ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

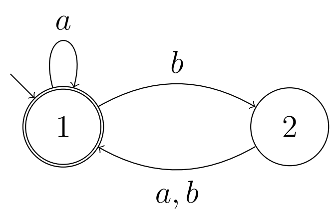
Prof. Dr. Andreas Podelski
Matthias Heizmann
Alexander Nutz
Christian Schilling

11.11.2014
Abgabe bis spätestens Montag 17.11.2014, 16 Uhr
in den Briefkästen in Gebäude 51

4. Übungsblatt zur Vorlesung
Theoretische Informatik

Aufgabe 1: Endlicher Automat $\rightsquigarrow$ regulärer Ausdruck 4+3 Punkte

(a) Bestimmen Sie mit dem im Skript vorgestellten Verfahren (Satz von Kleene „$\Leftarrow$“) einen regulären Ausdruck, welcher die Sprache des folgenden DEAs beschreibt.



Für reguläre Ausdrücke α, β, γ gelten für die Operationen “Konkatenation” und “Oder” die folgenden Regeln:

$$\begin{aligned}
\text{Assoziativität:}\quad & \alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma,\ \ \alpha(\beta\gamma) = (\alpha\beta)\gamma \\
\text{Kommutativität:}\quad & \alpha + \beta = \beta + \alpha \\
\text{Neutrale Elemente:}\quad & \emptyset + \alpha = \alpha,\ \ \varepsilon\alpha = \alpha,\ \ \alpha\varepsilon = \alpha \\
\text{Distributivität:}\quad & \alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma,\ \ (\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma \\
\text{Absorption:}\quad & \emptyset\alpha = \emptyset,\ \ \alpha\emptyset = \emptyset
\end{aligned}$$

Außerdem gelten für den Sternoperator die folgenden Regeln:

$$\varepsilon^* = \varepsilon,\ \ (\varepsilon + \alpha)^* = \alpha^*,\ \ (\varepsilon + \alpha)\alpha^* = \alpha^*,\ \ \alpha^*(\varepsilon + \alpha) = \alpha^*$$

Sie dürfen die regulären Ausdrücke, welche die Sprachen $L^k_{i,j}$ beschreiben, mithilfe dieser Regeln vereinfachen.

(b) Ein DEA $\mathcal{A} = (\Sigma, Q, \rightarrow, q_0, F)$ kann auch mithilfe eines auf Gleichungssystemen basierenden Verfahrens in einen regulären Ausdruck überführt werden.

Dazu wird für jeden Zustand q_i des Automaten eine Variable x_i vorgesehen. Die Variable x_i steht für einen regulären Ausdruck, der die vom Zustand q_i ausgehend akzeptierte Sprache beschreibt. Formal:

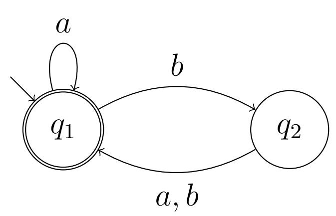
$$L(x_i) := \{w \in \Sigma^* \mid q_i \xrightarrow{w} q_F \text{ und } q_F \in F\}$$

Die vom Automaten akzeptierte Sprache wird dann durch den regulären Ausdruck x_0 (die zum Startzustand gehörende Variable) beschrieben. Der Ausdruck x_i zu der von einem Zustand q_i aus akzeptierten Sprache kann über die Ausdrücke seiner Nachfolge-Zustände definiert werden.
Für jeden Zustand wird dazu eine Gleichung aufgestellt. Wenn z.B. von Zustand q_1 genau die Zustände q_2 durch a und q_3 durch b erreichbar sind, erhält man die Gleichung

$$x_1 = a \cdot x_2 + b \cdot x_3.$$

Für einen Endzustand wird außerdem ε hinzugefügt (in diesem Fall $x_1 = a \cdot x_2 + b \cdot x_3 + \varepsilon$).

Betrachten Sie nun folgenden DEA (vgl. auch Aufgabe 1):



Stellen Sie das entsprechende Gleichungssystem auf und geben Sie eine Lösung durch Einsetzen an. In diesem Fall ist nach x_1 aufzulösen, da diese Variable dem Startzustand zugeordnet ist.

Sie dürfen die regulären Ausdrücke mithilfe der Regeln aus Aufgabe 1 vereinfachen. Zusätzlich gilt die *Regel von Arden*: Die Gleichung $x = (\alpha \cdot x) + \beta$ hat $x = \alpha^* \cdot \beta$ als Lösung und diese Lösung ist eindeutig, wenn ε nicht in $L(\alpha)$ enthalten ist.

Aufgabe 2: Regulärer Ausdruck $\rightsquigarrow$ endlicher Automat 4+3 Punkte

Betrachten Sie den folgenden regulären Ausdruck.

$$\alpha \equiv (a + b(a + b))^*$$

Konstruieren Sie zu diesem Ausdruck folgendermaßen jeweils einen äquivalenten endlichen Automaten.

(a) Verwenden Sie das im Skript vorgestellte Verfahren (Satz von Kleene „$\Rightarrow$“). Geben Sie die Automaten für jeden Teilschritt explizit an. Es genügt, Duplikate einmalig anzugeben.

(b) Verwenden Sie das nachfolgende Verfahren basierend auf Residuen. Geben Sie alle Zwischenschritte an. Sie dürfen die regulären Ausdrücke wieder vereinfachen.

Dazu definieren wir zunächst:

Definition (Residuum für reguläre Ausdrücke)**.** *Seien e, e_1, e_2 reguläre Ausdrücke über dem Alphabet Σ. Induktiv über den Aufbau von regulären Ausdrücken definieren wir das Residuum $a^{-1}e$ für $a \in \Sigma$:*

- $a^{-1}\emptyset = \emptyset$
- $a^{-1}\varepsilon = \emptyset$
- $a^{-1}a = \varepsilon$
- $a^{-1}b = \emptyset$
- $a^{-1}(e_1 + e_2) = (a^{-1}e_1) + (a^{-1}e_2)$
- $a^{-1}(e_1 \cdot e_2) = (a^{-1}e_1) \cdot e_2,$ *falls* $\varepsilon \notin L(e_1)$
  $a^{-1}(e_1 \cdot e_2) = (a^{-1}e_1) \cdot e_2 + (a^{-1}e_2),$ *falls* $\varepsilon \in L(e_1)$
- $a^{-1}e^* = a^{-1}(\varepsilon + e \cdot e^*)$

Nun definieren wir das Verfahren, um einen regulären Ausdruck e_0 in einen DEA umzuwandeln. Dabei ordnen wir jeweils der Sprache eines regulären Ausdrucks e_i den Zustand q_i zu.

1. Der Sprache $L(e_0)$ wird ein neuer Startzustand q_0 zugeordnet. Setze $j := 1$.
2. Wähle einen Zustand q_i und ein Symbol $a \in \Sigma$, für das noch keine ausgehende Kante existiert. Berechne $e_j := a^{-1}e_i$. Sollte noch kein Zustand q für $L(e_j)$ existieren, so erstelle einen neuen Zustand q_j und erhöhe j (d.h. $j := j + 1$). Sei q nun der Zustand für $L(e_j)$. Füge eine Kante $q_i \xrightarrow{a} q$ hinzu.
3. Wiederhole Schritt 2, bis in jedem Zustand alle ausgehenden Kanten existieren.
4. Ein Zustand q_i wird zum Endzustand genau dann, wenn $\varepsilon \in L(e_i)$.