Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 2

Abgabe: 06.05.2024, 12:00 CEST

- Die Hausaufgaben werden in folgender Reihenfolge korrigiert: H2.4, H2.5, H2.6, H2.3

AT-Aufgabe H2.1. (*Dem Tutor, dem die Studis vertrauen*) keine Korrektur

Bearbeiten Sie folgende Aufgaben in Automata Tutor (Aufgaben H2.1 a–f). Konstruieren Sie reguläre Ausdrücke für die folgenden Sprachen über dem Alphabet $\Sigma := \{a, b\}$. Beachten Sie dabei, dass die Bedingungen ab Aufgabenteil (b) sukzessiv kombiniert werden, die Sprache für (d) soll also beispielsweise die Bedingungen von (b) und (c) erfüllen.

(a) $\{(ab)^n a(ba)^n : n \in \mathbb{N}\}$.

(b) Die Sprache aller Wörter, die nicht mit einem a beginnen.

(c) Zusätzlich müssen die Wörter mit einem b enden.

(d) Zusätzlich müssen die Wörter eine ungerade Anzahl von as haben.

(e) Zusätzlich dürfen die Wörter bb nicht enthalten.

(f) Sei $L := L(ab^* \mid a^*b(a \mid b)^*)$. Geben Sie drei Wörter in L und drei Wörter nicht in L an.

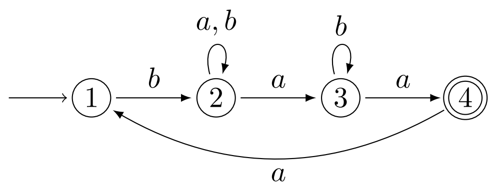
AT-Aufgabe H2.2. (*DFA-Konstruktion*) keine Korrektur

Bearbeiten Sie die Aufgaben 2(a)–(n) (außer (e)) auf Seite 88 des Buches Automata, Computability and Complexity: Theory and Applications in Automata Tutor. Die Beschreibungen der Sprachen auf Deutsch können Sie in Automata Tutor finden.

Aufgabe H2.3. (*Potenzmengenkonstruktion*)

Konvertieren Sie den folgenden NFA über dem Alphabet $\Sigma = \{a, b\}$ zu einem DFA, indem Sie die Potenzmengenkonstruktion aus der Vorlesung verwenden.

Hinweis: Es genügt, nur die vom Startzustand erreichbaren Zustände zu konstruieren.



Aufgabe H2.4. (*Sag a!*)

Sei $\Sigma = \{a, b\}$ und sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA über Σ. Geben Sie einen NFA N' für die Sprache $\{w \in L(N) \mid |w|_a \geq 1\}$ aller Wörter aus $L(N)$, die mindestens ein a enthalten, formal (als Tupel) an.

Hinweis: Sie dürfen $(q, a, r) \in \delta$ statt $r \in \delta(q, a)$ schreiben.

Aufgabe H2.5. (*Union of the States*)

Sei $\Sigma := \{\mathtt{a}, ..., \mathtt{z}\}$ und $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA über Σ. Wir nennen M *synchronisierend*, wenn es einen Zustand $q \in Q$ gibt, sodass jeder Zustand nach Lesen des Wortes `theo` in q landet. Formal gilt also $\hat{\delta}(r, \mathtt{theo}) = q$ für alle $r \in Q$.

(a) Sei $L := \Sigma^*\{\mathtt{theo}\}$, also die Sprache aller Wörter, die auf `theo` enden. Beweisen Sie, dass jeder synchronisierende DFA M alle oder keine Wörter aus L akzeptiert, d.h. $L(M) \cap L \in \{\emptyset, L\}$. Gehen Sie dabei kleinschrittig vor und beziehen sich, wenn möglich, auf die entsprechenden Definitionen.

Wir betrachtet nun auch NFAs. Einen NFA $M = (Q, \Sigma, \delta, q_0, F)$ bezeichnen wir demnach als *synchronisierend*, wenn es ein $q \in Q$ gibt, sodass $\hat{\delta}(\{r\}, \mathtt{theo}) = \{q\}$ für jedes $r \in Q$ gilt.

(b) Konstruieren Sie einen synchronisierenden NFA, der das Wort `tumtheo` akzeptiert, aber `lmutheo` nicht. Begründen Sie, wieso Ihr NFA synchronisierend ist.

Update: Sie können diese Aufgabe nun auch in apotheosis lösen.

Aufgabe H2.6. (*Regex-Rekursion*)

In den Übungsaufgaben haben wir eine Funktion definiert, die für einen regulären Ausdruck r bestimmt, ob die Sprache $L(r)$ unendlich viele Wörter enthält. Nun wollen wir eine Funktion $\mathsf{contains}(\mathsf{a}, r)$ definieren, die für einen Buchstaben $\mathtt{a} \in \Sigma$ und regulären Ausdruck r berechnet, ob $\mathtt{a}$ in jedem Wort aus $L(r)$ vorkommt. Für alle $w \in L(r)$ soll also gelten, dass $(\exists u, v \in \Sigma^*.\ w = u\mathsf{a}v) =: P(w)$.

(a) Geben Sie eine rekursive Funktion an, indem Sie das folgende Gerüst vervollständigen.

- $\mathsf{contains}(\mathsf{a}, \boldsymbol{\emptyset}) = \ldots$
- $\mathsf{contains}(\mathsf{a}, \boldsymbol{\epsilon}) = \ldots$
- $\mathsf{contains}(\mathsf{a}, \mathsf{x}) = \ldots$
- $\mathsf{contains}(\mathsf{a}, r^*) = \ldots$
- $\mathsf{contains}(\mathsf{a}, r_1|r_2) = \ldots$
- $\mathsf{contains}(\mathsf{a}, r_1 r_2) = \ldots$

(b) Beweisen Sie die Korrektheit ihrer Funktion mit struktureller Induktion. Kennzeichnen Sie dabei die Induktionshypothesen und deren Anwendungen deutlich.

Quizaufgabe H2.7. (*Sparmaßnahmen*) keine Korrektur

Die Universitätsleitung zeigt sich entsetzt über die große Menge an Tinte, die in die fettgedruckten Symbole $\boldsymbol{\epsilon}$ und $\boldsymbol{\emptyset}$ in regulären Ausdrücken fließt. Deshalb wird angeordnet, dass diese Symbole künftig – wenn möglich – vermieden werden. Jeder reguläre Ausdruck soll in eine der folgenden Formen gebracht werden:

(F1) r (F2) $r \mid \boldsymbol{\epsilon}$ (F3) $\boldsymbol{\epsilon}$ (F4) $\boldsymbol{\emptyset}$

Hierfür ist r ein beliebiger regulärer Ausdruck, der weder $\boldsymbol{\epsilon}$ noch $\boldsymbol{\emptyset}$ enthält.

Beruhigen Sie ihre verzweifelten Kollegen, indem Sie beweisen, dass jeder Ausdruck in diese Form gebracht werden kann. Zeigen Sie, dass zu jedem regulären Ausdruck r' ein äquivalenter Ausdruck r existiert, der in einer der obigen Formen ist.

Hinweise: Sie können strukturelle Induktion verwenden. Sie dürfen ohne Beweis verwenden, dass $(r \mid \boldsymbol{\epsilon})^* \equiv r^*$ für jeden regulären Ausdruck r gilt.

Knobelaufgabe H2.8. (*Auf den Kopf gefallen*)

Sei $\Sigma \neq \emptyset$ und $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA, sodass der Automat $M' = (Q, \Sigma, \delta', q_0, F)$, den man erhält, wenn man alle Transitionen in M umdreht, auch ein DFA ist. Formal gilt also $\delta'(r, c) = q$ für alle $\delta(q, c) = r$. Zeigen Sie, dass $L(M)$ unendlich oder leer ist.