

FORMALE SYSTEME

ÜBUNG 4

Eric Kunze

`eric.kunze@tu-dresden.de`

TU Dresden, 11. November 2021

Aufgabe 1 & 2:
Reguläre Ausdrücke

REGULÄRE AUSDRÜCKE

Die Menge der **regulärer Ausdrücke** über einem Alphabet Σ ist induktiv wie folgt definiert:

- ▶ \emptyset ist ein regulärer Ausdruck
- ▶ ε ist ein regulärer Ausdruck
- ▶ a ist ein regulärer Ausdruck für jedes $a \in \Sigma$
- ▶ Wenn α und β reguläre Ausdrücke sind,
dann sind auch $(\alpha\beta)$, $(\alpha \mid \beta)$ und $(\alpha)^*$ reguläre Ausdrücke

Die **Sprache eines regulären Ausdrucks** α wird mit $L(\alpha)$ bezeichnet und rekursiv definiert:

- | | |
|---|--|
| ▶ $L(\emptyset) = \emptyset$ | ▶ $L((\alpha\beta)) = L(\alpha) \circ L(\beta)$ |
| ▶ $L(\varepsilon) = \{\varepsilon\}$ | ▶ $L((\alpha \mid \beta)) = L(\alpha) \cup L(\beta)$ |
| ▶ $L(a) = \{a\} \quad \forall a \in \Sigma$ | ▶ $L((\alpha)^*) = L(\alpha)^*$ |

AUFGABE 1

Gegeben sind das Alphabet $\Sigma = \{a, b, c\}$ und die Sprache

$$L = \{w \in \Sigma^* \mid \begin{array}{l} \text{es gibt } u, v \in \Sigma^* \text{ mit } w = ubabcv \text{ und} \\ \text{es gibt } u, v \in \Sigma^* \text{ mit } w = ucccv \text{ und} \\ \text{es gibt kein } u \in \Sigma^* \text{ mit } w = au \end{array}\}.$$

Geben Sie für L einen regulären Ausdruck r mit $L = L(r)$ an.

AUFGABE 2

Beweisen Sie die folgenden Gleichungen für reguläre Ausdrücke r , s und t .

Erinnerung: $r \equiv s$ bedeutet $L(r) = L(s)$

a) $r \mid s \equiv s \mid r$

b) $(r \mid s) \mid t \equiv r \mid (s \mid t)$

c) $(rs)t \equiv r(st)$

d) $r(s \mid t) \equiv rs \mid rt$

e) $\emptyset^* \equiv \varepsilon$

f) $(r^*)^* \equiv r^*$

g) $r^* \equiv rr^* \mid \varepsilon$

h) $(\varepsilon \mid r)^* \equiv r^*$

Aufgabe 3:

RegExp \rightarrow *NFA*

REGEXP \rightarrow NFA: KOMPOSITIONELLE METHODE

Für einen Ausdruck α definieren wir rekursiv den ε -NFA $\mathcal{M}(\alpha)$:

Grundfälle:

- ▶ Wenn $\alpha = \emptyset$ dann $\mathcal{M}(\alpha) = \rightarrow \textcircled{A}$
- ▶ Wenn $\alpha = \varepsilon$ dann $\mathcal{M}(\alpha) = \rightarrow \textcircled{\textcircled{A}}$
- ▶ Wenn $\alpha = \textcolor{red}{a}$ dann $\mathcal{M}(\alpha) = \rightarrow \textcircled{A} \xrightarrow{\textcolor{red}{a}} \textcircled{\textcircled{B}}$

Rekursive Fälle: Wir bezeichnen mit $\text{elim}_\varepsilon(\mathcal{M})$ den NFA, der aus einem ε -NFA \mathcal{M} durch Eliminierung der ε -Übergänge entsteht.

- ▶ Wenn $\alpha = (\beta\gamma)$ dann $\mathcal{M}(\alpha) = \text{elim}_\varepsilon(\mathcal{M}(\beta) \odot \mathcal{M}(\gamma))$
- ▶ Wenn $\alpha = (\beta \mid \gamma)$ dann $\mathcal{M}(\alpha) = \mathcal{M}(\beta) \oplus \mathcal{M}(\gamma)$
- ▶ Wenn $\alpha = (\beta)^*$ dann $\mathcal{M}(\alpha) = \text{elim}_\varepsilon(\mathcal{M}(\beta)^*)$

REGEXP \rightarrow NFA: EXPLIZITE KONSTRUKTION

Gegeben: regulärer Ausdruck α ohne \emptyset

Initialisierung: $\mathcal{M}_\alpha = \rightarrow (A) \xrightarrow{\alpha} (B)$

Solange es in \mathcal{M}_α Übergänge $q \xrightarrow{\beta} p$ gibt, die mit einem Ausdruck $\beta \notin \{\varepsilon\} \cup \Sigma$ beschriftet sind, wende eine der folgenden Regeln an:

► Ersetze $q \xrightarrow{(\gamma_1 \gamma_2)} p$ durch $q \xrightarrow{\gamma_1} r \xrightarrow{\gamma_2} p$

► Ersetze $q \xrightarrow{(\gamma_1 \mid \gamma_2)} p$ durch $q \xrightarrow{\gamma_1} p$ und $q \xrightarrow{\gamma_2} p$

► Ersetze $q \xrightarrow{(\gamma)^*} p$ durch $q \xrightarrow{\varepsilon} r \xrightarrow{\varepsilon} p$ mit einem Selbstübergang $r \xrightarrow{\gamma} r$

AUFGABE 3

Geben Sie zu jedem der regulären Ausdrücke r_i einen NFA \mathcal{M}_i mit $L(\mathcal{M}_i) = L(r_i)$ an.

a) $r_1 = (ab)^*$

b) $r_2 = a(b \mid c)a^* \mid a^*$

Wenden Sie dabei jeweils den *kompositionellen Ansatz* sowie den *expliziten Ansatz* zur Konstruktion von NFAs aus der Vorlesung an.

Aufgabe 4:

NFA* \rightarrow *RegExp

NFA \rightarrow REGEXP: ERSETZUNGSMETHODE

Gegeben: NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$

Gesucht: regulärer Ausdruck α mit $\mathbf{L}(\alpha) = \mathbf{L}(\mathcal{M})$

Idee:

Für jeden Zustand $q \in Q$, berechne einen regulären Ausdruck α_q für die Sprache $\mathbf{L}(\alpha_q) = \mathbf{L}(\mathcal{M}_q)$ mit $\mathcal{M}_q = \langle Q, \Sigma, \delta, \{q\}, F \rangle$

Für $Q_0 = \{q_1, q_2, \dots, q_n\}$ gilt dann

$$\mathbf{L}(\mathcal{M}) = \bigcup_{q \in Q_0} \mathbf{L}(\alpha_q) = \mathbf{L}(\alpha_{q_1} \mid \alpha_{q_2} \mid \dots \mid \alpha_{q_n})$$

(1) **Vereinfache den Automaten** (entferne offensichtlich unnötige Zustände)

(2) **Bestimme das Gleichungssystem**

Intuition: Beschreibe α_q in Abhängigkeit von Folgezuständen

▷ Für jeden Zustand $q \in Q \setminus F$: $\alpha_q \equiv \sum_{a \in \Sigma} \sum_{p \in \delta(q,a)} a \alpha_p$

▷ Für jeden Zustand $q \in F$: $\alpha_q \equiv \varepsilon \mid \sum_{a \in \Sigma} \sum_{p \in \delta(q,a)} a \alpha_p$

(3) **Löse das Gleichungssystem** durch Einsetzen und

Regel von Arden: Aus $\alpha \equiv \beta \alpha \mid \gamma$ mit $\varepsilon \notin L(\beta)$ folgt $\alpha \equiv \beta^* \gamma$.

(4) **Gib den Ausdruck für die Sprache des NFA an**

Für $Q_0 = \{q_1, q_2, \dots, q_n\}$ gilt dann

$$L(\mathcal{M}) = \bigcup_{q \in Q_0} L(\alpha_q) = L(\alpha_{q_1} \mid \alpha_{q_2} \mid \dots \mid \alpha_{q_n})$$

NFA \rightarrow REGEXP: DYNAMISCHE ERMITTLUNG

Gegeben: NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$

Gesucht: regulärer Ausdruck α mit $\mathbf{L}(\alpha) = \mathbf{L}(\mathcal{M})$

Ansatz:

Für jedes Paar von Zuständen $q, p \in Q$, berechne einen regulären Ausdruck $\alpha_{q,p}$ für die Sprache $\mathbf{L}(\alpha_{q,p}) = \mathbf{L}(\mathcal{M}_{q,p})$ mit $\mathcal{M}_{q,p} = \langle Q, \Sigma, \delta, \{q\}, \{p\} \rangle$

Dann gilt:

$$\mathbf{L}(\mathcal{M}) = \bigcup_{q \in Q_0} \bigcup_{p \in F} \mathbf{L}(\alpha_{q,p}) = \mathbf{L} \left(\sum_{q \in Q_0} \sum_{p \in F} \alpha_{q,p} \right)$$

- ▶ $\mathbf{L}^k[i, j] \dots$ Sprache mit Start in q_i , Ende in q_j und nutzt nur Zwischenzustände q_1, \dots, q_k
- ▶ $\alpha^k[i, j]$ zugehöriger regulärer Ausdruck

Idee: lasse immer mehr Zwischenzustände von q nach p zu

- ▶ $k = n$: nutze alle Zustände — Ergebnis ablesbar
- ▶ $k = 0$: nutze keine Zwischenzustände — $\alpha^0[i, j]$ direkt ablesbar:

Sei $\{a_1, \dots, a_m\} = \{a \in \Sigma \mid q_i \xrightarrow{a} q_j\}$ die Menge der Beschriftungen von direkten Übergängen von q_i zu q_j .

- ▷ Falls $i \neq j$, dann ist $\alpha^0[i, j] = a_1 \mid \dots \mid a_m$
- ▷ Falls $i = j$, dann ist $\alpha^0[i, j] = a_1 \mid \dots \mid a_m \mid \varepsilon$

Update-Formel:

$$\alpha^{k+1}[i, j] = \alpha^k[i, j] \mid (\alpha^k[i, k+1](\alpha^k[k+1, k+1])^* \alpha^k[k+1, j])$$

vgl. VL „Algorithmen & Datenstrukturen“, Prozess-Problem im
Aho-Hopcroft-Ullman-Algorithmus

AUFGABE 4

Entwickeln Sie für die Sprache L über dem Alphabet $\Sigma = \{a, b, c\}$ einen regulären Ausdruck r mit $L = L(r)$. Für alle Wörter $w \in L$ gilt:

- ▶ w enthält aaa .
- ▶ w endet mit c .
- ▶ Die Anzahl der b in w ist gerade.

Aufgabe 5:

Minimierung von Automaten

ÄQUIVALENZ VON ZUSTÄNDEN & QUOTIENTENAUTOMAT

DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle \rightsquigarrow$ DFA $\mathcal{M}_q = \langle Q, \Sigma, \delta, q, F \rangle$

Äquivalenz von Zuständen:	$p \sim_{\mathcal{M}} q \Leftrightarrow \mathbf{L}(\mathcal{M}_p) = \mathbf{L}(\mathcal{M}_q)$
Äquivalenzklasse:	$[q]_{\sim} = \{p \in Q \mid q \sim p\}$
Quotient von $P \subseteq Q$:	$P/_{\sim} = \{[p]_{\sim} \mid p \in P\}$

Quotientenautomat: Verschmelzen von äquivalenten Zuständen

Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle$ mit totaler Übergangsfunktion ist der Quotientenautomat $\mathcal{M}/_{\sim} = \langle Q/_{\sim}, \Sigma, \delta_{\sim}, [q_0]_{\sim}, F/_{\sim} \rangle$ gegeben durch

- ▶ $Q/_{\sim} = \{[q]_{\sim} \mid q \in Q\}$
- ▶ $\delta_{\sim}([q]_{\sim}, a) = [\delta(q, a)]_{\sim}$
- ▶ $F/_{\sim} = \{[q]_{\sim} \mid q \in F\}$

Bestimmung von \sim :

- Initialisiere $\sim := \emptyset$
- **Regel 1:** Für jedes Paar von Zuständen $\langle q, p \rangle \in Q \times Q$:
falls $q \in F$ und $p \notin F$, dann „speichere $q \not\sim p$ “
- **Regel 2:** Für jedes Paar $\langle q, p \rangle \in Q \times Q \setminus \sim$ und jedes $a \in \Sigma$:
falls $\delta(q, a) \not\sim \delta(p, a)$ dann „speichere $q \not\sim p$ “
- Wiederhole Regel 2 bis keine Änderungen mehr auftreten
- Das Ergebnis ist $(Q \times Q) \setminus \sim$

Beispiel: Für einen DFA mit Zuständen $Q = \{A, B, C, D, E\}$ genügt eine Tabelle mit zehn Feldern (statt $5^2 = 25$).

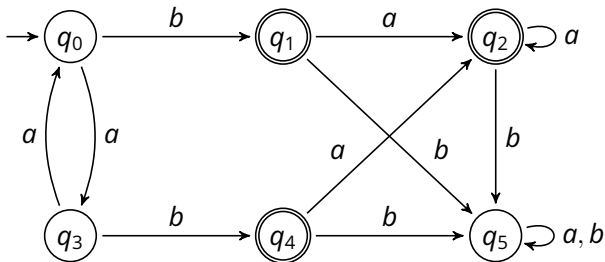
(dazu reihen wir Zustände vertikal in umgekehrter Reihenfolge)

	A	B	C	D
E				
D				
C				
B				

AUFGABE 5

Berechnen Sie für folgenden DFA

$\mathcal{M} = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{a, b\}, \delta, q_0, \{q_1, q_2, q_4\})$ mit δ :



die Äquivalenzrelation $\sim_{\mathcal{M}}$, und geben Sie den Quotientenautomaten \mathcal{M}/\sim an.