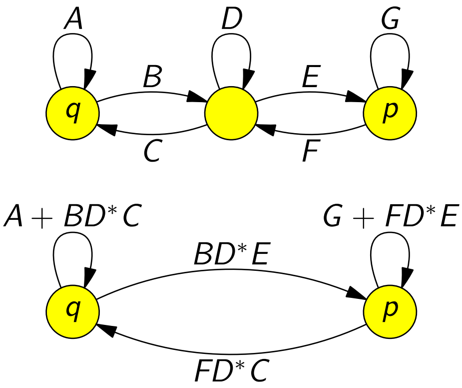
Eliminierung eines Zustandes

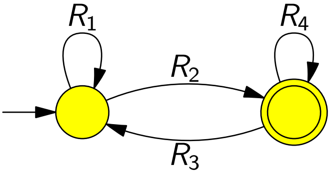


Für *alle* Paare (p, q)!

Eliminierung von Zuständen

Allgemeines Vorgehen:

1. Starte mit einem NFA mit nur einem Endzustand
2. Eliminiere Zustände bis auf den Start- und Endzustand
3. Lies den regulären Ausdruck ab



Gesuchter Ausdruck ist:

$$R_1^* R_2 (R_4 + R_3 R_1^* R_2)^*$$

Theorem (Pumping-Lemma)

Sei L eine reguläre Sprache. Dann gibt $n \in \mathbf{N}$, so daß jedes Wort $w \in L$ mit $|w| \geq n$ in Wörter $w = xyz$ zerlegt werden kann mit

1. $|xy| \leq n$
2. $|y| > 0$
3. $xy^i z \in L$ *für alle* $i \geq 0$

Beweis.

Es sei $L = L(M)$ für einen DFA M mit n Zuständen.
Ist $|w| \geq n$ dann durchläuft M einen Zustand doppelt.
p sei der erste solche Zustand.
Es gibt also $xyz = w$, mit

- $\hat{\delta}(q_0, x) = p$,
- $\hat{\delta}(p, y) = p$, $|xy| \leq n$, $|y| > 0$,
- $xy^i z \in L$

$\square$

Das Pumping-Lemma als Spiel

Gegeben sei eine Sprache $L \subseteq \Sigma^*$.

1. Alice wählt eine Zahl n.
2. Bob wählt ein Wort $w \in L$ mit $|w| \geq n$.
3. Alice wählt $x, y, z \in \Sigma^*$ mit $xyz = w$, $|xy| \leq n$, $|y| > 0$.
4. Bob wählt eine Zahl i.

Alice gewinnt, wenn $xy^iz \in L$.
Bob gewinnt, wenn $xy^iz \notin L$.

Falls L regulär ist, kann Alice immer gewinnen.

Gewinnstrategie für Bob $\Longrightarrow$ L nicht regulär!

Beispiel

Es sei $L = \{\, a^n b^n \mid n \geq 0 \,\}$.

1. Alice wählt eine Zahl n.
2. Bob wählt $w = a^n b^n$.
3. Alice wählt $x, y, z \in \{a, b\}^*$ mit $xyz = w$, $|xy| \leq n$, $|y| > 0$.
4. Bob wählt $i = 2$.

Bob hat gewonnen:

$xyyz$ enthält eine ungleiche Anzahl von a und b.

Also ist L (leider) nicht regulär.

Beispiel

Es sei $L = \{ a^{n^3} \mid n \geq 0 \}$.

1. Alice wählt eine Zahl n.
2. Bob wählt $w = a^{m^3}$ mit $m = n + 30$.
3. Alice wählt $x, y, z \in a^*$ mit $xyz = w$, $|xy| \leq n$, $|y| > 0$.
4. Bob denkt: $x = a^r$, $y = a^s$, $z = a^t$ mit $r + s + t = m^3$.
   Bob wählt $i = 0$ und denkt weiter:

$$(m-1)^3 = m^3 - 3m^2 + 3m - 1 < m^3 - m \leq |xy^0z| = r + t < m^3$$

Das Leerheitsproblem für reguläre Sprachen

Theorem

Das Leerheitsproblem für reguläre Sprachen ist in polynomieller Zeit lösbar:

Eingabe: Ein DFA, NFA oder ein regulärer Ausdruck für Sprache L

Frage: Ist $L = \emptyset$?

Beweis.

DFA, NFA: Ist ein Endzustand vom Anfangszustand erreichbar? Lineare Laufzeit mit Tiefensuche.

Regulärer Ausdruck: In NFA verwandeln. □

Das Wortproblem für reguläre Sprachen

Theorem

Das Wortproblem für reguläre Sprachen ist in quadratischer (DFA: linearer) Zeit lösbar:

Eingabe: Ein DFA, NFA oder ein regulärer Ausdruck für Sprache L

Frage: Ist $w \in L$?

Beweis.

DFA: Algorithmus aus Vorlesung
Laufzeit: $O(|M| + |w|)$
NFA: Algorithmus aus Vorlesung
Laufzeit: $O(|M| + |Q|^2 \cdot |\Sigma| + |Q|^2 \cdot |w|)$

Regulärer Ausdruck: In NFA verwandeln. $\square$

Das Universalitätsproblem für reguläre Sprachen

Theorem

Das Universalitätsproblem für reguläre Sprachen ist algorithmisch lösbar:

Eingabe: DFA, NFA oder regulärer Ausdruck für Sprache $L \subseteq \Sigma^*$

Frage: Ist $L = \Sigma^*$*?*

Beweis.

DFA: Sind alle erreichbaren Zustände Endzustände?
Lineare Laufzeit mit Tiefensuche.

NFA, regulärer Ausdruck: In DFA verwandeln.
Exponentielle Laufzeit. □

Das Endlichkeitsproblem für reguläre Sprachen

Theorem

Das Endlichkeitsproblem für reguläre Sprachen ist in linearer Zeit lösbar:

Eingabe: DFA, NFA oder regulärer Ausdruck für Sprache $L \subseteq \Sigma^*$

Frage: Ist L *endlich?*

Beweis.

NFA, DFA: Gibt es

1. eine starke Zusammenhangskomponente, die vom Startzustand erreichbar ist, und
2. von der aus ein Endzustand erreichbar ist?

Lineare Laufzeit (Konstruktion der starken Zusammenhangskomponenten: Datenstrukturen und Algorithmen) □

Das Äquivalenzproblem für reguläre Sprachen

Theorem

Das Äquivalenzproblem für reguläre Sprachen ist algorithmisch lösbar:

Eingabe: Zwei DFAs, NFAs oder reguläre Ausdrücke für Sprachen L_1 und L_2

Frage: Ist $L_1 = L_2$?

Beweis.

DFA: Minimiere beide und teste ob sie dann isomorph sind.

NFA, Regulärer Ausdruck: In DFA verwandeln. $\square$

Kontextfreie Grammatik

Definition

Eine *kontextfreie Grammatik* ist ein 4-Tupel (N, T, P, S)

- N ein Alphabet der *Nonterminale* oder *Variablen*
- T ein Alphabet der *Terminalsymbole*
- P Menge von *Produktionen* der Form $A \to \alpha$ mit $A \in N$ und $\alpha \in (N \cup T)^*$
- $S \in N$ das *Startsymbol*

Wir verlangen $N \cap T = \emptyset$.