

Einführung in die theoretische Informatik
Sommersemester 2018 – Übungsblatt 2

AUFGABE 2.1. (*Wichtige Begriffe*)

Stufe A

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

- DFA
- NFA
- akzeptierte Sprache
- Zustandsgraph
- Akeptanzbedingung von DFAs/NFAs
- Potenzmengenkonstruktion
- rechtslineare Grammatik

AUFGABE 2.2. (*Automata Tutor*)

Stufe B

AutomataTutor ist umgezogen. Sie finden es nun unter <http://vmesparza8.informatik.tu-muenchen.de>. Deshalb müssen Sie sich leider erneut registrieren.

- Die Aufgaben 2.3 und 2.4 stehen auf AutomataTutor zur Verfügung. Dort können Sie direkt sehen, ob sie die richtige Lösung haben und bekommen individuelle Hilfestellung beim Lösen der Aufgaben. Klicken Sie dafür nach Anmeldung/Registrierung auf "Practise Problem Set" und lösen sie die Probleme von Woche 2.

AUFGABE 2.3.

Stufe B

Geben Sie für jede der folgenden Sprachen über dem Alphabet $\Sigma = \{a, b, c\}$ einen NFA an, der genau die jeweilige Sprache erkennt:

- Alle Wörter, die `aaab` enthalten.
- Alle Wörter, deren 3-letzter Buchstabe ein `a` ist, z.B. `babb`. Der Automat sollte nicht mehr als 4 Zustände haben.

AUFGABE 2.4.

Stufe B

Geben Sie für jede der folgenden Sprachen über dem Alphabet $\Sigma = \{a, b, c\}$ einen DFA (graphisch) an, der genau die jeweilige Sprache erkennt:

- Alle Wörter, die mit einem `b` beginnen.
- Alle Wörter gerader Länge.
- Alle Wörter ungerader Länge, die auf ein `c` enden.
- Die Sprache, die nur das leere Wort enthält.
- Alle Wörter, die `aab` enthalten.
- Alle Wörter, die eine durch drei teilbare Anzahl von `c` enthalten.
- $L = \{aabcaa, aacaa, baa\}$
- Alle Wörter, deren 3-letzter Buchstabe ein `a` ist, z.B. `babb`.

Beschreiben Sie dann in eigenen Worten, wie man im Allgemeinen einen Automaten konstruiert, der eine Sprache erkennt, die ...

- am Anfang/Ende jedes Wortes eine bestimmte Sequenz von Buchstaben fordert
- nur Worte gerader/ungerader Länge enthält
- von jedem Wort verlangt, eine bestimmte Anzahl an Buchstaben zu enthalten
- die nur Worte enthält, die eine bestimmte Sequenz von Buchstaben enthält
- deren Worte an einer fixierten Position einen eigenen bestimmten Buchstaben haben müssen.

AUFGABE 2.5.

Stufe B / C

Sei $\Sigma = \{a, b\}$ und $A_k = \{w_1 \dots w_l \in \Sigma^* \mid w_{l-k} = a\}$ die Sprache aller Wörter über Σ , die an der $(k+1)$ -ten Stelle von rechts ein `a` stehen haben. Insbesondere ist A_0 die Menge aller Wörter, die auf `a` enden. *Versuchen Sie in beiden Aufgabenteilen NFAs und DFAs mit möglichst wenigen Zuständen anzugeben.*

- Geben Sie jeweils einen NFA für A_0 , A_1 und A_2 an.
- Geben Sie jeweils einen DFA für A_0 und A_1 an.
- Beschreiben Sie in eigenen Worten, wie der DFA A_n und der NFA A_n für beliebige $n \in \mathbb{N}$ aussehen.
- Beurteilen Sie die folgende Aussage: *Es gibt einen NFA für A_n mit $O(n)$ -vielen Zuständen, aber jeder DFA zu A_n hat mindestens $\Theta(2^n)$ -viele Zustände.*

Hinweis: Bearbeiten Sie diese Aufgabe auch für die Sprache $B_n := \{w \in \Sigma^* \mid \exists i : w_i = w_{i+n}\}$, um die aus der Vorlesung bekannten Konzepte auf eine neue Sprache anzuwenden (A_k findet sich bereits in den Vorlesungsfolien!).

AUFGABE 2.6.

Sei $\Sigma = \{a, b, c\}$ und $L = \{w \in \Sigma^* \mid \exists x \in \Sigma. |w|_x = 0\}$.

- Konstruieren Sie einen NFA N mit genau 4 Zuständen und $L(N) = L$.
- Determinisieren Sie den NFA N aus (a) mittels der Potenzmengenkonstruktion.
- Geben Sie eine rechtslineare Grammatik G (gemäß Satz 3.7) an, so dass $L(G) = L(D)$.
- Übersetzen Sie die Grammatik G (gemäß Satz 3.11) in einen NFA N' , so dass $L(G) = L(N')$.
- Vergleichen Sie die NFAs N' und N bezüglich Zustands- und Transitionszahl. Diskutieren Sie dann, inwiefern Sie Ihre Beobachtung verallgemeinern können.

Hinweis: Wir bezeichnen mit $|w|_x$ die Anzahl der Vorkommen des Buchstabens $x \in \Sigma$ in $w \in \Sigma^*$.

AUFGABE 2.7.

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 $N = (Q, \Sigma, \delta, q_0, F)$ gibt es einen NFA $N' = (Q', \Sigma', \delta', q'_0, F')$ mit $L(N) = L(N')$ und ...

- der Startzustand hat keine eingehenden Kanten.
- kein Endzustand hat eine ausgehende Kante.
- für jeden Zustand gilt: alle eingehenden Kanten sind mit demselben Zeichen beschriftet.
- für jeden Zustand gilt: alle ausgehenden Kanten sind mit demselben Zeichen beschriftet.

AUFGABE 2.8.

Wir erweitern NFAs um die Möglichkeit, mehrere Startzustände $I \subseteq Q$ zu besitzen. Ein NFA $N = (Q, \Sigma, \delta, I, F)$ akzeptiert ein Wort $w \in \Sigma^*$ genau dann, wenn

$$\emptyset \neq F \cap \bigcup_{q_0 \in I} \hat{\delta}(q_0, w),$$

d.h. wenn es mindestens einen akzeptierenden Ablauf von irgendeinem Startzustand aus gibt.

- Geben Sie einen NFA $N' = (Q', \Sigma', \delta', I', F')$ an, der die Sprache L aus Aufgabe 2.6 erkennt, aber $|Q'| < 4$.
- Entscheiden Sie, ob die folgenden Aussagen korrekt sind oder nicht und begründen Sie Ihre Behauptung, indem Sie entweder ein Gegenbeispiel oder einen passenden Beweis angeben.

Für jeden NFA $N = (Q, \Sigma, \delta, I, F)$ existiert ein NFA $N' = (Q', \Sigma', \delta', I', F')$ mit $L(N) = L(N')$, so dass

- $|I'| = 1$.
- $|F'| = 1$.
- $|I'| = 1$ und $|F'| = 1$.

Definition (Substring, Superstring und Superstringsprache)

Sei Σ ein Alphabet und seien v, w Wörter über Σ^* .

- v ist ein *Substring* von w genau dann, wenn es Wörter $v_1, v_2 \in \Sigma^*$ gibt mit $w = v_1 v v_2$. Wir schreiben $v \trianglelefteq w$ für v ist ein Substring von w .
- Wir nennen w *Superstring* von v , wenn $v \trianglelefteq w$.
- Die *Superstring-Sprache* für ein Wort $v \in \Sigma^*$ ist definiert als $L_{v \trianglelefteq} := \{u \in \Sigma^* \mid v \trianglelefteq u\}$

AUFGABE 2.9.

Sei $\Sigma = \{m, i, a\}$ das Alphabet.

- Sei $w = mia$ und $v = mammamia$.
 - Geben Sie einen NFA an, der die Sprache $L_{w \trianglelefteq}$ akzeptiert.
 - Geben Sie einen DFA an, der die Sprache $L_{w \trianglelefteq}$ akzeptiert und nicht mehr Zustände als Ihr NFA aus der vorherigen Teilaufgabe hat.
 - Geben Sie einen NFA an, der die Sprache $L_{v \trianglelefteq}$ akzeptiert.
 - Geben Sie einen DFA an, der die Sprache $L_{v \trianglelefteq}$ akzeptiert und dabei nicht mehr Zustände als Ihr NFA aus der vorherigen Teilaufgabe hat.
- Beschreiben Sie die Unterschiede und Gemeinsamkeiten der beiden NFAs und DFAs aus der vorherigen Teilaufgabe. Erklären Sie in möglichst einfacher Sprache, wie Sie es erreicht haben, dass die NFAs und DFAs jeweils die gleiche Anzahl an Zuständen hatten.
- Entwickeln Sie eine allgemeine Konstruktionsvorschrift für NFAs und DFAs, die die Sprache $L_{w \trianglelefteq}$ für ein beliebiges Wort $w \in \Sigma^*$. Nutzen Sie dafür ihre Beobachtungen aus den vorherigen Aufgabenteilen.