Prof. T. Nipkow
K. Kappelmann, J. Rädle, L. Stevens

KEINE ABGABE

Technische Universität München
Lehrstuhl für Logik und Verifikation

Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt 2

AUFGABE 2.1. (*Wichtige Begriffe*) Stufe A

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

- ε-NFA
- Potenzmengenkonstruktion
- rechtslineare Grammatik
- Komplementierung
- Produktkonstruktion
- regulärer Ausdruck
- rekursive Prozedur
- Strukturelle Induktion

AUFGABE 2.2. (*Automata Tutor*) Stufe B

Lösen Sie die Aufgaben unter "English to Regular Expression" und "Regular Expression to NFA". Beachten Sie, dass der AutomataTutor automatisch $c \,|\, \varepsilon$ als $c?$ abkürzt.

AUFGABE 2.3. (*Tick Tock Boom, Blow Up*) Stufe C

Mit $|w|_x$ bezeichnen wir die Anzahl der Vorkommen des Buchstabens $x \in \Sigma$ in $w \in \Sigma^*$. Sei $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ und $\mathsf{L} = \left\{ w \in \Sigma^* \mid \exists x \in \Sigma.\ |w|_x = 0 \right\}$.

(a) Konstruieren Sie einen NFA N mit genau 4 Zuständen und $\mathsf{L}(\mathsf{N}) = \mathsf{L}$.
(b) Determinisieren Sie den NFA N aus (a) mittels der Potenzmengenkonstruktion um einen DFA D mit $\mathsf{L}(\mathsf{D}) = \mathsf{L}(\mathsf{N})$ zu erhalten.
(c) Geben Sie eine rechtslineare Grammatik G (gemäß Satz 3.13) an, sodass $\mathsf{L}(\mathsf{G}) = \mathsf{L}(\mathsf{D})$.
(d) Übersetzen Sie die Grammatik G (gemäß Satz 3.14) in einen ε-NFA N', sodass $\mathsf{L}(\mathsf{N}') = \mathsf{L}(\mathsf{G})$.
(e) Vergleichen Sie die NFAs N' und N bezüglich Zustands- und Transitionszahl. Diskutieren Sie dann, inwiefern Sie Ihre Beobachtung verallgemeinern können.

AUFGABE 2.4. (*I LoVe[1] Induction*) Stufe D

Vervollständigen Sie den Beweis von Satz 3.14 aus der Vorlesung:

Für jede rechtslineare Grammatik G gibt es einen NFA M mit $\mathsf{L}(\mathsf{G}) = \mathsf{L}(\mathsf{M})$.

AUFGABE 2.5. Stufe D

Entscheiden Sie, ob die folgenden Aussagen korrekt sind oder nicht und begründen Sie Ihre Behauptung, indem Sie entweder ein Gegenbeispiel oder eine passende Konstruktion angeben.

Für jeden NFA $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ gibt es einen NFA $\mathsf{N}' = (\mathsf{Q}', \Sigma', \delta', \mathsf{q}_0', \mathsf{F}')$ mit $\mathsf{L}(\mathsf{N}) = \mathsf{L}(\mathsf{N}')$ und ...

(a) der Startzustand hat keine eingehenden Kanten.
(b) kein Endzustand hat eine ausgehende Kante.
(c) für jeden Zustand gilt: alle eingehenden Kanten sind mit demselben Zeichen beschriftet.
(d) für jeden Zustand gilt: alle ausgehenden Kanten sind mit demselben Zeichen beschriftet.

AUFGABE 2.6. (*Automata Tutor*) Stufe B

Hinweis: Die folgenden Aufgaben stehen auch im AutomataTutor zur Verfügung. Zudem finden Sie dort weitere neue Aufgaben. Die Aufgaben finden Sie unter den Kategorien "English to Regular Expression" und "Regular Expression to NFA". Einige der Aufgaben sind auch bepunktete Hausaufgaben. Beachten Sie, dass der AutomataTutor automatisch $c \,|\, \varepsilon$ als $c?$ abkürzt.

Geben Sie für jede der folgenden Sprachen einen regulären Ausdruck an, der genau die Sprache beschreibt. Verwenden Sie für die ersten drei Aufgaben das Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ und für die nächsten beiden $\Sigma = \{0, 1\}$.

(a) Wörter gerader Länge.
(b) Wörter, die mit einem a beginnen und enden, sowie Wörter, die mit einem b beginnen und enden.
(c) Wörter, in denen kein a neben einem b steht.
(d) Zahlen in Binärdarstellung (most-significant-bit-first), die durch 2 teilbar sind.
(e) Zahlen in Binärdarstellung (most-significant-bit-first), die nicht durch 4 teilbar sind.
(f) **Knobelaufgabe:** Sei $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Wörter, die gleich oft die Zeichenketten ab und ba enthalten.
Beispiel: Das Wort abab enthält zweimal ab, aber nur einmal ba und soll somit *kein* Element der Sprache sein.

Stufe C

AUFGABE 2.7. (*Draw Me Like One of Your Deterministic Automata*)

(a) Geben Sie für jeden Teilausdruck des regulären Ausdrucks $((\mathsf{a}\emptyset)^*\mathsf{b} \mid \mathsf{ab})^*$ einen ε-NFA an. Verwenden Sie hierbei das Verfahren aus der Vorlesung ohne Abänderung, d.h. Sie sollen weder die regulären Ausdrücke noch die Automaten, die Sie konstruieren, vereinfachen.
Hinweis: Es gibt 9 Teilausdrücke. Der Ausdruck selbst ist auch ein Teilausdruck.

(b) Überführen Sie den ε-NFA für den gesamten Ausdruck aus Teilaufgabe a) in einen DFA. Kombinieren Sie hierzu die Potenzmengenkonstruktion mit der Umwandlung eines ε-NFA in einen NFA. (Vorlesung Folie 47)

AUFGABE 2.8. Stufe C

Geben Sie eine rekursive Prozedur $\mathit{empty}(r)$ an, die für einen gegebenen regulären Ausdruck r entscheidet, ob $L(r) = \emptyset$. Für Ihre Definition sollten Sie das folgende Gerüst verwenden:

- $\mathit{empty}(\emptyset) =$
- $\mathit{empty}(\mathsf{a}) =$
- $\mathit{empty}(\varepsilon) =$
- $\mathit{empty}(\alpha\beta) =$
- $\mathit{empty}(\alpha \mid \beta) =$
- $\mathit{empty}(\alpha^*) =$

Beweisen Sie mittels struktureller Induktion, dass Ihre Definition korrekt ist.

[1] LoVe := Logic and Verification