Prof. Javier Esparza
Philipp Czerner, Martin Helfrich

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2021 – Hausaufgabenblatt 3

- Beachten Sie die Abgabemodalitäten auf der Vorlesungswebsite!
- Sei $\Phi := \{\{1\}, \{2\}, \{3, 4\}, \{5\}\}$. Nach dem Abgabedatum werden wir für jede Menge $A \in \Phi$ eine zufällige Aufgabe $a \in A$ wählen und korrigieren.
- Es werden diese Aufgaben korrigiert: H3.1, H3.2, H3.3, H3.5
- Wenn Sie einen Beweis aufstellen, von dem Sie wissen, das einzelne Schritte problematisch oder unvollständig sind, merken Sie dies bitte in Ihrer Lösung an, damit wir das bei der Korrektur positiv berücksichtigen können.

Aufgabe H3.1. (*ein paar as, dann ein b...*) 1+1+1 Punkte

Geben sie an, zu welchen regulären Ausdrücken die gleiche Sprache gehört. Bestimmen Sie außerdem für jede Sprache, die Sie so identifizieren, ein Wort, das nur in dieser Sprache liegen. Die sechs regulären Ausdrücke erzeugen drei unterschiedliche Sprachen.

Beispiel: Sie stellen fest, dass (1,2), (3,4) und (5,6) die gleiche Sprache erzeugen. Dann müssen Sie ein Wort in $L_{1,2} \setminus (L_{3,4} \cup L_{5,6})$ finden, eines in $L_{3,4} \setminus (L_{1,2} \cup L_{5,6})$, und eines in $L_{5,6} \setminus (L_{1,2} \cup L_{3,4})$.

Wie üblich, schreiben wir r^+ anstelle von rr^*, für einen regulären Ausdruck r.

(1) $b \mid a^+(baa^+)^*(b|bab)$

(2) $(a \mid ab)^+ \mid a(b^*a\emptyset)^+$

(3) $(a^+b)^*$

(4) $(aba \mid a)^*b$

(5) $a(a^*ba)^*a^*(b \mid \epsilon)$

(6) $\emptyset^* \mid a(a \mid ba)^*b$

Lösungsskizze.

- $(1) \equiv (4)$: b
- $(2) \equiv (5)$: a (alle Wörter in a^+ sind möglich)
- $(3) \equiv (6)$: ε

Aufgabe H3.2. (*Matriririririkelnummern*) 2+2+1 Punkte

An der Technischen Hochschule Estlingen-Oberfeld werden Matrikelnummern nach einem kuriosen System vergeben. Matrikelnummern haben die Form `#36442` – eine Raute gefolgt von einer Zahl. Die Quersumme der Zahl muss hierbei eine Primzahl sein, eine alte estlinger Tradition. Nun hält der technische Wandel auch nicht vor Estlingen-Oberfeld ein, und der Rektor versucht verzweifelt, einen endlichen Automaten zu finden, der korrekte Matrikelnummern akzeptiert.

(a) Sei $\Sigma := \{0, ..., 9\}$, und $L := \{\#\}\{w \in \Sigma^* : \sum_{i \geq 1} w_i \text{ prim}\}$, also die Sprache der Dezimalzahlen, deren Quersumme eine Primzahl ist, mit einer Raute vorneweg. Beweisen Sie, dass L nicht regulär ist.

Der Rektor hat sich in der Zwischenzeit einen neuen Ansatz überlegt. Die Primzahlquersumme bräuchte man eigentlich nur in der traditionellen estlinger Form, für neue Studenten müsse man es nicht so genau nehmen – `##1234` sei ja auch eine prima Matrikelnummer (für nicht-Estlinger).

(b) Zeigen Sie, dass die Sprache $L' := L \cup \{\#^i w : i \neq 1, w \in \Sigma^*\}$ die Eigenschaft des Pumping-Lemmas erfüllt.

(c) Beweisen Sie, dass L' dennoch nicht regulär ist, z.B. indem Sie Abschlusseigenschaften regulärer Sprachen verwenden.

Lösungsskizze.

(a) Wir nehmen an, L wäre regulär. Nun verwenden wir das Pumping-Lemma, um einen Widerspruch herzuleiten. Sei also $n \in \mathbb{N}$ beliebig (n ist die „Pumping-Lemma Zahl"). Es gibt unendlich viele Primzahlen, wir können dementsprechend eine Primzahl $p \geq n$ wählen. Als Wort z wählen wir nun $\#1^p$. Offensichtlich gilt $z \in L$ und $|z| \geq n$. Das Pumping-Lemma gibt uns nun eine Zerlegung $z = uvw$, sodass $v \neq \varepsilon$ und $uv^iw \in L$ für alle $i \in \mathbb{N}$. Falls v die Raute enthält, ist das Wort $uv^0w = w$ bereits nichts in L enthalten. Ansonsten sei $z' := uv^{p+1}w$, aber für dessen Quersumme gilt

$$\sum_{i>1} z'_i = \sum_{i>1} z_i + p \sum_{i \geq 1} v_i = p + p|v| = p(|v| + 1)$$

Da $v \neq \varepsilon$ und somit $|v| > 0$ ist $p(|v|+1)$ aber keine Primzahl, da es durch die Zahlen p und $|v| + 1$ teilbar ist. In jedem Fall konnten wir das Wort z also „aufpumpen", sodass es nicht mehr in L liegt. Dies steht aber im Widerspruch zur Aussage des Pumping-Lemma, somit kann L nicht regulär sein.

(b) Wir wählen $n := 2$. Sei nun $z \in L'$ beliebig, mit $|z| \geq 2$. Nun gibt es folgende Fälle, abhängig von der Anzahl an Rauten k, mit denen z beginnt.

- $k = 0$: Dann ist $z \notin L$ und $z \in \Sigma^*$. Wir wählen $u := \varepsilon$, $v := z_1$, $w := z_2z_3...$ und für jedes i gilt $uv^iw \in \Sigma^* \subseteq L'$.
- $k = 1$: Hier gilt $z \in L$. Wir wählen wieder $u := \varepsilon$, $v := z_1 = \#$, $w := z_2z_3....$ Für jedes $i \neq 1$ gilt $uv^iw \in \{\#^iw : w \in \Sigma^*\} \subseteq L'$, und für $i = 1$ ist $uv^iw = z \in L'$ bereits klar.
- $k = 2 \vee k > 3$: Nun verwenden wir für v die ersten beiden Zeichen, also $u := \varepsilon$, $v := z_1z_2 = \#\#$, $w = z_3z_4...$. Somit kann uv^iw nie mit genau einer Raute anfangen und $uv^iw \in L'$.
- $k = 3$: Ähnlich zum Fall $k = 1$ wählen wir $u := \varepsilon$, $v := z_1 = \#$, $w := z_2z_3....$ Für jedes i gilt $uv^iw \in \{\#^jw : j \geq 2, w \in \Sigma^*\} \subseteq L'$.

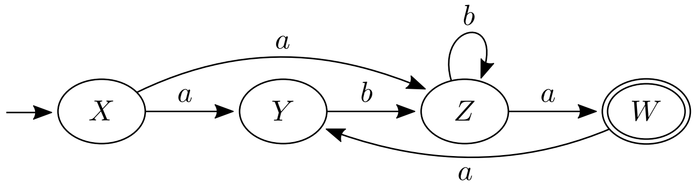
(c) Sei $L_1 := \{\#\}\Sigma^*$ die Sprache der Matrikelnummern, die mit genau einer Raute beginnen. Die Sprache L_1 ist offensichtlich regulär. Um einen Widerspruch zu erzeugen, nehmen wir an, dass L' regulär sei. Nach den Abschlusseigenschaften regulärer Sprachen wäre dann auch $L_1 \cap L'$ regulär, es gilt aber $L_1 \cap L' = L$ und wir haben bereits gezeigt, dass L nicht regulär ist. Dies ist ein Widerspruch, also kann L' auch nicht regulär sein.

Anmerkung: Es lässt sich auch auf direktem Wege beweisen, dass L' nicht regulär ist, z.B. mit der Methode über Residualsprachen, die wir im Kontext des Satzes von Myhill-Nerode kennen lernen werden.

Aufgabe H3.3. (*NFA zu RA*) 3+1 Punkte

(a) Konvertieren Sie den folgenden NFA M zu einem regulären Ausdruck, indem Sie ein geeignetes Gleichungssystem aufstellen und dieses lösen.

(b) Geben Sie einen äquivalenten NFA an, der 4 Zustände aber nur 5 Transitionen hat.



Lösungsskizze. **(a)** Wir stellen folgende Gleichungen auf:

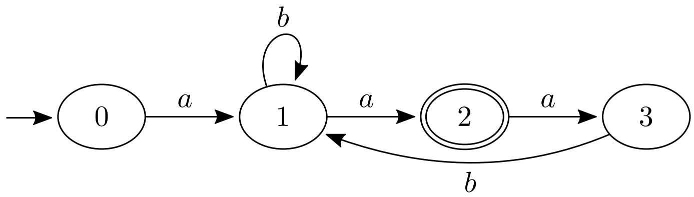
$$\begin{aligned} X &\equiv aY \mid aZ \\ Y &\equiv bZ \\ Z &\equiv bZ \mid aW \\ W &\equiv aY \mid \epsilon \end{aligned} \quad \overset{(1)}{\rightsquigarrow} \quad \begin{aligned} X &\equiv abZ \mid aZ \\ Y &\equiv bZ \\ Z &\equiv b^*aW \\ W &\equiv abZ \mid \epsilon \end{aligned} \quad \overset{(2)}{\rightsquigarrow} \quad \begin{aligned} X &\equiv abb^*aW \mid ab^*aW \\ Y &\equiv bb^*aW \\ Z &\equiv b^*aW \\ W &\equiv abb^*aW \mid \epsilon \end{aligned}$$

$$\overset{(3)}{\rightsquigarrow} \quad \begin{aligned} X &\equiv ab^*aW \\ Y &\equiv bb^*aW \\ Z &\equiv b^*aW \\ W &\equiv (abb^*a)^* \end{aligned} \quad \overset{(4)}{\rightsquigarrow} \quad \begin{aligned} X &\equiv ab^*a(abb^*a)^* \\ Y &\equiv bb^*a(abb^*a)^* \\ Z &\equiv b^*a(abb^*a)^* \\ W &\equiv (abb^*a)^* \end{aligned}$$

Erklärung: (1) Wir setzen $Y \equiv bZ$ bei X und W ein und wenden Ardens Lemma auf Z an. (2) Wir setzen $Z \equiv b^*aW$ bei X, Y und W ein. (3) Wir vereinfachen die Gleichung von X, da $L(abb^*aW) \subseteq L(ab^*aW)$, außerdem wenden wir Ardens Lemma auf W an. (4) Wir setzen $W \equiv (abb^*a)^*$ in den anderen Gleichungen ein.

Somit erhalten wir $L(M) = L(ab^*a(abb^*a)^*)$.

(b) Wir konvertieren unseren regulären Ausdruck direkt zu einem NFA. Zustände $0, 1, 2$ bilden das ab^*a Präfix ab, also ist 2 akzeptierend. Um die $(abb^*a)^*$ Schleife einzuleiten, geht der Automat zuerst von 2 zu 3 – da der b^*a Teil bereits am Anfang vorkommt, können wir aber direkt von 3 zu 1 gehen.



Aufgabe H3.4. (*Spieglein, Spieglein, an der Wand...*) 2+1+1 Punkte

Sei $\Sigma := \{a, b\}$. Wie üblich definieren wir w^R als die Spiegelung des Wortes w. Für eine beliebige Sprache L definieren wir nun $L^R := \{w^R : w \in L\}$.

(a) Zeigen Sie, dass L^R regulär ist, wenn L regulär ist.

(b) Sei M ein NFA mit n Zuständen. Konstruieren Sie einen NFA für $L(M)^R$ mit höchstens $n + 1$ Zuständen.

(c) Sei r ein regulärer Ausdruck. Konstruieren Sie einen regulären Ausdruck r' für $L(r)^R$ mit Länge $|r|$.

Hinweise: Sie müssen die Teilaufgaben nicht in der Reihenfolge lösen, in der sie gestellt wurden. Bei der Länge eines regulären Ausdrucks zählen wir Klammern nicht, die Länge von $(ab \mid \boldsymbol{\epsilon})^*$ wäre also 5. Bei (c) bietet es sich an, ähnlich zur strukturellen Induktion, ein rekursives Verfahren anzugeben.

Lösungsskizze. Wir beginnen mit Teilaufgaben (b) und (c). Aus letzterer folgt (a) dann unmittelbar.

(b) Idee: Wenn M ein Wort akzeptiert, gibt es einen Lauf von einem Start- in einen Endzustand. Der neue NFA soll diesen Lauf rückwärts ausführen; das bedeutet, wir wollen die letzte Transition zuerst rückwärts ausführen, dann die vorletzte, und so weiter. Wir müssen in einem Finalzustand starten, wir wissen jedoch nicht, in welchem. Dies wird der neue NFA am Anfang erraten. Wenn wir von irgendeinem dieser Finalzustände aus beim Rückwärtsausführen im Initialzustand landen, dann wollen wir das Wort akzeptieren.

Sei $M = (Q, \delta, \Sigma, q_0, F)$. Wir definieren uns einen NFA $M' := (Q', \delta', \Sigma, q_0', F')$ mit $Q' := Q \uplus \{q_0'\}$, wobei $q_0' \notin Q$ ein neuer Zustand ist und unser Startzustand wird. Wenn wir von q_0' ein Zeichen a lesen, gehen wir in einen beliebigen Zustand, den man von F mit einem a erreichen kann, wir setzen also

$$\delta'(q_0', a) := \delta'(F, a) \qquad \text{für alle } a \in \Sigma.$$

Ansonsten übernehmen wir die Transitionen aus M, drehen sie aber um:

$$\delta'(q', a) := \{q \in Q : q' \in \delta(q, a)\} \qquad \text{für alle } q' \in Q, a \in \Sigma$$

Es gibt also eine Transition von q' zu q in M' genau dann, wenn es eine von q zu q' in M gibt. Die akzeptierenden Zustände sind $F' = \{q_0\}$, falls $\varepsilon \notin L(M)$, sonst $F' = \{q_0, q_0'\}$.

Dieser NFA hat wie gefordert $n + 1$ Zustände.

(c) Wir geben die Konstruktion rekursiv an. Es gibt folgende Möglichkeiten:

- $r \in \{\boldsymbol{\emptyset}, \boldsymbol{\epsilon}\} \cup \Sigma$: Wir setzen $r' := r$, da hier $L(r) = L(r)^R$ gilt.
- $r = (s)^*$: Wir konstruieren rekursiv einen regulären Ausdruck s' mit $L(s') = L(s)^R$ und setzen $r' := (s')^*$. Dann gilt
$$L(r)^R = (L(s)^*)^R = (L(s)^R)^* = L(s')^* = L((s')^*) = L(r') \quad .$$
- $r = st$: Rekursiv gibt es reguläre Ausdrücke s', t' mit $L(s') = L(s)^R$ und $L(t') = L(t)^R$. Wir definieren $r' := t's'$ und erhalten
$$L(r)^R = L(st)^R = L(t)^R L(s)^R = L(t')L(s') = L(t's') = L(r') \quad .$$
- $r = s \mid t$: Wieder gibt es s', t' mit $L(s') = L(s)^R$ und $L(t') = L(t)^R$. Wir definieren $r' := s' \mid t'$ und erhalten
$$L(r)^R = L(s \mid t)^R = L(s)^R \cup L(t)^R = L(s') \cup L(t') = L(s' \mid t') = L(r') \quad .$$

In allen Fällen bleibt die Länge des regulären Ausdrucks erhalten.

(a) Sei L regulär. Dann gibt es einen regulären Ausdruck r mit $L(r) = L$. Wie wir in Teilaufgabe (c) gezeigt haben, existiert ein regulärer Ausdruck r' mit $L^R = L(r)^R = L(r')$, also ist L^R regulär.

Bonusaufgabe H3.5. (*Nach den Sternen greifen*) 3 Bonuspunkte

Sei L eine reguläre Sprache. Zeigen Sie, dass $L' := \{w : \{w\}^* \subseteq L\}$ regulär ist.

Lösungsskizze. Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA, der L akzeptiert. Die Idee ist, dass wir einen DFA M' für L' konstruieren, der sich nach Einlesen eines Wortes $w \in \Sigma^*$ merkt, wie M sich auf dem Wort w verhält. Dafür benötigen wir lediglich die Funktion $f_w : Q \to Q$, die wir definieren als $f_w(q) := \hat{\delta}(q, w)$ für alle Zustände $q \in Q$. Es gilt also beispielsweise $f_{ww}(q) = f_w(f_w(q))$; oder auch $f_{wv} = f_v \circ f_w$ für alle Wörter w, v. Beachten Sie, dass es zwar unendliche viele Wörter w gibt, aber nur endlich viele Funktionen f_w.

Schreiben wir nun f_w^k für die k-fache Anwendung von f_w (also $f_w^0(q) := q$ und $f_w^{k+1}(q) := f_w^k(f_w(q))$), dann ist gilt allgemein $f_{w^k}(q) = f_w^k(q)$ für alle $k \in \mathbb{N}$, und $w \in L'$ ist äquivalent zu $q_0, f_w(q_0), f_w^2(q_0), ... \in F$.

Für den DFA $M' = (Q', \Sigma, \delta', q_0', F')$ verwenden wir als Zustandsmenge die Menge aller Funktionen von Q nach Q (üblicherweise als Q^Q geschrieben), also $Q' := Q^Q$. Der Anfangszustand ist $q_0' := f_\varepsilon = \mathrm{Id}$, also die Identitätsfunktion. Wenn wir ein Zeichen $c \in \Sigma$ lesen, wollen wir von f_w zu f_{wc} übergehen, es soll also gelten

$$\delta'(f_w, c) = f_{wc} = f_c \circ f_w \qquad \text{für alle } w \in \Sigma^*$$

Der Term $f_c \circ f_w$ hängt nur von f_w (dem aktuellen Zustand) und c ab – insbesondere hängt es nicht unmittelbar von w ab. Es ist für einen Zustandsübergang also (glücklicherweise) nicht wichtig, mit welchem Wort wir den Zustand betreten haben.

Für einen Zustand $f \in Q'$ und Zeichen c definieren wir also $\delta'(f, c) := f_c \circ f$. Die Finalzustände sind nun $F' := \{f \in Q' \mid \forall i \in \mathbb{N} : f^i(q_0) \in F\}$; wie oben erläutert ist $f_w^i(q_0)$ der Zustand, den M erreicht, nachdem er das Wort w genau i-mal eingelesen hat. Wir merken an, dass es uns zwar genügen würde, die Menge F' mathematisch zu definieren, ohne dass wir angeben, wie man feststellt, ob ein bestimmter Zustand $q \in Q'$ nun tatsächlich dort enthalten wäre. Allerdings lässt sich die Eigenschaft $\forall i : f^i(q_0) \in F$ auch unschwer entscheiden: man berechnet sukzessiv die Folge $q_0, f(q_0), f^2(q_0), ...$ und überprüft, ob alle Zustände in F enthalten sind. Sobald sich ein Zustand wiederholt, kann man aufhören.

Abschließend beweisen wir noch formal, dass M' tatsächlich L' akzeptiert. Es gilt

$$w \in L' \iff \{w\}^* \subseteq L \iff \forall i : w^i \in L \iff \forall i : \hat{\delta}(q_0, w^i) \in F \iff \forall i : f_{w^i}(q_0) \in F$$

Da $f_{w^i} = f_w^i$ nach Definition von $\hat{\delta}$, erhalten wir also $w \in L'$ genau dann, wenn $f_w \in F'$. Es genügt also, $\hat{\delta}'(q_0', w) = f_w$ für alle $w \in \Sigma^*$ zu zeigen. Dies folgt über Induktion; als Basis erhalten wir $\hat{\delta}'(q_0', \varepsilon) = q_0' = \mathrm{Id} = f_\varepsilon$. Für den Induktionsschritt sei nun $w \in \Sigma^*$ und $c \in \Sigma$ beliebig.

$$\hat{\delta}'(q_0', wc) = \delta'(\hat{\delta}'(q_0', w), c) \overset{\text{IA}}{=} \delta'(f_w, c) = f_c \circ f_w = f_{wc}$$