Reguläre Ausdrücke

Prof. Dr. David Sabel

LFE Theoretische Informatik

Letzte Änderung der Folien: 16. Mai 2022

Wiederholung: 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 Tupel $M = (Z, \Sigma, \delta, S, E)$ wobei

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

Wiederholung: NFA mit ε-Übergängen: Akzeptierte Sprache

Akzeptierte Sprache eines NFA mit ε-Übergängen

Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA mit ε-Übergängen.
Wir definieren $\widetilde{\delta} : (\mathcal{P}(Z) \times \Sigma^*) \to \mathcal{P}(Z)$ induktiv durch:

$$\begin{aligned} \widetilde{\delta}(X, \varepsilon) &:= X \\ \widetilde{\delta}(X, aw) &:= \bigcup_{z \in X} \widetilde{\delta}(clos_\varepsilon(\delta(z, a)), w) \text{ für alle } X \subseteq Z \end{aligned}$$

Die von M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* \mid \widetilde{\delta}(clos_\varepsilon(S), w) \cap E \neq \emptyset\}$$

Dabei ist $clos_\varepsilon(X)$ die ε-Hülle einer Zustandsmenge $X \subseteq Z$:

$$clos_\varepsilon(X) := \begin{cases} X, & \text{wenn } \bigcup_{z \in X} \delta(z, \varepsilon) \subseteq X \\ clos_\varepsilon(X \cup \bigcup_{z \in X} \delta(z, \varepsilon)), & \text{sonst} \end{cases}$$

Eindeutige Start- und Endzustände

Satz 4.6.8

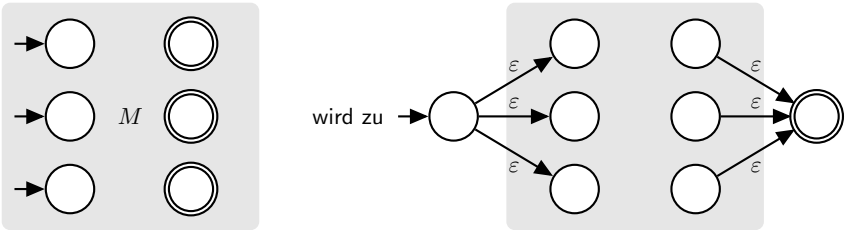
Für jeden NFA M mit ε-Übergängen gibt es einen NFA M' mit ε-Übergängen, sodass $L(M) = L(M')$ und M' genau einen Startzustand und genau einen Endzustand hat, wobei diese beiden Zustände verschieden sind.

Eindeutige Start- und Endzustände

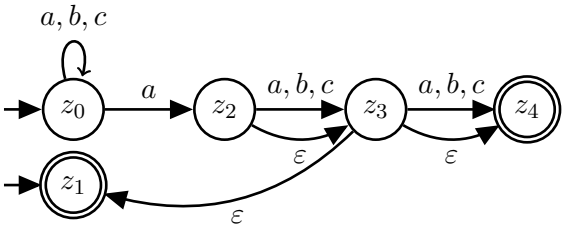
Satz 4.6.8

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.

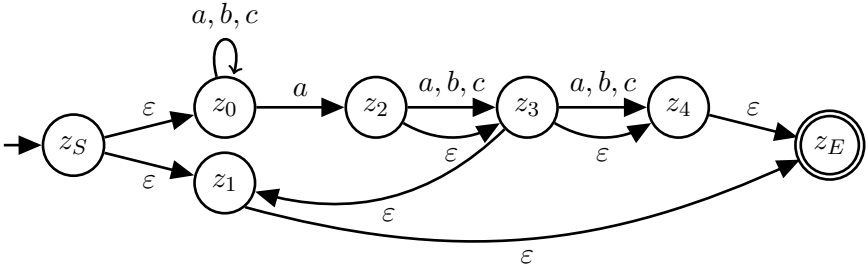
Beweis: Konstruiere M' aus M, durch Hinzufügen eines neuen Start- und eines neuen Endzustand mit ε-Übergängen:



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. Ein regulärer Ausdruck über Σ ist 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 regulärer Ausdruck ist, dann auch $(\alpha)^*$

Reguläre Ausdrücke (3)

Erzeugte Sprache

Die von einem regulären Ausdruck α erzeugte Sprache $L(\alpha)$ ist induktiv über dessen Struktur 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

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

  erzeugt alle Worte 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 Worte über $\{a, b, c\}$, die an viertletzter Stelle ein a haben und 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 Uhrzeiten im 24-Stunden-Format

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

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.