

Lösungsvorschlag zur Übung 3 zur Vorlesung
Theoretische Informatik für Studierende der Medieninformatik

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.

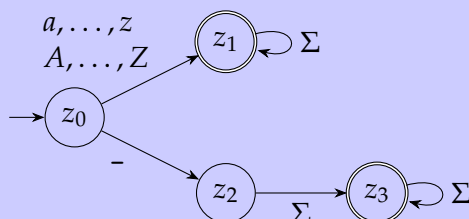
- a) Viele Programmiersprachen erlauben nur Variablennamen, die Regeln wie diese erfüllen:
- Ein Variablenname kann Unterstriche, kleine und große Buchstaben (a–z, A–Z) und Ziffern enthalten.
 - Ein Variablenname muss mindestens ein Zeichen enthalten.
 - Ein Variablenname darf nicht mit einer Ziffer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

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

LÖSUNGSVORSCHLAG:

Wir definieren einen NFA über dem Alphabet

$$\Sigma = \{_, a, \dots, z, A, \dots, Z, 0, \dots, 9\}$$



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

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

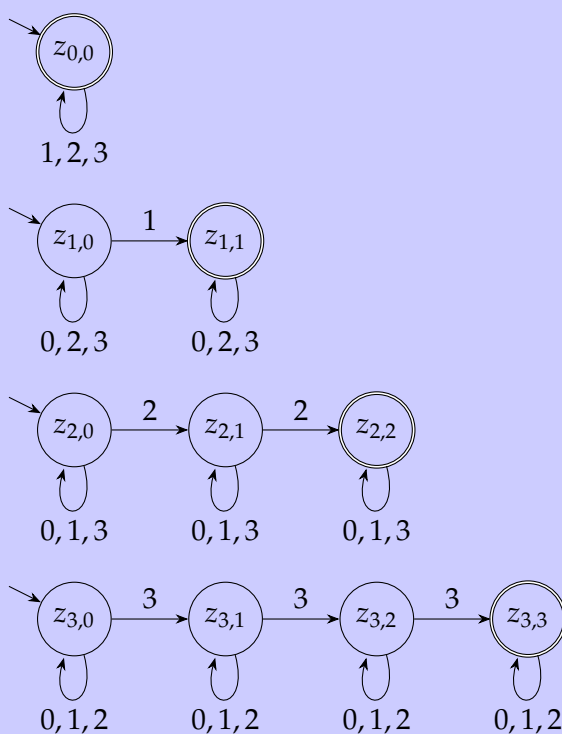
Das heißt, die Sprache L enthält genau die Wörter w , für die gilt: Es gibt eine Zahl $i \in \{0, \dots, n\}$ sodass das Wort w das Symbol i genau i -mal enthält.

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 die Übergangsfunktion (in Abhängigkeit von n) explizit an. Geben Sie außerdem den Zustandsgraph von A_3 an.

LÖSUNGSVORSCHLAG:

Der folgende Automat A_3 erkennt L_3 :

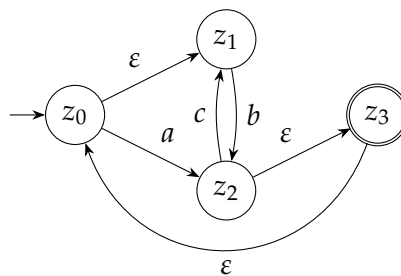


Die Konstruktion für beliebige n verallgemeinert diesen Automaten:

$$\begin{aligned}
 A_n &= (Z_n, \Sigma_n, \delta_n, S_n, E_n) \\
 Z_n &= \{z_{i,0}, \dots, z_{i,i} \mid i \in \Sigma_n\} \\
 S_n &= \{z_{i,0} \mid i \in \Sigma_n\} \\
 E_n &= \{z_{i,i} \mid i \in \Sigma_n\} \\
 \delta_n(z_{i,j}, k) &= \begin{cases} \{z_{i,j+1}\} & \text{für } i \in \Sigma_n, k = i, j \in \{0, \dots, i-1\} \\ \{z_{i,j}\} & \text{für } i, j \in \Sigma_n, k \in \Sigma_n \setminus \{i\} \\ \emptyset & \text{sonst} \end{cases}
 \end{aligned}$$

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.

LÖSUNGSVORSCHLAG:

Die Startzustände von A'_1 sind die Zustände von A_1 , die von z_0 aus nur mit ε -Übergängen erreichbar sind. Diese Bedingung wird (außer von z_0) nur von z_1 erfüllt. Die Endzustände von A'_1 sind die von A_1 .

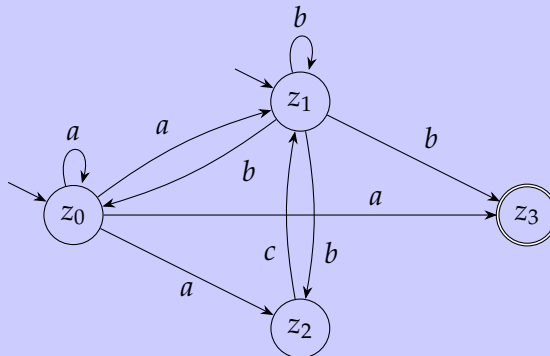
Als Übergänge von A'_1 verwenden wir zunächst alle Übergänge aus A_1 außer den ε -Übergängen. Weiterhin betrachten wir alle Übergangsfolgen in A_1 von der Form

$$z_{i_1} \xrightarrow{d} z_{i_2} \xrightarrow{\varepsilon} z_{i_3} \xrightarrow{\varepsilon} \dots \xrightarrow{\varepsilon} z_{i_n}$$

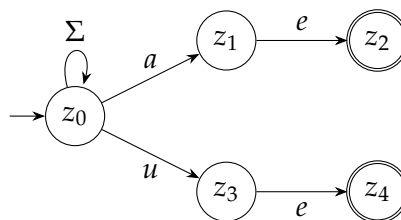
Für jede solche Folge fügen wir die Übergänge $z_{i_1} \xrightarrow{d} z_{i_2}$, $z_{i_1} \xrightarrow{d} z_{i_3}$, \dots , $z_{i_1} \xrightarrow{d} z_{i_n}$ zu A'_1 hinzu. Es ergeben sich folgende Übergänge:

- Aus der Übergangsfolge $z_0 \xrightarrow{a} z_2 \xrightarrow{\varepsilon} z_3 \xrightarrow{\varepsilon} z_0 \xrightarrow{\varepsilon} z_1$ erhalten wir $z_0 \xrightarrow{a} z_3$, $z_0 \xrightarrow{a} z_0$ und $z_0 \xrightarrow{a} z_1$.
- Aus der Übergangsfolge $z_1 \xrightarrow{b} z_2 \xrightarrow{\varepsilon} z_3 \xrightarrow{\varepsilon} z_0 \xrightarrow{\varepsilon} z_1$ erhalten wir $z_1 \xrightarrow{b} z_3$, $z_1 \xrightarrow{b} z_0$ und $z_1 \xrightarrow{b} z_1$.

Insgesamt ist A'_1 also:



- 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.



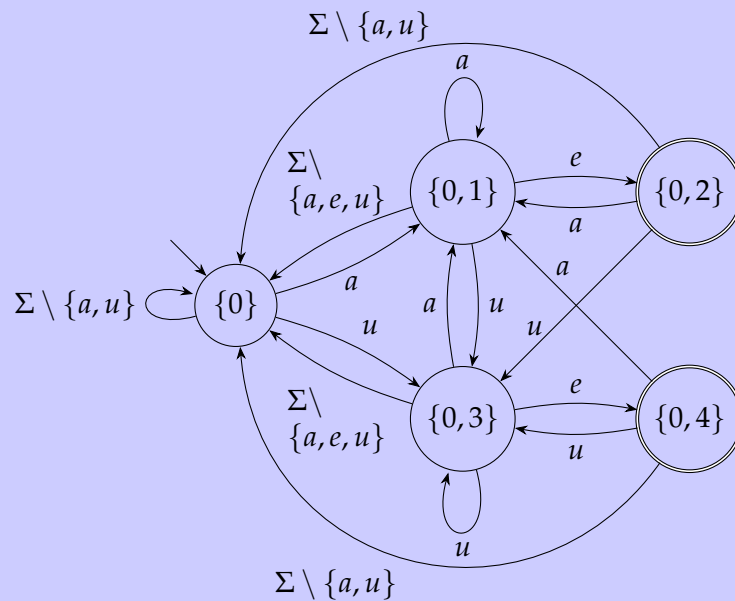
Die Suche wird wesentlich beschleunigt, wenn wir A_2 in einen DFA umwandeln. Verwenden Sie deshalb die Potenzmengenkonstruktion, um einen DFA A'_2 mit $L(A'_2) = L(A_2)$ zu konstruieren. Geben Sie außer dem Zustandsgraph von A'_2 auch die Rechenschritte an, die Sie bei der Potenzmengenkonstruktion ausgeführt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

LÖSUNGSVORSCHLAG:
Potenzmengenkonstruktion:

Start	→	Ziel	Start	→	Ziel
{0}	a	{0,1}	{0,2}	a	{0,1}
{0}	u	{0,3}	{0,2}	u	{0,3}
{0}	$\Sigma \setminus \{a,u\}$	{0}	{0,2}	$\Sigma \setminus \{a,u\}$	{0}
{0,1}	a	{0,1}	{0,4}	a	{0,1}
{0,1}	u	{0,3}	{0,4}	u	{0,3}
{0,1}	e	{0,2}	{0,4}	$\Sigma \setminus \{a,u\}$	{0}
{0,1}	$\Sigma \setminus \{a,e,u\}$	{0}			
{0,3}	a	{0,1}			
{0,3}	u	{0,3}			
{0,3}	e	{0,4}			
{0,3}	$\Sigma \setminus \{a,e,u\}$	{0}			

Die Tabelle enthält die Zustände und Übergänge von A'_2 . Startzustand ist {0}. Endzustände sind die Zustände, die 2 und 4 enthalten (da q_2 und q_4 im ursprünglichen Automaten Endzustände waren), also {0,2} und {0,4}.

Daraus ergibt sich der Zustandsgraph von A'_2 :



Nebenbei: A_2 erkennt nicht die Sprache der Wörter, die ae oder ue enthalten, sondern die Sprache der Wörter, die mit ae oder ue enden. Wir können A_2 aber trotzdem verwenden, um Wörter zu erkennen, die ae oder ue enthalten, indem wir A_2 auf einem gegebenen Wort ausführen und bestimmen, ob A_2 jemals in einen Endzustand kommt.