

FGI-1 – Formale Grundlagen der Informatik I

Logik, Automaten und Formale Sprachen

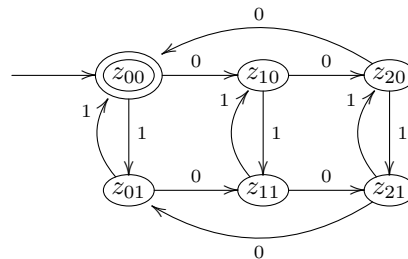
Musterlösung 2: Endliche Automaten und reguläre Sprachen

Präsenzteil am 14.-17. April – Abgabe am 21.-24. April 2015

Präsenzaufgabe 2.1: Geben Sie einen DFA A an, der die Sprache L aller Worte $w \in \{0,1\}^*$ akzeptiert, bei denen die Anzahl der 0en durch 3 und die Anzahl der 1en durch 2 teilbar ist (jeweils ohne Rest, d.h. ein Wort w soll genau dann akzeptiert werden, wenn $|w|_0$ ein Vielfaches von 3 und $|w|_1$ ein Vielfaches von 2 ist).

Zeigen Sie, dass Ihr Automat das Gewünschte leistet, indem Sie zwei Mengeninklusionen beweisen.

Lösung: In dem Zustand z_{xy} merken wir uns jeweils die Zahl $|w|_0 \bmod 3 = x$ bzw. $|w|_1 \bmod 2 = y$ des bisher gelesenen Teilwortes w , d.h. den Rest bei Division durch 3 bzw. 2. Oder anders: Bei z_{xy} werden mit dem x die 0en (modulo 3) gezählt und mit dem y die 1en (modulo 2). Man kann an den Kantenübergängen des Zustandsübergangsdiagramms schnell ablesen, dass dies tatsächlich gegeben ist: Der Nullzähler erhöht sich immer um eins (modulo 3), wenn eine 0 gelesen wird und der Einszähler immer um eins (modulo 2), wenn eine 1 gelesen wird. Der einzige Endzustand ist z_{00} .

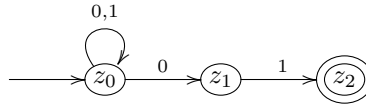


Mit der Vorüberlegung oben, gelingt der Beweis schnell:

Ist $w \in L$, so ist $|w|_0 \bmod 3 = 0$ und $|w|_1 \bmod 2 = 0$, d.h. der Automat A liest das Wort w nach Konstruktion vollständig und ist dann im Zustand z_{00} . Dort akzeptiert er, also gilt auch $w \in L(A)$.

Ist umgekehrt $w \in L(A)$, so muss das Lesen von w den Automaten A in den Zustand z_{00} überführen, da dies der einzige Endzustand ist. Dies ist aber gleichbedeutend damit, dass nach Konstruktion von A auch $|w|_0 \bmod 3 = 0$ und $|w|_1 \bmod 2 = 0$, also $w \in L$ gilt.

Präsenzaufgabe 2.2: Gegeben sei der folgende NFA A :



1. Welche Sprache akzeptiert dieser NFA (ohne Beweis)?

Lösung: Es ist $L(A) = \{0, 1\}^* \cdot \{0\} \cdot \{1\}$. A akzeptiert also genau die Wörter (über $\{0, 1\}$), die mit 01 enden. Man kann dies auch recht schnell beweisen:

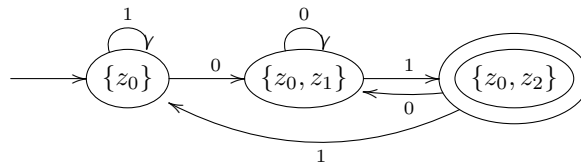
- $\{0, 1\}^* \cdot \{01\} \subseteq L(A)$. Sei $w \in \{0, 1\}^* \cdot \{01\}$, dann gilt $w = v01$ mit $v \in \{0, 1\}^*$. Es gilt nun $(z_0, v01) \vdash^* (z_0, 01) \vdash (z_1, 1) \vdash (z_2, \lambda)$ und da $z_2 \in Z_{end}$ wird w akzeptiert.
 - $L(A) \subseteq \{0, 1\}^* \cdot \{01\}$. Sei $w \in L(A)$, dann muss w auf 01 enden, da sonst der einzige Endzustand z_2 nicht erreicht wird. Dies ist nur über die Zustände z_0 und z_1 möglich. In z_0 kann vorher ein beliebiges Wort gelesen werden, d.h. w ist von der Form $\{0, 1\}^* \cdot \{01\}$ wie gewünscht.
2. Geben Sie **alle** Rechnung von A auf dem Wort 01101 an. Geben sie zudem die Menge der nach jedem Buchstaben erreichten Zustände an (also formal die Menge $\hat{\delta}(z_0, v)$ für jedes Präfix v von 01101).

Lösung: Es gibt eine akzeptierende und zwei nicht akzeptierende Rechnungen: $(z_0, 01101) \vdash (z_0, 1101) \vdash (z_0, 101) \vdash (z_0, 01) \vdash (z_1, 1) \vdash (z_2, \lambda)$ und $(z_0, 01101) \vdash (z_0, 1101) \vdash (z_0, 101) \vdash (z_0, 01) \vdash (z_0, 1) \vdash (z_0, \lambda)$ und $(z_0, 01101) \vdash (z_1, 1101) \vdash (z_2, 101)$.

Zudem gilt: $\hat{\delta}(z_0, 0) = \{z_0, z_1\}$, $\hat{\delta}(z_0, 01) = \{z_0, z_2\}$, $\hat{\delta}(z_0, 011) = \{z_0\}$, $\hat{\delta}(z_0, 0110) = \{z_0, z_1\}$ und $\hat{\delta}(z_0, 01101) = \{z_0, z_2\}$.

3. Konstruieren Sie mittels der Potenzautomatenkonstruktion einen DFA B mit $L(B) = L(A)$ (d.h. einen äquivalenten DFA).

Lösung: Anwenden der Konstruktion ergibt den folgenden DFA B (wobei wir nur die initiale Zusammenhangskomponente konstruieren, d.h. ausgehend von $\{z_0\}$ die Zustände, die tatsächlich erreichbar sind).



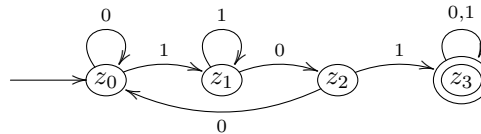
Da die Richtigkeit des Verfahrens bewiesen ist, müssen wir nun nicht mehr extra zeigen, dass tatsächlich $L(B) = L(A)$ gilt, sondern dürfen dies aus der Richtigkeit des Verfahrens folgern.

Übungsaufgabe 2.3: Geben Sie einen DFA an, der alle Zeichenketten über dem Alphabet $\{0, 1\}$ akzeptiert, die 101 als Teilwort enthalten. Bspw. soll also 11010 akzeptiert werden, da 101 vom zweiten bis vierten Buchstaben auftritt.

VON
4

Zeigen Sie insbesondere, dass Ihr Automat das Gewünschte leistet, indem Sie zwei Mengeneinklusionen beweisen.

Lösung: Der DFA A sei durch folgendes Zustandsdiagramm gegeben:

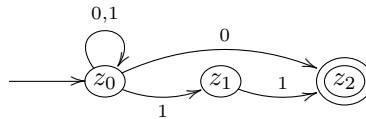


Intuitiv ist z_0 der Zustand, in dem wir noch nichts Hilfreiches gelesen haben. z_1 ist der Zustand, in dem wir die 1 (den ersten Buchstaben unseres gesuchten Wortes) gelesen haben. z_2 ist der Zustand, in dem wir 10 (die ersten zwei Buchstaben unseres gesuchten Wortes) gelesen haben. Und z_3 ist der Zustand, in dem wir das gesuchte Wort 101 gefunden haben. Eine andere (sinnvollere) Bezeichnung der Zustände wäre $z_0 =: z_s$ (s für 'Start'), $z_2 =: z_{10}$ (bisher 10 gelesen) und $z_3 =: z_{101}$ (101 gelesen) gewesen. Wir haben z_0, z_1, z_2, z_3 gewählt, um zu verdeutlichen, dass die Benennung der Zustände ganz beliebig sein kann. Mit der oben genannte Benennung kann aber schnell erklärt werden, was mit den Zuständen beabsichtigt ist.

Sei $M \subseteq \{0, 1\}^*$ die Menge von Wörtern, die 101 als Teilwort enthalten. Wir wollen nun noch $L(A) = M$ formal zeigen.

- $L(A) \subseteq M$. Sei $w \in L(A)$. Ein Wort w , das von $L(A)$ akzeptiert wird, muss nun aber die Zeichenkette 101 enthalten, da sonst der Zustand z_3 nicht erreicht wird. z_3 kann nämlich nur über eine 1-Kante von z_2 aus (erstmalig) erreicht werden. z_2 wiederum kann ebenfalls nur über eine 0-Kante von z_1 aus erreicht werden. Und letztendlich kann z_1 nur über eine 1-Kante (von z_0 oder z_1 aus) erreicht werden. Damit enthält w die Zeichenkette 101 und folglich gilt auch $w \in M$.
- $M \subseteq L(A)$. Sei $w \in M$. Dann gibt es eine Stelle, an der zum ersten Mal das Teilwort 101 auftritt. w lässt sich also zerlegen in $v \cdot 101 \cdot v'$, wobei v die Zeichenkette 101 nicht enthält. A ist vollständig, kann also v (und auch w) ganz lesen, kann aber nach Lesen von v nicht in z_3 sein, da sonst v akzeptiert werden würde, was im Widerspruch zum eben Gezeigten stünde, dass jedes von A akzeptierte Wort die Zeichenkette 101 enthalten muss. (Dies ist hier nicht weiter wichtig, da die Argumentation auch klappt, wenn A in z_3 ist, kann in anderen Beweisen aber wichtig sein.) A ist nach Lesen von v also in z_0, z_1 oder z_2 . Aus allen diesen Zuständen wird der Automat durch 101 in z_3 überführt. Dort kann v' gelesen werden und das Wort w wird dann akzeptiert.

Übungsaufgabe 2.4: Gegeben sei der folgende NFA A :



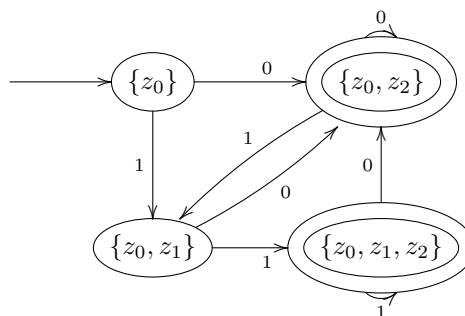
VON
4

1. Welche Sprache akzeptiert dieser NFA? Beweisen Sie Ihre Behauptung!

Lösung: Es ist $L(A) = \{0, 1\}^* \cdot \{0, 11\}$, d.h. A akzeptiert genau die Wörter aus 0en und 1en, die mit 0 oder 11 enden. Wir wollen dies nachfolgend noch beweisen. Sei $M := \{0, 1\}^* \cdot \{0, 11\}$.

- Sei $w \in L(A)$. Dann muss w mit 0 oder 11 enden, da nur dies in den Endzustand führt. Vorher muss der Automat dann in z_0 gewesen sein, wo er ein beliebiges $v \in \{0, 1\}^*$ lesen kann. Damit gilt $w = v0$ oder $w = v11$ mit $v \in \{0, 1\}^*$. In beiden Fällen gilt auch $w \in M$.
 - Sei $w \in M$. Es gibt zwei Fälle: $w = v0$ oder $w = v11$ mit $v \in \{0, 1\}^*$. Im ersten Fall kann v in z_0 gelesen werden. Die 0 überführt den Automaten dann in z_2 , wo akzeptiert wird. Analog überführt auch 11 den Automaten in z_2 . Folglich ist $w \in L(A)$.
2. Konstruieren Sie mittels der Potenzautomatenkonstruktion einen DFA B mit $L(B) = L(A)$ (d.h. einen äquivalenten DFA).

Lösung: Anwenden der Konstruktion ergibt den folgenden DFA B (wobei wir nur die initiale Zusammenhangskomponente konstruieren, d.h. die Zustände, die tatsächlich erreichbar sind).



Übungsaufgabe 2.5:

VON
4

1. Sei L eine reguläre Sprache und a ein Symbol. Zeigen Sie, dass dann auch

$$a \cdot L := \{w \mid w = av, v \in L\}$$

eine reguläre Sprache ist. (*Hinweis: Gehen Sie bspw. von einem DFA für L aus und wandeln Sie diesen so um, dass ein DFA für $a \cdot L$ entsteht.*)

2. Seien $L_1, L_2 \subseteq \Sigma^*$ reguläre Sprachen. Zeigen Sie, dass dann auch

$$L_1 \leftarrow L_2 := \{w \in \Sigma^* \mid \exists v \in L_1 \exists u \in L_2 \exists v_1, v_2 \in \Sigma^* : v = v_1 v_2 \text{ und } w = v_1 u v_2\}$$

regulär ist.

(*Hinweis: Knifflig! Probieren Sie mit Kopien der Automaten für die einzelnen Sprachen zu arbeiten.*)

Lösung:

1. Sei L eine reguläre Sprache, dann gibt es einen DFA A mit $L(A) = L$. Wir konstruieren nun einen neuen DFA A' wie folgt: A' hat zunächst die gleiche Zustandsmenge, Kantenübergänge und Endzustände wie A . Wir fügen nun einen neuen Startzustand $z_{0,A'}$ hinzu. Ferner fügen wir eine a -Kante von $z_{0,A'}$ zum (ehemaligen) Startzustand von A hinzu.

Wir zeigen nun $L(A') = a \cdot L$. Sei zunächst $w \in L(A')$, da w von A' akzeptiert wird, beginnt w mit a , da dies die einzige Kante aus dem Startzustand $z_{0,A'}$ hinaus ist. Das Restwort muss dann aufgrund der Konstruktion von A' von A akzeptiert werden. Es lässt sich w also zerlegen in $a \cdot v$ mit $v \in L(A)$, d.h. w ist in $a \cdot L$.

Ist andersherum $w \in a \cdot L$, so gilt $w = av$ mit $v \in L$. Dieses Wort kann nach Konstruktion von A' gelesen und dann akzeptiert werden. Das a überführt A' zunächst in den Startzustand von A . Von dort kann dann v gelesen werden und wir landen in einem Endzustand von A (da $v \in L(A)$) und damit auch von A' (da die Endzustände übernommen wurden). Es gilt folglich auch $w \in L(A')$.

2. Die Sprache $L_1 \leftarrow L_2$ ist dadurch charakterisiert, dass ein Wort $v \in L_1$ zunächst nur zum Teil gelesen wird. Dann wird ein Wort $u \in L_2$ gelesen und dann wird der Rest von v gelesen. Die Idee bei der nachfolgenden Konstruktion ist, dass man einen Automaten A_1 für L_1 in jedem Zustand verlassen kann und in den Startzustand eines Automaten A_2 für L_2 gelangen kann. Von jedem Endzustand von A_2 aus kann man dann in genau den Zustand gelangen aus dem man A_1 verlassen hat. Da man sich aber merken muss, welchen Zustand von A_1 man verlassen hat, sind mehrere Kopien von A_2 (und letztendlich auch A_1) nötig.

Seien A_1 und A_2 vDFA's für L_1 und L_2 , d.h. es gilt $L(A_1) = L_1$ und $L(A_2) = L_2$. Wir konstruieren einen NFA A für $L_1 \leftarrow L_2$ wie folgt: Sei $Z_1 = \{z_1, \dots, z_n\}$ die Zustandsmenge von A_1 und $Z_2 = \{z'_1, \dots, z'_n\}$ die Zustandsmenge von A_2 . z_1 und z'_1 seien die Startzustände. Sei ferner $n := |Z_1|$ die Anzahl der Zustände von A_1 . Es seien $A_{1,0}, A_{1,1}, \dots, A_{1,n}$ Kopien von A_1 und $A_{2,1}, A_{2,2}, \dots, A_{2,n}$ Kopien von A_2 (wir haben also $n+1$ Kopien von A_1 und n Kopien von A_2). Die Zustände indizieren wir entsprechend, bspw. ist $z_{1,4}$ der dem Zustand z_1 in A entsprechende Zustand in $A_{1,4}$.

In $A_{1,0}$ gibt es nun zusätzlich zu den Kanten von A_1 noch aus jedem Zustand $z_{i,0}$ heraus eine λ -Kante zu dem (ehemaligen) Startzustand $z'_{1,i}$ in $A_{2,i}$. In den $A_{2,i}$ wird sich durch i

also gemerkt aus welchem Zustand A_1 verlassen wurde. Aus jedem (ehemaligen) Endzustand von $A_{2,i}$ gibt es daher eine λ -Kante in den Zustand $z_{i,i}$ von $A_{1,i}$ (man beachte, dass $i = 0$ hier nicht möglich ist).

Dies schließt die Konstruktion beinahe ab: Startzustand von A ist der Zustand $z_{1,0}$, also der Startzustand von A_1 in der Kopie $A_{1,0}$. Endzustände von A sind all die Endzustände von A_1 in den Kopien $A_{1,1}, A_{1,2}, \dots, A_{1,n}$.

Wir müssen nun noch zeigen, dass A die gewünschte Sprache akzeptiert. Ist $w \in L_1 \leftarrow L_2$, so lässt sich w zerlegen in $w = v_1uv_2$ wie in der Mengenbeschreibung angegeben. Nach Lesen von v_1 ist A_1 in einem Zustand z_i . Der Automat A ist dann in dem entsprechenden Zustand $z_{i,0}$ und kann $A_{1,0}$ nun verlassen und in die Kopie $A_{2,i}$ von A_2 übergehen. Dort kann u gelesen werden. Da $u \in L(A_2)$ gilt, ist A_2 nach Lesen von u in einem Endzustand und A damit in der Kopie $A_{2,i}$ in einem Zustand, der durch eine λ -Kante nach $A_{1,i}$ verlassen werden kann. Dort befindet sich A nun in einer Kopie genau jenes Zustandes in dem $A_{1,0}$ verlassen wurde (d.h. im Zustand $z_{i,i}$). Hier kann nun v_2 gelesen werden und A gelangt in einen Endzustand, da ja $v_1v_2 = v \in L_1$ galt, d.h. A akzeptiert w .

Ist umgekehrt $w \in L(A)$, so lässt sich nach Konstruktion das Wort in einen Teil v_1 zerlegen, der in $A_{1,0}$ gelesen wird, einen Teil u , der in einem $A_{2,i}$ gelesen wird und einen Teil v_2 , der in $A_{1,i}$ gelesen wird. Nach Konstruktion von A gilt $u \in L(A_2)$ und $v_1v_2 = v \in L(A_1)$ (wobei hier die Kopien von A_1 wichtig sind und dass v_2 gerade in der Kopie jenes Zustandes angefangen wird zu lesen, in dem $A_{1,0}$ nach Lesen von v_1 verlassen wurde) und damit insgesamt $w = v_1uv_2 \in L_1 \leftarrow L_2$.

Informationen und Unterlagen zur Veranstaltung unter:

<http://www.informatik.uni-hamburg.de/TGI/lehre/v1/SS15/FGI1>