Reguläre Ausdrücke

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik

Stand: 9. Mai 2023
Folien ursprünglich von PD Dr. David Sabel

(Wiederholung für FSK:) NFAs mit ε-Übergängen

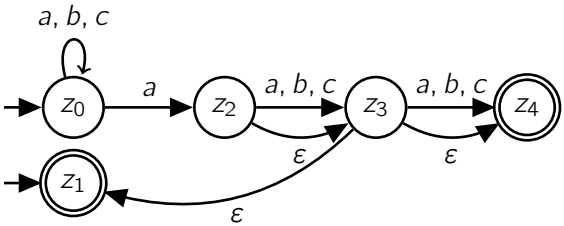
- ε-Übergänge erlauben Zustandswechsel ohne Lesen eines Zeichens (es wird sozusagen das leere Wort ε gelesen).
- Ausdruckskraft ändert sich mit ε-Übergängen nicht.
- ε-Übergänge machen manche Konstruktionen einfacher.

Definition (NFA mit ε-Übergängen)

Ein nichtdeterministischer endlicher Automat mit ε-Übergängen (NFA mit ε-Übergängen) ist ein 5-Tupel $M = (Z, \Sigma, \delta, S, E)$ wobei

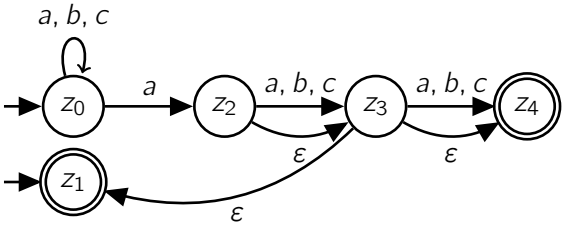
- Z ist eine endliche Menge von Zuständen,
- Σ ist das (endliche) Eingabealphabet mit $Z \cap \Sigma = \emptyset$,
- $\delta : Z \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Z)$ ist die Zustandsüberführungsfunktion,
- $S \subseteq Z$ ist die Menge der Startzustände und
- $E \subseteq Z$ ist die Menge der Endzustände.

(Wiederholung für FSK:) Beispiel: NFA mit ε-Übergängen



Akzeptierte Sprache: ?

(Wiederholung für FSK:) Beispiel: NFA mit ε-Übergängen



Akzeptierte Sprache:

alle Wörter aus $\{a, b, c\}^*$, die an letzter, vorletzter oder drittletzter Postion ein a haben, und das leere Wort

Eindeutige Start- und Endzustände

Satz 4.6.8

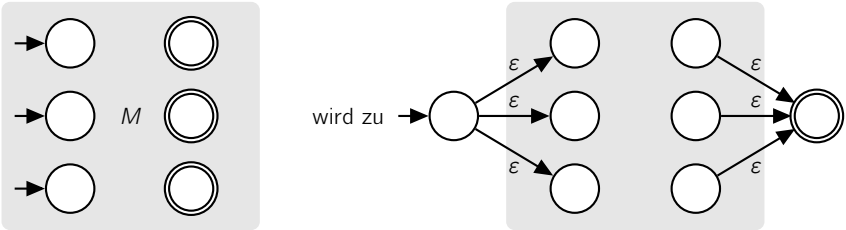
Für jeden NFA M mit ε-Übergängen gibt es einen NFA M' mit ε-Übergängen, sodass $L(M) = L(M')$ und M' genau einen Startzustand und genau einen Endzustand hat, wobei diese beiden Zustände verschieden sind.

Eindeutige Start- und Endzustände

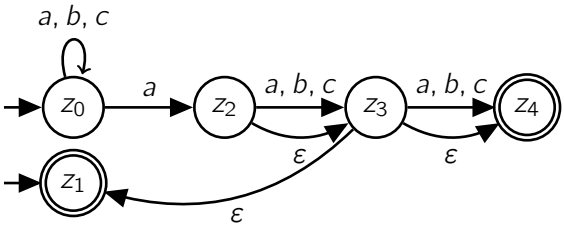
Satz 4.6.8

Für jeden NFA M mit ε-Übergängen gibt es einen NFA M' mit ε-Übergängen, sodass $L(M) = L(M')$ und M' genau einen Startzustand und genau einen Endzustand hat, wobei diese beiden Zustände verschieden sind.

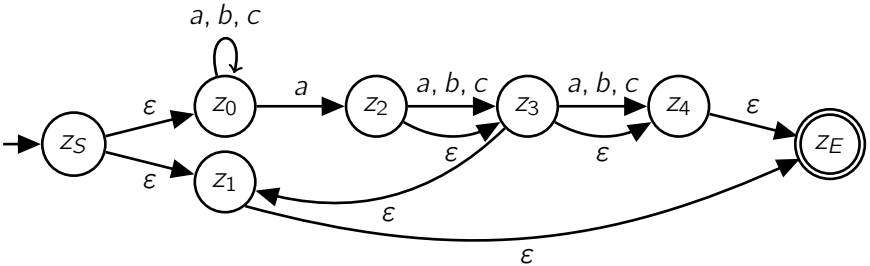
Beweis: Konstruiere M' aus M, durch Hinzufügen eines neuen Start- und eines neuen Endzustand mit ε-Übergängen:



Beispiel



wird zu



Reguläre Ausdrücke

- Reguläre Ausdrücke sind (wie Automaten und Grammatiken) ein Formalismus zur Repräsentation von Sprachen.

- Praktische Verwendung: Regex-Bibliotheken in Programmiersprachen oder bei der Shell-Programmierung zum Suchen und Ersetzen von Zeichenketten (verwenden meist **erweiterte** reguläre Ausdrücke)

- Aufbau regulärer Ausdrücke:
  Basisausdrücke und Operatoren zum Zusammensetzen

Definition (Regulärer Ausdruck)

Sei Σ ein Alphabet. Die regulären Ausdrücke über Σ sind induktiv definiert:

- $\emptyset$ ist ein regulärer Ausdruck
- ε ist ein regulärer Ausdruck
- a mit $a \in \Sigma$ ist ein regulärer Ausdruck
- Wenn α_1 und α_2 reguläre Ausdrücke sind, dann auch $\alpha_1\alpha_2$
- Wenn α_1 und α_2 reguläre Ausdrücke sind, dann auch $(\alpha_1|\alpha_2)$
- Wenn α ein regulärer Ausdruck ist, dann auch $(\alpha)^*$

Reguläre Ausdrücke (3)

Erzeugte Sprache

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

$$
\begin{aligned}
L(\emptyset) &:= \emptyset \\
L(\varepsilon) &:= \{\varepsilon\} \\
L(a) &:= \{a\} \quad \text{für } a \in \Sigma \\
L(\alpha_1\alpha_2) &:= L(\alpha_1)L(\alpha_2) = \{uv \mid u \in L(\alpha_1), v \in L(\alpha_2)\} \\
L(\alpha_1|\alpha_2) &:= L(\alpha_1) \cup L(\alpha_2) \\
L((\alpha)^*) &:= L(\alpha)^*
\end{aligned}
$$

Für alle regulären Ausdrücke $\alpha_1, \alpha_2, \alpha_3$ gilt:

$$L((\alpha_1|\alpha_2)|\alpha_3) = L(\alpha_1|(\alpha_2|\alpha_3))$$

Daher lassen wir Klammern weg und schreiben $(\alpha_1|\alpha_2|\ldots|\alpha_n)$.

Beispiele (1)

- $(a|b)^*aa(a|b)^*$

  erzeugt alle Wörter über $\{a, b\}$, die ?

- $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$

  erzeugt alle Wörter über $\{a, b, c\}$, die ?

- $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3))):$
  $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$

  erzeugt alle Wörter über $\{0, 1, \ldots, 9, :\}$, die ?

- Eine endliche Sprache $S = \{w_1, \ldots, w_n\}$ wird durch ? erzeugt

Beispiele (1)

- $(a|b)^*aa(a|b)^*$

  erzeugt alle Wörter über $\{a, b\}$, die zwei aufeinanderfolgende a's enthalten

- $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$

  erzeugt alle Wörter über $\{a, b, c\}$, die ?

- $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3)))$:
  $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$

  erzeugt alle Wörter über $\{0, 1, \ldots, 9, :\}$, die ?

- Eine endliche Sprache $S = \{w_1, \ldots, w_n\}$ wird durch ? erzeugt

Beispiele (1)

- $(a|b)^*aa(a|b)^*$

  erzeugt alle Wörter über $\{a, b\}$, die zwei aufeinanderfolgende a's enthalten

- $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$

  erzeugt alle Wörter über $\{a, b, c\}$, die an viertletzter Stelle ein a haben sowie das leere Wort

- $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3))):$
  $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$

  erzeugt alle Wörter über $\{0, 1, \ldots, 9, :\}$, die ?

- Eine endliche Sprache $S = \{w_1, \ldots, w_n\}$ wird durch ? erzeugt

Beispiele (1)

- $(a|b)^*aa(a|b)^*$

  erzeugt alle Wörter über $\{a, b\}$, die zwei aufeinanderfolgende a's enthalten

- $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$

  erzeugt alle Wörter über $\{a, b, c\}$, die an viertletzter Stelle ein a haben sowie das leere Wort

- $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3))):$
  $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$

  erzeugt alle Wörter über $\{0, 1, \ldots, 9, :\}$, die Uhrzeiten im 24-Stunden-Format entsprechen

- Eine endliche Sprache $S = \{w_1, \ldots, w_n\}$ wird durch ? erzeugt

Beispiele (1)

- $(a|b)^*aa(a|b)^*$

  erzeugt alle Wörter über $\{a, b\}$, die zwei aufeinanderfolgende a's enthalten

- $(\varepsilon|((a|b|c)^*a(a|b|c)(a|b|c)(a|b|c)))$

  erzeugt alle Wörter über $\{a, b, c\}$, die an viertletzter Stelle ein a haben sowie das leere Wort

- $((0|1|2|3|4|5|6|7|8|9)|1(0|1|2|3|4|5|6|7|8|9)|(2(0|1|2|3))):$
  $((0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9))$

  erzeugt alle Wörter über $\{0, 1, \ldots, 9, :\}$, die Uhrzeiten im 24-Stunden-Format entsprechen

- Eine endliche Sprache $S = \{w_1, \ldots, w_n\}$ wird durch $(w_1|\cdots|w_n)$ erzeugt

Beispiele (2)

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

- L_1 = alle Wörter über $\{a, b\}$, die das Teilwort *abba* enthalten
- L_2 = alle Wörter über $\{a, b\}$, die das Teilwort *aba* mindestens 2 Mal enthalten
- L_3 = alle Wörter über $\{a, b\}$, die das Teilwort *aaa nicht* enthalten

Beispiele (2)

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

- L_1 = alle Wörter über $\{a, b\}$, die das Teilwort *abba* enthalten

$$(a|b)^* abba (a|b)^*$$

- L_2 = alle Wörter über $\{a, b\}$, die das Teilwort *aba* mindestens 2 Mal enthalten

- L_3 = alle Wörter über $\{a, b\}$, die das Teilwort *aaa nicht* enthalten

Beispiele (2)

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

- L_1 = alle Wörter über $\{a, b\}$, die das Teilwort *abba* enthalten

$$(a|b)^* abba (a|b)^*$$

- L_2 = alle Wörter über $\{a, b\}$, die das Teilwort *aba* mindestens 2 Mal enthalten

$$\big((a|b)^* aba (a|b)^* aba (a|b)^* \mid (a|b)^* ababa (a|b)^*\big)$$

- L_3 = alle Wörter über $\{a, b\}$, die das Teilwort *aaa nicht* enthalten

Beispiele (2)

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

- $L_1 =$ alle Wörter über $\{a, b\}$, die das Teilwort *abba* enthalten

$$(a|b)^* abba (a|b)^*$$

- $L_2 =$ alle Wörter über $\{a, b\}$, die das Teilwort *aba* mindestens 2 Mal enthalten

$$\big((a|b)^* aba (a|b)^* aba (a|b)^* \mid (a|b)^* ababa (a|b)^*\big)$$

- $L_3 =$ alle Wörter über $\{a, b\}$, die das Teilwort *aaa nicht* enthalten

$$(b|ab|aab)^* (\varepsilon|a|aa)$$

Beispiel: grep

```
$ grep -E " d(er|ie|as) neue" faust.txt
  Nein, er gefällt mir nicht, der neue Burgemeister!
  Allein der neue Trieb erwacht,
  Da seh' ich auch die neue Wohnung,
  Noch blendet ihn der neue Tag.

$ grep -E "(der|die|das) Q[a-z]*" faust.txt
  Von dem der Quell sich ewig sprudelnd stürzt,
  Vom ganzen Praß die Quintessenz.

$ grep -E "(( )*Gretchen[[:punct:]]*){2,}" faust.txt
  Gretchen!  Gretchen!
```

Satz von Kleene

Theorem 4.7.4 (Satz von Kleene)

Reguläre Ausdrücke erzeugen genau die regulären Sprachen.

Beweis in zwei Teilen:

1. Jede von einem regulären Ausdruck erzeugte Sprache ist regulär.
2. Für jede reguläre Sprache gibt es einen regulären Ausdruck, der sie erzeugt.