2. Aufgabenblatt zur Vorlesung

Grundlagen der Theoretischen Informatik, SoSe 2008

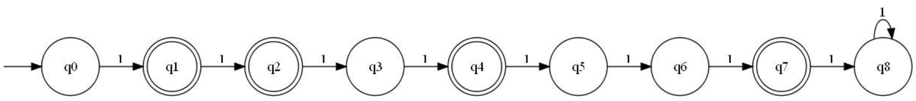
(Dr. Frank Hoffmann)

Lösung von Manuel Jain und Benjamin Bortfeldt

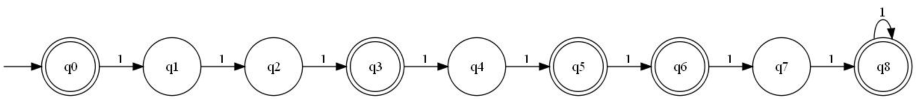
Aufgabe 1 Einelementiges Alphabet (4 Punkte)

(a) Geben Sie einen dfa an, der genau die Sprache $L = \{1^z \mid z = 3m+5n, n, m \in \mathbb{N}\}$ akzeptiert.

Lösung: Hierzu kann man sich erstmal klar machen, dass es nur endlich viele Zahlen $z \in \mathbb{N}$ gibt, welche sich nicht durch $3m + 5n$ darstellen lassen. Das ist darauf zurückzuführen, dass der $ggt(3,5) = 1$. Wenn man mal drei aufeinanderfolgende Zahlen gefunden hat, welche sich darstellen lassen, so lassen sich auch alle folgenden darstellen (für jede dieser drei Zahlen das m inkrementieren). Diese Zahlen sind: 1,2,4,7.
Idee: Konstruktion des Automaten, der die Komplementärsprache erkennt:

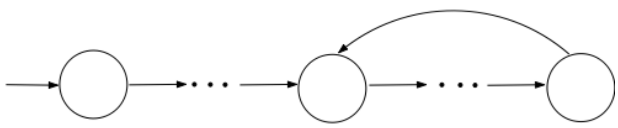


Wir wir aus Aufgabe 5 wissen, kann man nun akz. Zustände und nicht akz. Zustände austauschen um den Automaten zur Komplementärsprache zu bilden:



(b) Wie sehen reguläre Sprachen über einem einelementigen Alphabet aus? (Betrachten Sie dazu die entsprechenden dfa's) Geben Sie eine Charakterisierung an!

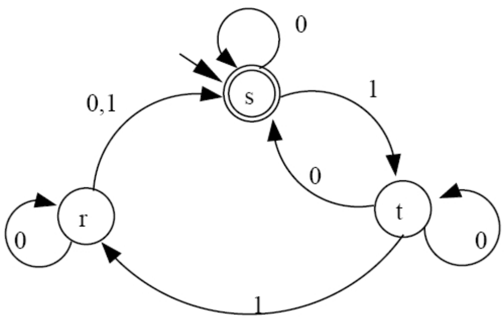
Lösung: Allgemein sieht ein Automat für eine Sprache über einem unären Alphabet so aus:



Man kann diesen Automaten in zwei Teile unterteilen. Einmal den Teil vom Startzustand bis zu dem in der Mitte. Dieser Teil wird als nichtperiodischer Teil bezeichnet. Und zum Anderen den Schlußteil, welchen man als periodischen Teil bezeichnet.
Wichtig ist, dass der nichtperiodische Teil nicht zwingend existieren muss. Der periodische Teil kann auch nur einen Zustand enthalten. Man kann also mit einem solchen Automaten nur Akzeptanzkriterien festlegen, welche sich auf die Anzahl der Buchstaben beziehen. Daraus kann man leicht folgern, dass eine Sprache über einem unären Alphabet lediglich über die Anzahl ihrer Buchstaben charakterisierbar ist.

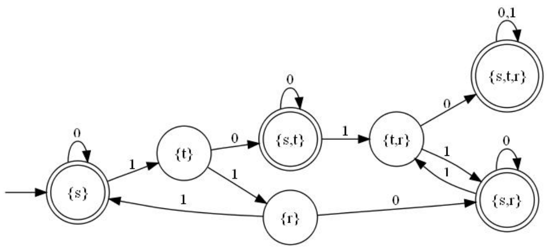
Aufgabe 2 Zustandsdiagramm nfa (6 Punkte)

Betrachten Sie das folgende Zustandsüberführungsdiagramm eines nfa.



(a) Konstruieren Sie einen äquivalenten deterministischen Automaten!

Lösung :



Der Automat hat nur 7 Zustände, obwohl die Mächtigkeit der Potenzmenge 8 ist. Das ist darauf zurückzuführen, dass der Zustand, der die leere Menge darstellt niemals erreicht wird, folglich braucht man ihn nicht im Diagramm integrieren (Stichwort: Automatenminimierung).

(b) Welche Sprache wird von diesem Automaten erkannt? (Regulärer Ausdruck)

Lösung: Um den Klenee-Algorithmus leichter anwenden zu können, werden die Zustände folgendermaßen durchnummeriert:

$\{s\}$	1
$\{t\}$	2
$\{r\}$	3
$\{s,t\}$	4
$\{t,r\}$	5
$\{s,r\}$	6
$\{s,t,r\}$	7

Der Kleene-Algorithmus sieht wie folgt aus:

$$L_{i,j}^{(0)} = \begin{cases} \{a \mid a \in \Sigma,\ \delta(q_i, a) = q_j\} & \text{, falls } i \neq j \\ \{a \mid a \in \Sigma,\ \delta(q_i, a) = q_j\} \cup \{\epsilon\} & \text{, falls } i = j \end{cases}$$

$$L_{i,j}^{(k)} = L_{i,j}^{(k)} \cup L_{i,k}^{(k-1)} \cdot \left(L_{k,k}^{(k-1)}\right)^* \cdot L_{k,j}^{(k-1)}$$

k \ i,j	1,1	1,2	1,3	1,4	1,5	$\cdots$
0	$0 \cup \epsilon$	1	$\emptyset$	$\emptyset$	$\emptyset$	
1	0^*	$1 \cup (0^*1)$	$\emptyset$	$\emptyset$	$\emptyset$	
$\vdots$						

So verfährt man nun weiter, bis man alle Zellen der Tabelle ausgefüllt hat. Leider sind dies 7^3-viele. Nachhher werden uns aber nur die Zellen interessieren, die vom Startzustand zu den akzeptierenden Zuständen führen, wobei alle Zustände als Zwischenzustände erlaubt sind (also maximales k). Der reguläre Ausdruck wird daher dann folgendermaßen gebildet:

$$\bigcup_{j \in F} L_{1,j}^{(n)} \quad ,n \text{ ist das maximale k. In unserem Fall 7.}$$

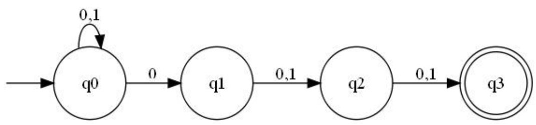
Er sieht nach Vereinfachung folgendermaßen aus:

$$(0 \cup 111)^*(\epsilon \cup 100^* \cup 100^*10(0 \cup 1)^* \cup (110 \cup 100^*11)(0 \cup 11)^*(\epsilon \cup 10(0 \cup 1)^*))$$

Aufgabe 3 nfa versus dfa (2 Punkte)

Geben Sie einen *nichtdeterministischen* endlichen Automaten an, der alle Wörter aus $\{0,1\}^*$ akzeptiert,die mindestens Länge 3 haben und deren drittletzter Buchstabe eine 0 ist. Der Automat soll *weniger* als 8 Zustände haben. (Vgl. Zettel 1, Aufgabe 2b)

Lösung :



Der Automat entscheidet bei gelesener 0 nichtdeterministisch, ob diese nun die drittletzte ist oder nicht. Beim Prüfen, ob es die drittletzte ist, geht er den Weg nach q_3. Da für q_3 allerdings keine weiteren Übergänge definiert sind, geht der Automat in einen nicht definierten Zustand, falls es doch nicht die drittletzte 0 war. Wörter der Länge 1 und 2 soll der Automat ja nicht akzeptieren.

Aufgabe 4 nfa ohne ϵ-Übergänge (2 Punkte)

Zeigen Sie: Für einen nfa mit n Zuständen und ϵ–Übergängen gibt es einen äquivalenten nfa mit nicht mehr Zuständen, der keine ϵ–Übergänge hat.

Lösung: Zu einem gegebenen nfa $M = (Q, \Sigma, \delta, q_o, F)$ mit ϵ-Übergängen, erhalten wir einen nfa M' ohne ϵ-Übergänge wie folgt:
Jeder Zustand q wird durch einen neuen Zustand $E(q)$ ersetzt, wobei $E(q)$ der ϵ-Abschluss von q ist und folgendermassen definiert wird:
$E(q) = \{q' \in Q | q' \text{ kann von } q \text{ mit } \geq 0\ \epsilon\text{-Übergängen erreicht werden}\}$.
Jeder Zustand $E(q)$ bei dem gilt $q' \in E(q) \wedge q' \in F$ wird als aktzeptierend gekennzeichnet. Unser neuer Startzustand ist $E(q_0)$.

Nun müssen die neuen Zustände im Zustandsdiagramm noch verknüpft werden. Wir verbinden die Zustände $E(q)$ und $E(p)$ mit einer Kante von $E(q)$ nach $E(p)$, die mit $a \in \Sigma$ beschriftet wird, genau dann wenn es ein $q' \in E(q)$ und ein $p' \in E(p)$ gibt, die in unserem ursprünglichen nfa mit einer Kante von q' nach p', die auch mit a beschriftet ist, verbunden war.

Aufgabe 5 Abschlusseigenschaften (6 Punkte)

(a) Zeigen Sie: L_1, L_2 regulär, dann sind auch $L_1^c = \Sigma^* \setminus L_1$, $L_1 \setminus L_2$ und $L_1 \cap L_2$ regulär.
zu zeigen: L_1^c regulär, wenn L regulär.
Wenn L_1 regulär, existiert ein dfa $M = (Q, \Sigma, \delta, q_0, F)$, sodass $L_1 = L(M)$. Für alle Wörter $w \in L_1$ gilt weiterhin, dass $\delta(q_0, w) \in F$. Für L_1^c muss aber gelten, dass kein Wort aus L_1 akzeptiert wird. Außerdem müssen alle Wörter $w \notin L_1$ akzeptiert werden. Wir bestimmen nun einen neuen Automaten $M' = (Q, \Sigma, \delta, q_0, F')$ wobei alle Elemente wie aus M sind mit Ausnahme von F'. Wenn wir nun $F' = Q \setminus F$ setzen, werden durch M' alle Wörter akzeptiert, die M verwerfen würde. Weiterhin werden bei M' alle Wörter verworfen, die M akzeptieren würde. Und genau das macht die Komplementärsprache aus.

zu zeigen: $L_1 \cap L_2$ regulär, wenn L_1, L_2 regulär.
Hier kann man die Morganschen Regeln verwenden, dann ergibt sich:

$$L_1 \cap L_2 = (L_1^c \cup L_2^c)^c$$

Jetzt wurde aber gerade gezeigt, dass das Komplement einer reg. Sprache wieder eine reg. Sprache ist. Nach der Vereinigung handelt es sich immer noch um eine reg. Sprache (wurde in der VL gezeigt). Dann nochmal das Komplement gebildet bleibt regulär.

zu zeigen: $L_1 \setminus L_2$ regulär, wenn L_1, L_2 regulär.
$L_1 \setminus L_2$ bedeutet ja: „alle Wörter aus Σ^*, für die gilt, dass sie in L_1 vorkommen und NICHT in L_2 vorkommen“. An diese neue Sprache werden also zwei Forderungen gestellt, die unbedingt erfüllt sein müssen. Als Mengen kann man das also so ausdrücken:

$$L_1 \setminus L_2 = \{w \in \Sigma^* \mid w \in L_1 \wedge w \notin L_2\} = L_1 \cap L_2^c$$

Das sind aber wieder Operationen, die angewendet auf reg. Sprachen wieder reg. Sprachen ergeben. Also handelt es sich um eine reg. Sprache.

(b) Zeigen Sie, daß die Klasse der dfa–Sprachen abgeschlossen gegenüber der folgenden Operation Φ ist:

$$\Phi(L) = \{x \mid x \in L \text{ und es gibt einen echten Präfix von } x \text{, der auch in } L \text{ liegt}\}$$

Lösung: Dies lässt sich elegant über reguläre Ausdrücke zeigen (zu jeder regulären Sprache gibt es schließlich einen solchen). Außerdem wurde in der Vorlesung bereits gezeigt, dass dfa-Sprachen gleich reguläre Sprachen sind.

Sei R nun ein regulärer Ausdruck, der die Sprache L erzeugt, dann ist die Sprache L' ereugt durch $R \circ \Sigma \circ \Sigma^*$ die Sprache, welche alle Wörter enthält, deren Präfix aus L kommt. Da man einen regulären Ausdruck dazu angeben kann, scheint L' auch regulär zu sein. In Aufgabenteil a wurde gezeigt, dass die Schnittmenge zweier regulärer Sprachen abgeschlossen ist. Daher konstruieren wir nun folgende Sprache:

$$L' \cap L$$

Da sind dann genau die Wörter drin für die gilt, dass sie einen Präfix haben, der in L liegt (so wurde L' ja konstruiert) und dass sie selber in L liegen (Durch den Schnitt).

Aufgabe 6 Verständnisfragen (ohne Punkte aber trotzdem verpflichtend)

Diese Aufgabe sollten Sie ohne Hilfsmittel lösen können. Die Tutoren können es zwar nicht überprüfen, aber Sie sich selbst schon. :-)

Richtig oder falsch? Geben Sie kurze Begründungen:

(a) Jede Teilsprache einer regulären Sprache ist regulär.
FALSCH. Da Σ^* eine reguläre Sprache ist, würde das bedeuten, dass jede Sprache regulär ist, da für alle Sprachen L gilt: $L \subseteq \Sigma^*$

(b) Sei k eine feste natürliche Zahl und L eine beliebige Sprache über $\Sigma = \{0, 1\}$. Dann ist $\text{Suff}_k(L) = \{v \mid v \text{ ist Suffix eines Wortes } w \in L, |v| \leq k\}$ eine reguläre Sprache.
RICHTIG. Es werden nur Wörter in dieser Sprache liegen, die eine feste vorgegebene Länge nicht überschreiten. Das Alphabet ist endlich und hat die Mächtigkeit 2. Also können in dieser Sprache sowieso nur maximal 2^k-viele Wörter liegen. Also hat die Sprache eine endliche Mächtigkeit und sowas ist immer regulär.

(c) Sei C eine beliebige Menge regulärer Sprachen. Dann ist $\bigcup_{L \in C} L$ auch regulär.
FALSCH. Da C n.V. beliebig ist, muss es sich nicht um eine endliche Menge handeln. Die Vereinigung ist aber nur bei endlich ofter Anwendung bzgl. Regularität abgeschlossen.