Sebastian Muskalla Prof. Dr. Roland Meyer	**Theoretische Informatik 1** **Übungsblatt 4**	TU Braunschweig Wintersemester 2018/19

Ausgabe: 28.11.2018 Abgabe: 6.12.2018, 14:00

Geben Sie Ihre Lösungen bis Donnerstag, 6.12.2018, 14:00 Uhr, durch Einwerfen in die Übungskästen neben Büro IZ 343 ab. Geben Sie in Gruppen von 4 Personen ab.

Aufgabe 1: NFAs mit internen Transitionen

a) Es sei $A = (\tau, A_\tau)$ ein NFA mit internen Transitionen über Σ. Hierbei ist $A_\tau = (Q, q_0, \rightarrow, Q_F)$ ein NFA über Σ.

Wir definieren den NFA $B = (Q, q_0, \rightarrow', Q'_F)$ über Σ, mit

$$Q'_F = \{q \in Q \mid \exists q_f \in Q_F \colon q_f \in \tau\text{-closure}(q)\} ,$$

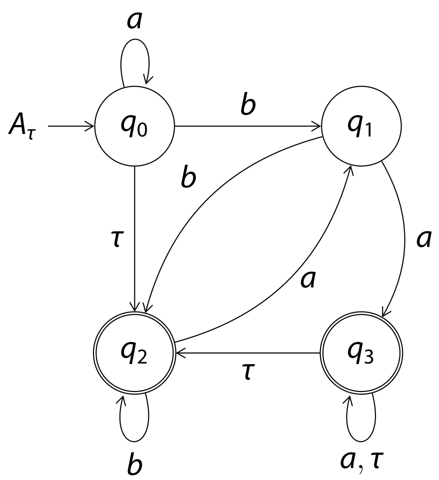
$$q \xrightarrow{a}{}' q' \quad \text{gdw.} \quad \exists q_1, q_2 \in Q \colon q_1 \in \tau\text{-closure}(q), q_1 \xrightarrow{a} q_2 \text{ in } A_\tau, q' \in \tau\text{-closure}(q_2) .$$

Beweisen Sie formal, dass $\mathcal{L}(A) = \mathcal{L}(B)$ gilt.

Hinweis: Es handelt sich hierbei um eine modifizierte Version der Konstruktion aus der Vorlesung. Wir verwenden hier bei Transitionen auf beiden Seiten den τ-Abschluss.

b) Gilt $\mathcal{L}(A) = \mathcal{L}(B)$ in Aufgabenteil a) auch dann noch, wenn wir $Q'_F = Q_F$ definieren? Begründen Sie Ihre Antwort!

c) Betrachten Sie den NFA mit internen Transitionen $A = (\tau, A_\tau)$ über $\Sigma = \{a, b\}$.



Geben Sie zu jedem Zustand q_i seinen τ-Abschluss τ-closure(q_i) an.

Benutzen Sie wahlweise das Verfahren aus der Vorlesung oder das Verfahren aus Aufgabenteil a), um einen zu A äquivalenten NFA (ohne interne Transitionen) zu bestimmen.

Aufgabe 2: Das Inklusionsproblem

Betrachten Sie das Inklusionsproblem für reguläre Sprachen.

Reguläre Inklusion (INCLUSIONREG)	
Gegeben:	NFAs A, B über Σ.
Entscheide:	Gilt $\mathcal{L}(A) \subseteq \mathcal{L}(B)$?

a) Geben Sie einen Algorithmus als Pseudocode an, der das Inklusionsproblem löst. Begründen Sie kurz, warum Ihr Algorithmus korrekt ist.

Sie dürfen hierbei die aus der Vorlesung bekannten Probleme wie z.B. EMPTYREG als Subroutinen aufrufen.

b) Analysieren Sie den Zeitverbrauch Ihres Algorithmus. Berechnen Sie hierzu insbesondere die Anzahl Zustände, die die auftretenden Automaten haben.

c) Lässt sich ein besseres Algorithmus für INCLUSIONREG finden, wenn wir garantieren, dass der eingegebene Automat B ein DFA ist? Begründen Sie Ihre Antwort.

d) Lässt sich ein besseres Algorithmus für INCLUSIONREG finden, wenn wir garantieren, dass der eingegebene Automat A ein DFA ist? Begründen Sie Ihre Antwort.

Aufgabe 3: Determinisierung ist teuer

In dieser Aufgabe wollen wir zeigen, dass sich manche Sprachen mit einem kleinen NFA beschreiben lassen, jeder DFA dafür jedoch zwangsweise riesig ist.

Für eine Zahl $k \in \mathbb{N}, k > 0$ sei

$$\mathcal{L}_{a@k} = \left\{ w \in \Sigma^* \;\middle|\; \text{Der } k\text{. letzte Buchstabe in } w \text{ ist } a \right\}$$

die Sprache der Wörter über $\Sigma = \{a, b\}$, deren k-ter Buchstabe von rechts ein a ist.

Beispielsweise ist $\mathcal{L}_{a@3} = \Sigma^*.a.(a \cup b).(a \cup b)$ die Sprache der Wörter, deren drittletzter Buchstabe a ist.

a) Zeigen Sie, wie man zu jedem $k \in \mathbb{N}, k > 0$ einen NFA $A_k = (Q, q_0, \rightarrow, Q_F)$ mit $\mathcal{L}(A_k) = \mathcal{L}_{a@k}$ konstruiert. Geben Sie diesen Automaten formal als Tupel an.

Sie müssen die Sprachgleichheit nicht beweisen.

Wie viele Zustände hat A_k?

b) Zeichnen Sie A_3 und bestimmen Sie seine Determinisierung A_3^{det} mittels der Rabin-Scott-Potenzmengenkonstruktion.

Vergleichen Sie die Zustandsanzahl von A_3 und A_3^{det}.

c) Sei $k \in \mathbb{N}, k > 0$ beliebig. Beweisen Sie, dass es für $\mathcal{L}_{a@k}$ keinen DFA B mit weniger als 2^k Zuständen gibt, so dass $\mathcal{L}(B) = \mathcal{L}_{a@k}$ gilt.

Hinweis: Gehen Sie wie folgt vor:

1. Angenommen es gäbe $B = (Q', q_0', \to', Q_F')$ mit $\mathcal{L}(B) = \mathcal{L}_{a@k}$ und $|Q'| < 2^k$.
2. Betrachten Sie die Menge Σ^k der Wörter der Länge k. Wie viele solcher Wörter gibt es?
3. Betrachten Sie zu jedem Wort $w \in \Sigma^k$ den (eindeutigen) Zustand q_w, in dem DFA B ist, nachdem er w gelesen hat.
4. Leiten Sie nun einen Widerspruch her.

Aufgabe 4: Äquivalenzrelationen

Es sei $\equiv \subseteq \Sigma^* \times \Sigma^*$ eine Äquivalenzrelation auf Wörtern. Wie üblich schreiben wir $u \equiv v$ (statt $(u, v) \in \equiv$), um auszudrücken, dass u und v gemäß $\equiv$ äquivalent sind.

a) Beweisen Sie formal die folgenden grundlegenden Eigenschaften von Äquivalenzrelationen:

- Jedes Wort ist in seiner Äquivalenzklasse enthalten: $u \in [u]_\equiv$.
- Die Äquivalenzklassen von äquivalenten Wörtern sind gleich: $u \equiv v \implies [u]_\equiv = [v]_\equiv$.
- Die Äquivalenzklassen von nicht-äquivalenten Wörtern sind disjunkt: $u \not\equiv v \implies [u]_\equiv \cap [v]_\equiv = \emptyset$.

b) Es sei $\mathcal{L} \subseteq \Sigma^*$ und $\equiv_\mathcal{L}$ die aus der Vorlesung bekannte Nerode-Rechtskongruenz mit

$$u \equiv_\mathcal{L} v \quad \text{gdw.} \quad \forall w \in \Sigma^*\colon u.w \in \mathcal{L} \iff v.w \in \mathcal{L}\,.$$

Beweisen Sie, dass $\equiv_\mathcal{L}$ tatsächlich eine Äquivalenzrelation und Rechtskongruenz ist. Letzteres bedeutet, dass für alle u, v mit $u \equiv_\mathcal{L} v$ und alle $x \in \Sigma^*$ gilt: $u.x \equiv_\mathcal{L} v.x$.

c) Es sei $A = (Q, q_0, \to, Q_F)$ ein DFA. Die Relation $\equiv_A \subseteq \Sigma^* \times \Sigma^*$ ist definiert durch:

$$u \equiv_A v \quad \text{gdw.} \quad \exists q \in Q\colon q_0 \xrightarrow{u} q \text{ und } q_0 \xrightarrow{v} q.$$

Zeigen Sie, dass $\equiv_A$ eine Äquivalenzrelation ist.

d) Ist $\equiv_A$ aus Aufgabenteil c) auch eine Äquivalenzrelation, wenn A ein NFA ist? Begründen Sie Ihre Antwort!