

Übung 4 zur Vorlesung Formale Sprachen und Komplexität

Wenn Sie Automaten angeben, tun Sie dies immer in Form eines Zustandsgraphen. Andere Formen der Darstellung (z.B. als Liste von Übergängen) werden nicht gewertet, da sie sehr viel aufwändiger zu korrigieren sind. Vergessen Sie nicht, im Zustandsgraph Start- und Endzustände zu markieren.

Reguläre Ausdrücke sind entsprechend Definition 4.7.1 im Vorlesungsskript anzugeben.

FSK4-1 Pumping-Lemma für reguläre Sprachen (2 Punkte)

Zeigen Sie mit dem Pumping-Lemma für reguläre Sprachen, dass die folgenden Sprachen nicht regulär sind.

- a) $L_1 = \{a^i b^j c^k d^i \mid i, j, k \in \mathbb{N}_{>0}\}$ über dem Alphabet $\Sigma_1 = \{a, b, c, d\}$.
b) $L_2 = L(G_2)$, wobei G_2 eine kontextfreie Grammatik ist mit

$$G_2 = (\{S, A, B\}, \{(\,), [,]\}, P, S) \\ P = \{S \rightarrow (S), S \rightarrow [S], S \rightarrow A, S \rightarrow B, A \rightarrow (), A \rightarrow [], B \rightarrow S, B \rightarrow BB\}$$

L_2 ist die Sprache der zueinander passenden eckigen und runden Klammern, d.h. es sind z.B. $([]) \in L_2$ und $()() \in L_2$, aber $([] \notin L_2$ und $) \notin L_2$ (vgl. Aufgabe FSK1-3).

FSK4-2 Reguläre Ausdrücke und Abschlusseigenschaften (2 Punkte)

- a) Betrachten Sie den regulären Ausdruck $\alpha = (a|b)^*(ab|ba)(a|b)^*$.
- Geben Sie einen NFA ohne ε -Übergänge an, der $L(\alpha)$ erkennt. Sie können die Algorithmen aus der Vorlesung zur Konstruktion eines NFA aus einem regulären Ausdruck und zur Elimination von ε -Übergängen verwenden, müssen aber nicht.
 - Geben Sie einen DFA an, der $L(\alpha)$ erkennt. Sie können die Potenzmengenkonstruktion verwenden, müssen aber nicht.
- b) Geben Sie reguläre Ausdrücke an, die die folgenden Sprachen erkennen.

- i) Die Sprache L_3 der Wörter über dem Alphabet $\Sigma_1 = \{a, b, c\}$, die mit a oder b anfangen und mindestens ein c enthalten.
 - ii) Die Sprache L_4 der Wörter über dem Alphabet $\Sigma_2 = \{a, b\}$, die keine zwei a 's hintereinander enthalten.
- c) Zeigen Sie mithilfe der Abschlusseigenschaften regulärer Sprachen, dass die Sprache $L_5 = \{a^i w d^{i+1} \mid i \in \mathbb{N}, w \in \Sigma_3^*\}$ über dem Alphabet $\Sigma_3 = \{a, b, c, d\}$ nicht regulär ist. Sie dürfen annehmen, dass die Sprache L_1 aus Aufgabe FSK4-1a) nicht regulär ist.

FSK4-3 Grammatik über Automaten zu Grammatik

(0 Punkte)

Gegeben sei die reguläre Grammatik

$$G = (\{S, A, B, C\}, \{a, b\}, \{S \rightarrow aA \mid bB, A \rightarrow bB, B \rightarrow bC, C \rightarrow aC \mid a\}, S)$$

- a) Erzeugen Sie gemäß der Konstruktion aus der Vorlesung aus G einen NFA A mit $L(G) = L(A)$.
- b) Erzeugen Sie mit der Potenzmengenkonstruktion aus A einen DFA B mit $L(B) = L(A)$. Geben Sie nur den vom Startzustand erreichbaren Teil von A an.
- c) Erzeugen Sie gemäß der Konstruktion aus der Vorlesung aus B eine Grammatik H mit $L(B) = L(H)$.
- d) Vergleichen Sie die Grammatiken G und H . Beschreiben Sie die Gemeinsamkeiten dieser Grammatiken, sowie ihre Unterschiede.

Überlegen Sie sich, wodurch diese Effekte zustande kommen.

FSK4-4 DNA-Analyse mit NFA

(0 Punkte)

Diese Aufgabe handelt von der Analyse von Desoxyribonukleinsäure (DNS/DNA) mithilfe von NFA. DNA ist eine Abfolge der Basen Adenin, Thymin, Guanin und Cytosin, typischerweise mit A, T, G und C abgekürzt. Dementsprechend ist das Alphabet aller Automaten in dieser Aufgabe $\Sigma = \{A, C, G, T\}$.

- a) Um das Vorkommen einer Basensequenz zu finden, wird aus dieser Sequenz ein NFA erzeugt, der alle Wörter akzeptiert, in denen diese Sequenz als Teilwort vorkommt.

Geben Sie einen NFA B an, der genau diejenigen Wörter akzeptiert, in denen $ACTC$ als Teilwort vorkommt.

Hinweis: Sie können (müssen aber nicht) dazu den regulären Ausdruck $(A|C|G|T)^*ACTC(A|C|G|T)^*$ verwenden, der genau diese Sprache akzeptiert.

Hinweis: Sie können auch einen DFA angeben, aber ein NFA ist übersichtlicher.

- b) Beim Kopieren von DNA kann es vorkommen, dass Fehler auftreten. Zum Beispiel kann eine Base durch eine andere ersetzt werden; es kann eine Base ausgelassen werden; es kann eine zusätzliche Base eingefügt werden; und es können auch komplexere Fehler auftreten. Zur Vereinfachung behandeln wir hier nur den Fall, dass eine Base durch eine andere ersetzt wird.

Aus einem NFA $D = (Z, \Sigma, \delta, S, E)$ kann ein NFA $F = (Z', \Sigma, \delta', S', E')$ erzeugt werden, der alle Wörter akzeptiert, die durch höchstens k fehlerhafte Ersetzungen aus D entsteht.

Dabei sind

- $Z' = Z \times \{0, \dots, k\}$
- $\delta'((q, i), a) = \{(q', i) \mid q' \in \delta(q, a)\} \cup \{(q', i+1) \mid (\exists b \in \Sigma. q' \in \delta(q, b) \wedge i+1 \leq k)\}$
- $S' = S \times \{0\} = \{(s, 0) \mid s \in S\}$
- $E' = E \times \{0, \dots, k\}$

Berechnen Sie mit der obigen Konstruktion einen NFA H aus B , der Wörter mit bis zu 2 Fehlern akzeptiert.

- c) Geben Sie an und begründen Sie, welche der folgenden Wörter von H akzeptiert werden. Prüfen Sie, ob Ihr Ergebnis korrekt ist, also ob die erkannten Wörter tatsächlich diejenigen sind, bei denen bis auf höchstens 2 Fehler das Wort *ACTC* als Teilwort vorkommt.

AAAACCCAAA, GAGGCGT, TAGCA, TCTCA

- d) Begründen Sie, dass die Konstruktion aus b) korrekt ist, also tatsächlich für jeden NFA D und jedes k einen NFA F liefert, der maximal k Fehler zulässt.

Hinweis: Sie können auch als Vorüberlegung dies erst für den NFA H zeigen.