

Übung 3 zur Vorlesung Theoretische Informatik für Studierende der Medieninformatik

Hinweis:

Die letzte Aufgabe auf diesem Blatt ist eine Aufgabe zur Klausurvorbereitung. Diese Aufgabe orientiert sich in Form und inhaltlichen Schwerpunkten an den Klausuraufgaben. Dies bedeutet jedoch nicht, dass die anderen Aufgaben nicht klausurrelevant sind.

Die Lösungen der Klausurvorbereitungs-Aufgaben werden am Ende der Bearbeitungszeit gesondert veröffentlicht, aber **nicht** im Tutorium besprochen. Die Lösungen der anderen Aufgaben werden bereits zu Beginn der Bearbeitungszeit veröffentlicht und können Ihnen bei der Bearbeitung helfen.

Wenn Sie Ihre Lösung innerhalb der Bearbeitungszeit über Moodle abgeben, erhalten Sie eine individuelle Korrektur. Die Abgabe ist freiwillig (aber nachdrücklich empfohlen).

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.

TIMI3-1 Konstruktion von NFAs

Verwenden Sie in dieser Aufgabe nur NFAs *ohne* ϵ -Übergänge.

- a) Viele Programmiersprachen erlauben nur Variablennamen, die Regeln wie diese erfüllen:
- Ein Variablenname kann Unterstriche, kleine und große Buchstaben (a–z, A–Z) und Ziffern enthalten.
 - Ein Variablenname muss mindestens ein Zeichen enthalten.
 - Ein Variablenname darf nicht mit einer Ziffer anfangen.
 - „_“ ist kein Variablenname.

Geben Sie einen NFA an, der genau die Variablennamen erkennt, die diesen Regeln folgen.

b) Sei n eine natürliche Zahl, $\Sigma_n = \{0, \dots, n\}$ und

$$L_n = \{w \in \Sigma_n^* \mid \exists i \in \Sigma_n, \#_i(w) = i\}$$

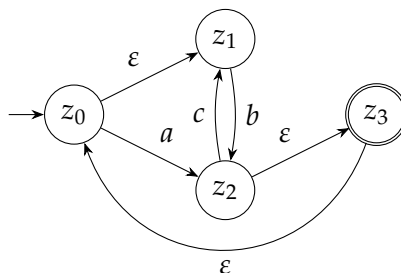
Das heißt, die Sprache L enthält genau die Wörter w , für die gilt: Es gibt eine Zahl $i \in \{0, \dots, n\}$ sodass das Wort w das Symbol i genau i -mal enthält.

Z.B. ist $2012323 \in L_3$, da dieses Wort genau 1-mal das Symbol 1 enthält. Ebenso ist $20311233 \in L_3$, da dieses Wort genau 2-mal das Symbol 2 enthält. Hingegen ist $0112223 \notin L_3$.

Geben Sie für jedes n einen NFA A_n an, der L_n erkennt. Beschreiben Sie ausnahmsweise A_n nicht durch einen Zustandsgraph, sondern geben Sie die Zustandsmenge, Start- und Endzustände und Übergänge (in Abhängigkeit von n) explizit an. Geben Sie außerdem den Zustandsgraph von A_3 an.

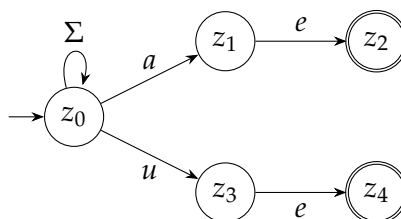
TIMI3-2 Entfernen von ε -Übergängen und Potenzmengenkonstruktion

a) Sei A_1 der folgende NFA über dem Alphabet $\{a, b, c\}$:



Geben Sie einen NFA A'_1 ohne ε -Übergänge mit $L(A'_1) = L(A_1)$ an. Verwenden Sie den Algorithmus zum Entfernen von ε -Übergängen aus der Vorlesung. Geben Sie die Zwischenschritte Ihrer Berechnung an. Das erlaubt uns, Ihnen für Folgefehler Teilpunkte zu geben.

b) Der folgende NFA A_2 über einem Alphabet $\Sigma \supseteq \{a, e, u\}$ kann verwendet werden, um in einem Text nach den Zeichenfolgen ae und ue zu suchen.



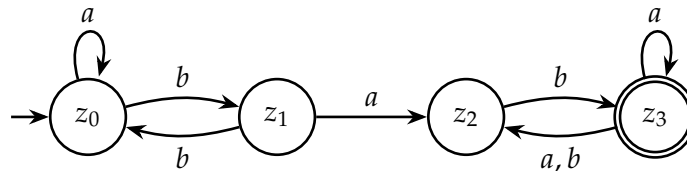
Die Suche wird wesentlich beschleunigt, wenn wir A_2 in einen DFA umwandeln. Verwenden Sie deshalb die Potenzmengenkonstruktion, um einen DFA A'_2 mit $L(A'_2) = L(A_2)$ zu konstruieren. Geben Sie außer dem Zustandsgraph von A'_2 auch die Rechenschritte an, die Sie bei der Potenzmengenkonstruktion ausgeführt haben. Das erlaubt uns, Ihnen bei Folgefehlern noch Teilpunkte zu geben.

Klausurvorbereitung TIMI-3-K

- a) Geben Sie einen NFA an, der die folgende Sprache L über dem Alphabet $\Sigma = \{a, b\}$ akzeptiert:

$$L = \{uvw \mid u, w \in \Sigma^*, v \in \{bab, aa\}\}$$

- b) Für diese Aufgabe betrachten wir folgenden NFA C über dem Alphabet $\{a, b\}$:



Berechnen Sie zum NFA C einen äquivalenten DFA D mit der Potenzmengenkonstruktion. Geben Sie nur den Zustandsgraphen des vom Startzustand erreichbaren Teils des Automaten an. Vergessen Sie nicht, den Startzustand und die Endzustände zu markieren.