges Verlassen von z_{k+1} und „Lauf“ bis zu z_j (vgl. $\alpha_{k+1,j}^k$).

Beispiel für den regulären Ausdruck zu einem DFA

DFA:



Regulärer Ausdruck dazu:

$$\begin{aligned}\alpha^2_{1,2} &= (\alpha^1_{1,2}|\alpha^1_{1,2}(\alpha^1_{2,2})^*\alpha^1_{2,2}) \\ &= \big((a|\varepsilon(\varepsilon)^*a)\,|\,(a|\varepsilon(\varepsilon)^*a)\,(\varepsilon|a(\varepsilon)^*a)^*\,(\varepsilon|a(\varepsilon)^*a)\big)\end{aligned}$$

denn

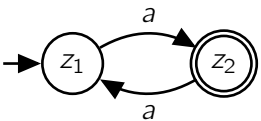
$$\alpha^1_{1,2} = (\alpha^0_{1,2}|\alpha^0_{1,1}(\alpha^0_{1,1})^*\alpha^0_{1,2}) = (a|\varepsilon(\varepsilon)^*a)$$

$$\alpha^1_{2,2} = (\alpha^0_{2,2}|\alpha^0_{2,1}(\alpha^0_{1,1})^*\alpha^0_{1,2}) = (\varepsilon|a(\varepsilon)^*a)$$

$$\alpha^0_{1,1} = \varepsilon \qquad \alpha^0_{2,2} = \varepsilon \qquad \alpha^0_{1,2} = a \qquad \alpha^0_{2,1} = a$$

Beispiel für den regulären Ausdruck zu einem DFA

DFA:



Regulärer Ausdruck dazu:

$$\begin{aligned}\alpha^2_{1,2} &= (\alpha^1_{1,2}|\alpha^1_{1,2}(\alpha^1_{2,2})^*\alpha^1_{2,2}) \\ &= \big((a|\varepsilon(\varepsilon)^*a)\,|\,(a|\varepsilon(\varepsilon)^*a)\,(\varepsilon|a(\varepsilon)^*a)^*\,(\varepsilon|a(\varepsilon)^*a)\big) \\ &\quad \text{äquivalent zu } a(aa)^* \text{ (durch Vereinfachung)}\end{aligned}$$

denn

$$\alpha^1_{1,2} = (\alpha^0_{1,2}|\alpha^0_{1,1}(\alpha^0_{1,1})^*\alpha^0_{1,2}) = (a|\varepsilon(\varepsilon)^*a)$$

$$\alpha^1_{2,2} = (\alpha^0_{2,2}|\alpha^0_{2,1}(\alpha^0_{1,1})^*\alpha^0_{1,2}) = (\varepsilon|a(\varepsilon)^*a)$$

$$\alpha^0_{1,1} = \varepsilon \qquad \alpha^0_{2,2} = \varepsilon \qquad \alpha^0_{1,2} = a \qquad \alpha^0_{2,1} = a$$

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(\alpha) = L(M)$.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(\alpha) = L(M)$.

Für $w \in \Sigma^*$ und z_i, z_j mit $\widetilde{\delta}(z_i, w) = z_j$ sei $visit_i(w) = q_1, \ldots, q_m$ die Folge der besuchten Zustände (wobei $q_1 = z_i$ und $q_m = z_j$).

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(\alpha) = L(M)$.

Für $w \in \Sigma^*$ und z_i, z_j mit $\widetilde{\delta}(z_i, w) = z_j$ sei $visit_i(w) = q_1, \ldots, q_m$ die Folge der besuchten Zustände (wobei $q_1 = z_i$ und $q_m = z_j$).

Wir definieren:

$$L_{i,j}^k = \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \widetilde{\delta}(z_i, w) = z_j \text{ und } visit_i(w) = q_1, \ldots, q_m, \\ \text{sodass für } 1 < l < m\text{: wenn } q_l = z_p \text{ dann } p \leq k \end{array} \right\}$$

$L_{i,j}^k$ enthält die Wörter, die von Zustand z_i zu Zustand z_j führen ohne dabei Zwischenzustände mit Index größer als k zu benutzen.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass $L(\alpha) = L(M)$.

Für $w \in \Sigma^*$ und z_i, z_j mit $\widetilde{\delta}(z_i, w) = z_j$ sei $visit_i(w) = q_1, \ldots, q_m$ die Folge der besuchten Zustände (wobei $q_1 = z_i$ und $q_m = z_j$).

Wir definieren:

$$L_{i,j}^k = \left\{ w \in \Sigma^* \;\middle|\; \begin{array}{l} \widetilde{\delta}(z_i, w) = z_j \text{ und } visit_i(w) = q_1, \ldots, q_m, \\ \text{sodass für } 1 < l < m\text{: wenn } q_l = z_p \text{ dann } p \leq k \end{array} \right\}$$

$L_{i,j}^k$ enthält die Wörter, die von Zustand z_i zu Zustand z_j führen ohne dabei Zwischenzustände mit Index größer als k zu benutzen.

Mit Induktion über k können wir zeigen, dass $L(\alpha_{i,j}^k) = L_{i,j}^k$ für jedes $i, j, k \in \{1, \ldots, n\}$. Siehe Skript.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei $E = \{z_{j_1}, \ldots, z_{j_m}\}$.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei $E = \{z_{j_1}, \ldots, z_{j_m}\}$.

Falls $m = 0$, dann $L(\alpha) = L(\emptyset) = \emptyset = L(M)$.

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei $E = \{z_{j_1}, \ldots, z_{j_m}\}$.

Falls $m = 0$, dann $L(\alpha) = L(\emptyset) = \emptyset = L(M)$.

Falls $m > 0$, dann

$L(\alpha)$

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei $E = \{z_{j_1}, \ldots, z_{j_m}\}$.

Falls $m = 0$, dann $L(\alpha) = L(\emptyset) = \emptyset = L(M)$.

Falls $m > 0$, dann

$$\begin{aligned} & L(\alpha) \\ = \; & L(\alpha^n_{1,j_1} | \cdots | \alpha^n_{1,j_m}) \end{aligned}$$

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei $E = \{z_{j_1}, \ldots, z_{j_m}\}$.

Falls $m = 0$, dann $L(\alpha) = L(\emptyset) = \emptyset = L(M)$.

Falls $m > 0$, dann

$$\begin{aligned} & L(\alpha) \\ = \; & L(\alpha^n_{1,j_1} | \cdots | \alpha^n_{1,j_m}) \\ = \; & L(\alpha^n_{1,j_1}) \cup \cdots \cup L(\alpha^n_{1,j_m}) \end{aligned}$$

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei $E = \{z_{j_1}, \ldots, z_{j_m}\}$.

Falls $m = 0$, dann $L(\alpha) = L(\emptyset) = \emptyset = L(M)$.

Falls $m > 0$, dann

$$\begin{aligned} & L(\alpha) \\ = \; & L(\alpha^n_{1,j_1} | \cdots | \alpha^n_{1,j_m}) \\ = \; & L(\alpha^n_{1,j_1}) \cup \cdots \cup L(\alpha^n_{1,j_m}) \\ = \; & L^n_{1,j_1} \cup \cdots \cup L^n_{1,j_m} \end{aligned}$$

Sprache von regulären Ausdrücken

Beweis (Fortsetzung)

Sei $E = \{z_{j_1}, \ldots, z_{j_m}\}$.

Falls $m = 0$, dann $L(\alpha) = L(\emptyset) = \emptyset = L(M)$.

Falls $m > 0$, dann

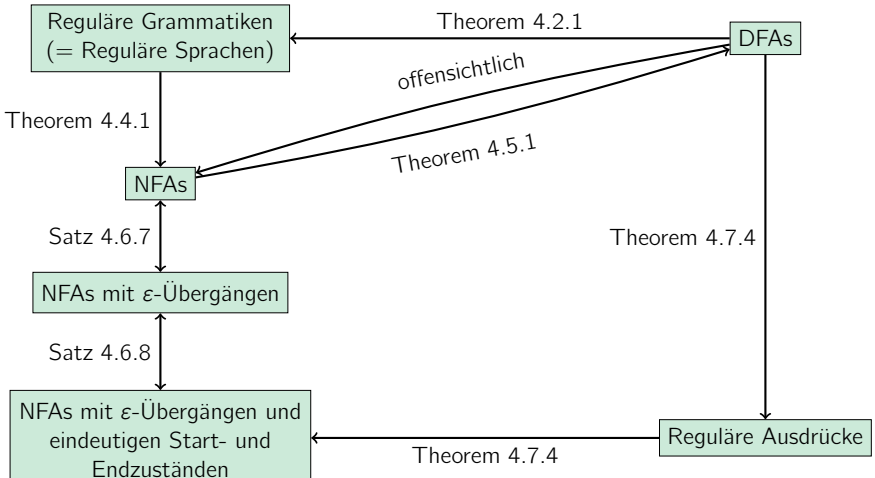
$$
\begin{aligned}
& L(\alpha) \\
&= L(\alpha^n_{1,j_1} | \cdots | \alpha^n_{1,j_m}) \\
&= L(\alpha^n_{1,j_1}) \cup \cdots \cup L(\alpha^n_{1,j_m}) \\
&= L^n_{1,j_1} \cup \cdots \cup L^n_{1,j_m} \\
&= L(M)
\end{aligned}
$$

□

Anwendungen von regulären Ausdrücken

- Reguläre Ausdrücke werden zur lexikalischen Analyse von Programmiersprachen und Domain Specific Languages verwendet, aber auch für Textsuche und -ersetzung in Texteditoren sowie in Befehlszeilenprogramm wie `sed` und AWK.
- Tools wie lex (für C/C++), ANTLR (für Java) und PLY (für Python) generieren lexikalische Analyser („Lexer“) aus regulären Ausdrücken.
- Moderne Programmiersprachen unterstützen reguläre Ausdrücke entweder nativ oder mithilfe einer Bibliothek.

Zusammenfassung der Formalismen für reguläre Sprachen



Kanten ($\rightarrow$) zeigen Kodierungen.