FGI-2 – Formale Grundlagen der Informatik II

Modellierung und Analyse von Informatiksystemen

Präsenzlösung 1 : Endliche Automaten

Präsenzteil am 17./18. 10. – Abgabe am 24. 10. 2016

Präsenzaufgabe 1.1:

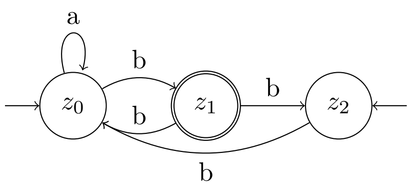
1. Wir wissen aus FGI-1, dass es zu jedem NFA A einen DFA B mit $L(A) = L(B)$ gibt. Kann man B aus A berechnen? Wenn ja, wie?

   Lösung: Potenzautomatenkonstruktion (siehe FGI-1).

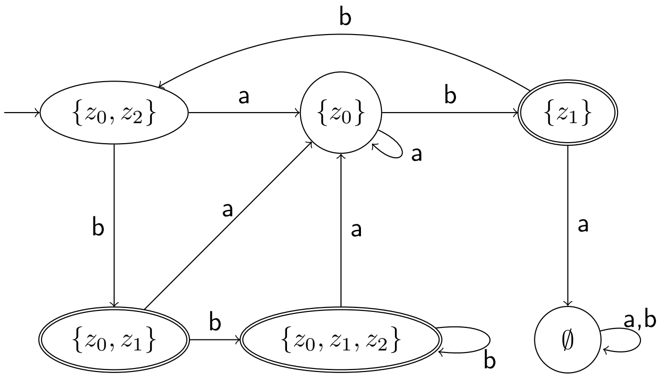
2. Sei A ein NFA mit $L(A) \subseteq \Sigma^*$. Geben Sie eine Konstruktionsvorschrift für einen NFA $\bar{A}$ an, für den $L(\bar{A}) = \Sigma^* \setminus L(A)$ gilt. (Tipp: Wandeln Sie zunächst A in einen DFA um.)

   Lösung: O.b.d.A. sei A ein vollständiger DFA. Wir konstruieren $\bar{A}$ aus A, indem wir in $\bar{A}$ genau die Zustände als Endzustände wählen, die dies in A nicht sind. Die restlichen Komponenten bleiben gleich. Auch $\bar{A}$ ist ein vDFA. Wurde in A ein Wort akzeptiert, dann in $\bar{A}$ nicht und umgekehrt.

3. Konstruieren Sie den Potenzautomaten für folgenden NFA:



   Lösung: Der Potenzautomat ergibt sich wie folgt:



Präsenzaufgabe 1.2:

1. Sei ein NFA $A := (Q, \Sigma, \delta, Q^0, F)$ gegeben. Überprüfen Sie die folgenden Konstrukte auf *syntaktische* Korrektheit (dabei handelt es sich *nicht* um reguläre Ausdrücke):

   - $L(A) = \emptyset$
   - $L(A) = \{\emptyset\}$
   - $L(A) = \lambda$
   - $L(A) = \{\lambda\}$

   Lösung: Nur $L(A) = \lambda$ ist nicht korrekt, da $L(A)$ stets eine Menge ist. Der Fall $L(A) = \{\emptyset\}$ erfordert jedoch $\emptyset \in \Sigma$, was bspw. auftritt, wenn es sich bei Σ um eine Potenzmenge handelt.

2. Erörtern Sie die Unterschiede zwischen den syntaktisch korrekten Ausdrücken. Können Sie Rückschlüsse auf die Bestandteile des NFA $(Q, \Sigma, \delta, Q^0, F)$ in den einzelnen Fällen ziehen?

   Lösung:

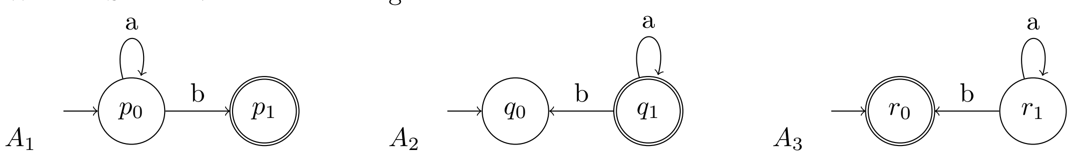
   - $L(A) = \emptyset$: Besagt, dass es kein Wort in der akzeptierten Sprache gibt. Es ist kein Rückschluss auf Σ und Q möglich. Kein Element aus F darf durch δ^* aus Q^0 erreichbar sein.
   - $L(A) = \{\emptyset\}$ akzeptiert nur ein einziges Wort, welches aus einem einzelnen $\emptyset$ Symbol besteht. Es muss $\emptyset \in \Sigma$ gelten und es dürfen Elemente aus F nur durch je einen einzigen $\emptyset$-Übergang (in δ) aus Elementen aus Q^0 erreichbar sein. Es muss $|Q| \geq 2$ gelten.
   - $L(A) = \{\lambda\}$ beschreibt eine Sprache, die nur das leere Wort enthält. Es ist kein Rückschluss auf Σ möglich. Es muss $F \cap Q^0 \neq \emptyset$ gelten und es darf kein $x \in \Sigma$, $q \in Q^0$, $q' \in Q$ und $q'' \in F$ geben mit $(q, x, q') \in \delta$ und $q'' \in \delta^*(q')$.

3. Beschreiben Sie ein Verfahren, welches für einen gegebenen nicht-deterministischen Automaten $A := (Q, \Sigma, \delta, Q^0, F)$ feststellt, ob $L(A) = \emptyset$ gilt.

   Lösung: Zuerst ist die initiale Zusammenhangskomponente von A zu bilden. Hierzu wird die Menge der erreichbaren Zustände Q' zunächst mit den Startzuständen von A gefüllt ($Q' := Q^0$) und anschließend so lange um Nachfolgezustände von allen bereits in Q' enthaltenen Zuständen ergänzt, bis keine Änderungen mehr auftreten.

   Enthält Q' keinen Endzustand (es gilt $Q' \cap F = \emptyset$), so akzeptiert A kein einziges Wort, also gilt $L(A) = \emptyset$.

4. Wenden Sie Ihr Verfahren auf folgende Automaten an:



   Lösung: $p_1 \in Q'_1 = \{p_0, p_1\}$ $\quad$ $q_1 \notin Q'_2 = \{q_0\}$ $\quad$ $r_0 \in Q'_3 = \{r_0\}$

   $L(A_1) = \{a\}^*\{b\} \neq \emptyset$ $\quad$ $L(A_2) = \emptyset$ $\quad$ $L(A_3) = \{\lambda\} \neq \emptyset$

5. Begründen Sie Korrektheit und Termination Ihres Verfahrens.

Lösung: *Korrektheit:* Die Argumentation wird leichter, wenn beide Seiten negiert werden:

$$\begin{aligned} & L(A) = \emptyset \quad \Leftrightarrow \quad Q' \cap F = \emptyset \\ \equiv \quad & L(A) \neq \emptyset \quad \Leftrightarrow \quad Q' \cap F \neq \emptyset \end{aligned}$$

Es sind nun **zwei** Richtungen zu zeigen:

(a) Falls $L(A) \neq \emptyset$, dann $Q' \cap F \neq \emptyset$.
Sei $w = a_1 a_2 \ldots a_n \in L(A)$. Dann gibt es eine Rechnung von einem Startzustand $q_0 \in Q^0$ zu einem Endzustand $q_e \in F$: $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_e$. Da q_{i+1} jeweils ein Nachfolgezustand von q_i ist und q_0 anfänglich in Q' aufgenommen wurde, ist am Ende auch $q_e \in Q'$. Also ist die Schnittmenge $Q' \cap F$ nicht leer.

(b) Falls $Q' \cap F \neq \emptyset$, dann $L(A) \neq \emptyset$.
Sei $q_e \in Q' \cap F$. Dann ist q_e entweder ein Startzustand (direkt in Q' enthalten) oder ein Nachfolgezustand eines zuvor schon in Q' aufgenommenen Zustandes q'. Für q' gilt Gleiches: Entweder ist $q' \in Q^0$ oder q' ist Nachfolgezustand von q''. Diese Argumentation lässt sich fortsetzen, bis man auf einen Startzustand q^m trifft. Die Übergänge $q^m \xrightarrow{a_1} \ldots \xrightarrow{a_{m-1}} q' \xrightarrow{a_m} q_e$ bilden eine Erfolgsrechnung im NFA A, so dass $w = a_1 a_2 \ldots a_n \in L(A)$ gilt. Also ist $L(A)$ nicht leer.

Termination: Die Konstruktion der Menge Q' bricht garantiert nach endlich vielen Schritten ab, da die Menge Q, aus welcher die Elemente für Q' stammen, endlich ist. Ob Q' einen Endzustand enthält, lässt sich durch reihenweises Überprüfen der in Q' enthaltenen Elemente in endlicher Zeit feststellen. Somit terminiert das Verfahren immer in endlicher Zeit.

6. Ist Ihr Verfahren ohne Modifikationen für deterministische und verallgemeinerte endliche Automaten anwendbar? Wenn nicht, was müsste modifiziert werden?

Lösung: Das Verfahren für NFA ist genauso für DFA anwendbar, da ein DFA lediglich ein Spezialfall eines NFA ist.

Die Argumentation zu Teilaufgabe 5 ist nicht davon abhängig, ob an den Kanten genau ein Symbol steht. Beim verallgemeinerten FA können die Teile $a_1 \ldots a_n$ als Teilwörter beliebiger Länge betrachtet werden.

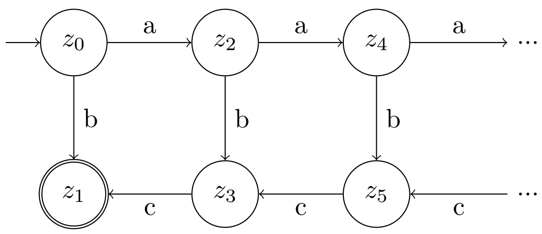
Übungsaufgabe 1.3: Machen Sie sich mit dem Tool Renew vertraut. $\boxed{10\%}$
Installieren Sie sofern noch nicht geschehen Java auf Ihrem System. Laden Sie sich das FGI2-Paket von Renew passend zu Ihrem Betriebssystem herunter (Renew selbst ist in Java geschrieben und plattformunabhängig, einige Plugins enthalten jedoch plattformabhängigen Code):
`http://www.informatik.uni-hamburg.de/TGI/lehre/vl/WS1617/FGI2/sec/ressourcen.html`
Entpacken Sie das Archiv und navigieren Sie in das Verzeichnis renew2.5/bin/{win, unix}/ und führen Sie die installrenew.bat (unter Windows) bzw. installrenew.sh (unter Linux/Mac OS X) aus. Das Skript erstellt einige weitere Skriptdateien in diesem Ordner. Nachdem das Skript durchgelaufen ist, können Sie Renew mittels „loadrenew.bat“ bzw. „loadrenew.sh“ starten.
Weitere Informationen über Renew sowie Dokumentation und Plugins finden Sie auf
`http://renew.de`.

Auf diesem Blatt benötigen Sie das Plugin für endliche Automaten, dieses ist im FGI2-Paket von Renew bereits vorinstalliert. Um die Palette des Plugins anzuzeigen, klicken Sie in im Menü „Tools“ auf „FA Drawing Tool/FA Drawing Tool“. Erzeugen Sie nun eine neues Automatendiagramm, indem Sie im Menü „File“ auf „New Drawing...“ klicken und „Finite Automata Drawing (.fa)“ auswählen. Mit der untersten Zeile im Renew Hauptmenü können Sie nun ihren Automaten zeichnen.

***Achtung:** Es ist verpflichtend notwendig zu jeder Aufgabe, in der ein Automatendiagramm gefordert ist, dieses als* Renew *2.5 .fa Datei abzugeben! Andere Abgabeformen (Scans, Bilder, pdfs, Texte, ...) werden nicht gewertet!*

Übungsaufgabe 1.4: 45%

1. Wie Sie wissen, steht die Abkürzung NFA für „Non-deterministic finite automaton“ oder auf deutsch „Nichtdeterministischer endlicher Automat“. Erläutern Sie kurz, was „endlich“ in diesem Zusammenhang bedeutet (Ein Satz genügt).
2. Betrachten Sie den folgenden (nicht endlichen) Automaten $A := (Q, \Sigma, \delta, Q^0, F)$ mit $\Sigma = \{a, b, c\}$:



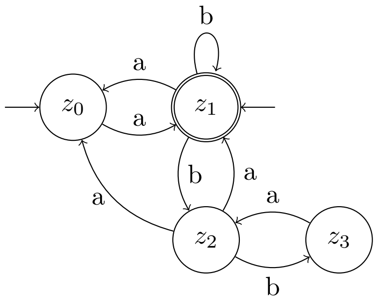
   Bestimmen Sie die Sprache $L(A)$, die dieser Automat akzeptiert, in Mengenschreibweise.
3. Ist $L(A)$ regulär? Begründen Sie Ihre Antwort!
4. Geben Sie eine Grammatik G mit $L(G) = L(A)$ an. Von welchem Typ muss Ihre Grammatik mindestens sein um $L(A)$ abbilden zu können?
5. Betrachten Sie nun die Menge $A_n = L(A) \cap \{w \in \Sigma^* \mid |w| = n\}$ (für $n \in \mathbb{N}$). Sofern möglich geben Sie folgende endliche Automaten an:
   - Endlicher Automat B mit $L(B) = A_5$.
   - Endlicher Automat C mit $L(C) = A_4$.

   Sollte dies nicht möglich sein, so begründen Sie warum es nicht möglich ist.

Übungsaufgabe 1.5: Zeigen Sie die zweite Teilaussage von Lemma 1.9: „Das Universalitätsproblem für NFA ist entscheidbar.“ 45%

1. Beschreiben Sie ein Verfahren, welches für einen gegebenen nichtdeterministischen Automaten $A := (Q, \Sigma, \delta, Q^0, F)$ feststellt, ob $L(A) = \Sigma^*$ gilt. Sie dürfen bereits bekannte Verfahren als Teilschritte verwenden, indem Sie Ein- und Ausgabedaten der Verfahren eindeutig benennen.

2. Wenden Sie Ihr Verfahren auf folgenden Automaten an. Dokumentieren Sie dabei die Zwischenergebnisse.



A_1

3. Begründen Sie Korrektheit und Termination Ihres Verfahrens.
4. Ist Ihr Verfahren ohne Modifikationen für deterministische und verallgemeinerte endliche Automaten anwendbar? Wenn nicht, was müsste modifiziert werden?

Mehr Details zur Veranstaltung unter:
http://www.informatik.uni-hamburg.de/TGI/lehre/vl/WS1617/FGI2/