technische universität dortmund

Thomas Schwentick

Jonas Schmidt, Jennifer Todtenhoefer
Erik van den Akker

Lehrstuhl 1
Logik in der Informatik

SoSe 2024 | **Warm-Up Blatt 4** | **06.05.2024**

Warm-Up-Aufgabe 4.1 [Zeichenkettensuche]

Wir betrachten die Problemstellung MultiSearch aus dem Vorlesungskapitel 6. Ein Text sei gegeben als ein Wort über dem Alphabet $\{a, \ldots, z\}$. Es soll überprüft werden, ob eine der Zeichenketten *paprika*, *apricot* oder *prima* im Text enthalten ist.

a) Geben Sie einen NFA an, der einen Text genau dann akzeptiert, wenn dieser eines der gesuchten Wörter enthält. Bezeichnen Sie dabei die Zustände des Automaten wie im Abschnitt 6.2 der Vorlesung („NFA zur Zeichenkettensuche: allgemein") mit dem bisher gelesenen Präfix.

b) Leiten Sie anschließend einen äquivalenten DFA wie in der Vorlesung beschrieben her. Die Herleitung soll direkt, wie in Abschnitt 6.2 der Vorlesung, erfolgen und *nicht* über die Potenzmengen-Konstruktion.

> **Hinweis**
>
> Der Übersicht halber müssen nicht alle Transitionen eingezeichnet werden. So können Sie beispielsweise schreiben „für alle Zustände q, für die keine ausgehende Transition für das Zeichen n eingezeichnet ist, gilt $\delta(q, n) = p$", wobei p ein Zustand in Ihrem Automaten ist, zu dem die Transitionen führen sollen.

c) Aus dem in Teilaufgabe b) konstruierten DFA wollen wir jetzt einen partiellen DFA für die Wortsuche erstellen, welcher die folgenden Bedingungen erfüllt:

- Alle Transitionen, die in den Startzustand führen, sollen weggelassen werden
- Alle Transitionen, die von einem Zustand, der nicht der Startzustand ist, in einen Zustand, der einem Präfix der Länge 1 entspricht, sollen weggelassen werden
- Alle Transitionen von akzeptierenden Zuständen zu sich selbst sollen weggelassen werden

Konstruieren Sie einen DFA, der die oben angegebenen Bedingungen erfüllt.

> **Hinweis**
>
> Diese Aufgabe soll auf die Reduktion des DFA aus **b)** für die Eingabe der Iltis Aufgabe 4.3.b vorbereiten.
> Der resultierende partielle DFA entscheidet **nicht** mehr dieselbe Sprache wie der DFA in **b)**!
> Ein Beispiel für einen solchen partiellen DFA ist auf Folie 6.17 angegeben.

Warm-Up-Aufgabe 4.2 [Kontextfreie Grammatiken: Interpretation und Eindeutigkeit]

Sei G die folgende kontextfreie Grammatik.

$$\begin{aligned} S &\to C \mid D \\ C &\to aCb \mid aC \mid a \\ D &\to aDb \mid Db \mid b \end{aligned}$$

a) Beschreiben Sie die Sprache $L(G)$ erst informell und dann formal (Mengennotation). Begründen Sie warum Ihre Beschreibung korrekt ist, indem Sie auf die Bedeutung der einzelnen Nichtterminalsymbole eingehen.

b) Geben Sie eine Ableitung und einen Ableitungsbaum für das Wort $w = aaabbbbb$ über G an.

c) Ist die Grammatik G eindeutig? Begründen Sie ihre Aussage.

Warm-Up-Aufgabe 4.3 [Kontextfreie Grammatiken: Konstruktion]

a) Wir betrachten die folgenden Sprachen L_1, L_2:

$$\begin{aligned} L_1 &= \{a^i b^i c^j d^j \mid i, j \in \mathbb{N}\} \\ L_2 &= \{a^i b^j c^j d^i \mid i, j \in \mathbb{N}\} \end{aligned}$$

Konstruieren Sie eine kontextfreie Grammatik für die Sprache $L_1 \cup L_2$

> **Hinweis**
>
> Hier ist es hilfreich sich zu überlegen, wie eine Regel für die Vereinigung von zwei Sprachen aussehen könnte.

b) Wir betrachten die Sprache $L_3 = \{a^i b^j c^k \mid i, j, k \in \mathbb{N}_0, j > i + k\}$.
Konstruieren Sie eine kontextfreie Grammatik für die Sprache L_3

Warm-Up-Aufgabe 4.4 [Grammatiken für reguläre Sprachen]

Sie haben in der Vorlesung gelernt, dass jede reguläre Sprache auch eine kontextfreie Sprache ist. Insbesondere lässt sich jede reguläre Sprache durch eine kontextfreie Grammatik beschreiben.
Da aber nicht jede kontextfreie Sprache regulär ist, stellt sich die Frage, ob für reguläre Sprachen nicht eine eingeschränkte Form kontextfreier Grammatiken ausreicht.
Eine solche eingeschränkte Klasse von Grammatiken gibt es in der Tat: die rechtslinearen Grammatiken.

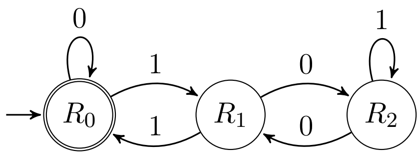
Definition: Eine Grammatik $G = (V, \Sigma, S, P)$ heißt *rechtslinear* wenn jede Regel in P von der Form

$$\begin{aligned} & A \to \sigma \\ & A \to \varepsilon \\ \text{oder} \quad & A \to \sigma B \end{aligned}$$

mit $A, B \in V$ und $\sigma \in \Sigma$ ist.

Wir wollen hier einmal an zwei Beispielen intuitiv zeigen, wie Automaten in rechtslineare Grammatiken und rechtslineare Grammatiken in Automaten umgewandelt werden können.

a) Betrachten Sie den folgenden DFA für die reguläre Sprache L_{drei} (bekannt von Folie 12 in Kapitel 5). Entwickeln Sie aus diesem Automaten eine rechtslineare Grammatik für L_{drei} und beschreiben Sie kurz Ihr Vorgehen.



b) Wir betrachten die folgende rechtslineare Grammatik G.

$$
\begin{aligned}
S &\to 0A|0E \\
A &\to 0B \mid 1C \\
B &\to 0A \mid 1D \\
C &\to 0D \mid 1A \\
D &\to 0C \mid 1B \mid \varepsilon \\
E &\to 0E \mid 0
\end{aligned}
$$

Konstruieren Sie einen ε-NFA, welcher die Sprache $L(G)$ entscheidet. Beschreiben Sie die Sprache $L(G)$ prägnant in wenigen umgangssprachlichen Sätzen.