Prof. Javier Esparza
Kevin Kappelmann, Simon Roßkopf, Lukas Stevens

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2023 – Übungsblatt 3

- **Tipp:** Auf dieser (und dort verlinkten) Website(n) können Sie interaktiv DFAs, NFAs, reguläre Ausdrücke,... erzeugen, simulieren, umwandeln,...
- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.
- Für den Rest des Semesters gilt: $\mathbb{N} := \mathbb{N}_0 := \{0, 1, 2, ...\}$ und $\mathbb{N}_+ := \mathbb{N}_0 \setminus \{0\}$.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Individualaufgabe Ü3.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- Ardens Lemma
- Produktkonstruktion

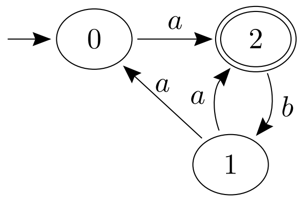
Übung und Nachbereitung

Übungsaufgabe Ü3.2. (*RE -> ϵ-NFA*)

Wandeln sie folgenden regulären Ausdruck $a \mid ((b \mid \emptyset)(a(b\emptyset)^*)^*)$ in einen ϵ-NFA um. Folgen Sie hierfür strikt dem Vorgehen aus der Vorlesung.

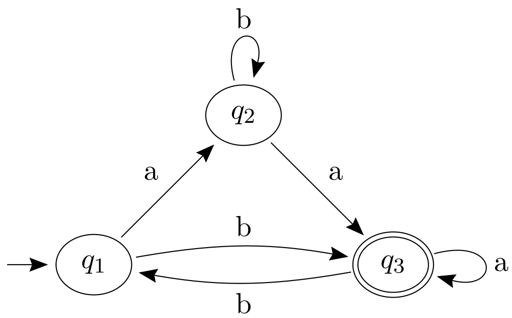
Übungsaufgabe Ü3.3. (*ϵ-NFA -> RE*)

Wandeln Sie folgenden ϵ-NFA in einen regulären Ausdruck um. Folgen Sie dem Vorgehen in der Vorlesung und eliminieren Sie die Zustände in aufsteigender Reihenfolge.



Übungsaufgabe Ü3.4. (*Ardens Lemma*)

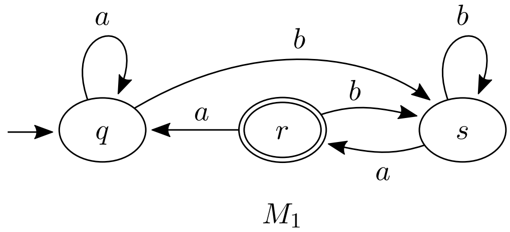
Gegeben sei folgender Automat $M = (\{q_1, q_2, q_3\}, \{a, b\}, \delta, q_1, \{q_3\})$:

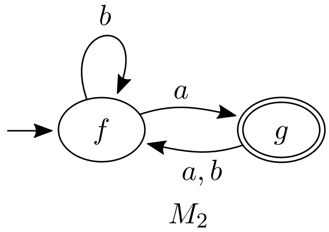


Berechnen Sie mit dem Gauß-Verfahren und Ardens Lemma einen regulären Ausdruck α mit $L(\alpha) = L(M)$.

Übungsaufgabe Ü3.5. (*Produktkonstruktion*)

(a) Konstruieren Sie einen DFA M mit $L(M) = L(M_1) \cap L(M_2)$, indem Sie die Produktkonstruktion verwenden. Ist M minimal?





(b) Konstruieren Sie nun einen Automaten für $L(M_1) \cup L(M_2)$.

Übungsaufgabe Ü3.6. (*Noch mehr Abschlussspaß*)

In der Vorlesung haben Sie verschiedenste Abschlusseigenschaften von regulären Sprachen gesehen. In dieser Aufgabe beweisen wir eine weitere solche Abschlusseigenschaft.

Betrachte zwei Alphabete Σ, Δ und eine Funktion $h : \Sigma \to \Delta^*$, die Buchstaben aus Σ zu Wörter in Δ^* umwandelt.

Jede solche Funktion kann rekursiv auf eine Funktion $h^* : \Sigma^* \to \Delta^*$ erweitert werden, die Wörter aus Σ^* zu Wörter in Δ^* umwandelt, indem sie sukzessiv die Buchstaben des Eingabewortes mit Hilfe von h transformiert. Formell:

$$h^*(\epsilon) \coloneqq \epsilon \qquad \text{und} \qquad h^*(aw) \coloneqq h(a)h^*(w) \quad \text{für } a \in \Sigma, w \in \Sigma^*$$

Beispiel: Sei $\Sigma = \{a, b, c\}$, $\Delta = \{T, H, \heartsuit, E, O\}$ und $h : \Sigma \to \Delta^*$ definiert durch

$$h(a) \coloneqq EO \qquad h(b) \coloneqq \heartsuit \qquad h(c) \coloneqq TH$$

Dann gilt

$$\begin{aligned} h^*(cab) &= h(c)h^*(ab) = h(c)h(a)h^*(b) = h(c)h(a)h(b)h^*(\epsilon) = h(c)h(a)h(b) \\ &= THh(a)h(b) = THEOh(b) = THEO\heartsuit \end{aligned}$$

Sei L nun eine Sprache über dem Alphabet Σ und $h : \Sigma \to \Delta^*$ ein Funktion. Wir definieren $h(L) \coloneqq \{h^*(w) \mid w \in L\}$. Beachte, dass $h(L) \subseteq \Delta^*$.

Zeigen Sie: wenn L regulär ist, dann ist auch $h(L)$ regulär. Für Ihren Beweis dürfen Sie folgende Gleichungen für alle Sprachen $L_1, L_2 \subseteq \Sigma^*$ ohne Beweis verwenden[1]:

[1] die Beweise überlassen wir Ihnen als Übung.

(i) $h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$

(ii) $h(L_1 L_2) = h(L_1)h(L_2)$

(iii) $h(L_1^*) = h(L_1)^*$