Formale Systeme Automaten und Prozesse

Abgabe: 25.05.2017

Georg C. Dorndorf Matr. Nr. 366511
Adrian C. Hinrichs Matr. Nr. 367129
Jan Bordihn Matr. Nr. 364705

Die Aufgaben sind zur besseren Darstellung in umgekehrter Reihenfolge

Aufgabe 6

Seie ein Alphabet Σ und alle regulären Ausdrücke über Σ als $\mathcal{P}$ gegeben.

Lösung: Der geforderte Algorithmus sei als Rekursive Funktion $f : \mathcal{P} \to \mathbb{F}_2$ modelliert, wobei 1 bedeutet, dass der reguläre Ausdruck nur endlich lange Wörter akzeptiert und 0 bedeutet, dass der reguläre Ausdruck auch unendlich Lange Worte akzeptiert. Der Algorithmus hat dann die folgenden (hierarchischen) Vorschriften:

$a \in \Sigma; R, S \in \mathcal{P}; \star \in \{*, ^+\}$

$$f(\varnothing) = 1 \quad \text{(I)}$$
$$f(\varnothing\star) = 1 \quad \text{(II)}$$
$$f(\varepsilon) = 1 \quad \text{(III)}$$
$$f(\varepsilon\star) = 1 \quad \text{(IV)}$$
$$f(a) = 1 \quad \text{(V)}$$
$$f(R\varepsilon) = 0 \quad \text{(VI)}$$
$$f(R\varnothing) = 0 \quad \text{(VII)}$$
$$f(R*) = 0 \quad \text{(VIII)}$$
$$f(R^+) = f(RR*) \quad \text{(IX)}$$
$$f(RS) = f(R) \cdot f(S) \quad \text{(X)}$$
$$f(R + S) = f(R) \cdot f(S) \quad \text{(XI)}$$

[1] Eine implementation des Algorithmus in der Sprache Haskell sieht wie folgt aus:

```
f :: [Char] -> Bool -> Bool
f (')':ss:s) b = b && if ss=='0' then (f ('0':s)
    ↪ True) else (f ('a':ss:s) True)
f (')':s) b = b && (f ('a':s) b)
f ('+':s) b = b && (f s b)
f ('0':s) _ = f s True
f ('0':'*':s) _ = f s True

f ('E':s) _ = f s True
f (a:'*':[]) _ = False
f (a:'*':s:ss:sss) b = if (s=='0') then (f (ss:
    ↪ sss) True)
                       else (if (s==')' && ss ==
                           ↪ '0') then (f sss
                           ↪ True )
                            else (b && (f (s:ss:sss)
                                ↪ False)))
f (a:'?':s) _ = f (a:a:"*") False && (f s True)
f (a:s) _ = f s True
f [] _ = True
```

[1] Offensichtlich gilt für alle $R \in \mathcal{P} : (R) = R$, weshalb tatsächlich alle regulären Ausdrücke (wie in der Vorlesung definiert) von dem Algorithmus untersucht werden können

Aufgabe 5

a)

$r = (a^* + b)^*$

$r = \emptyset$

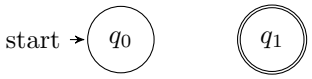

Abbildung 1: Thompson-Konstruktion: leere Eingabe

$r = \varepsilon$

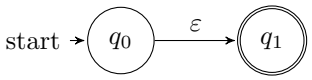

Abbildung 2: Thompson-Konstruktion: ε Eingabe

$r = a$

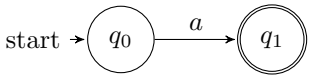

Abbildung 3: Thompson-Konstruktion: a Eingabe

$r = b$

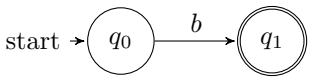

Abbildung 4: Thompson-Konstruktion: b Eingabe

$r = a^*$

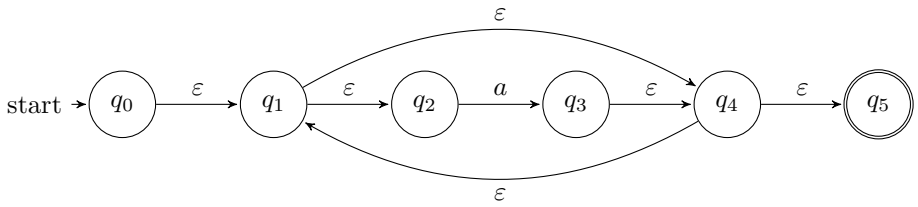

Abbildung 5: Thompson-Konstruktion: a^* Eingabe

$r = a^* + b$

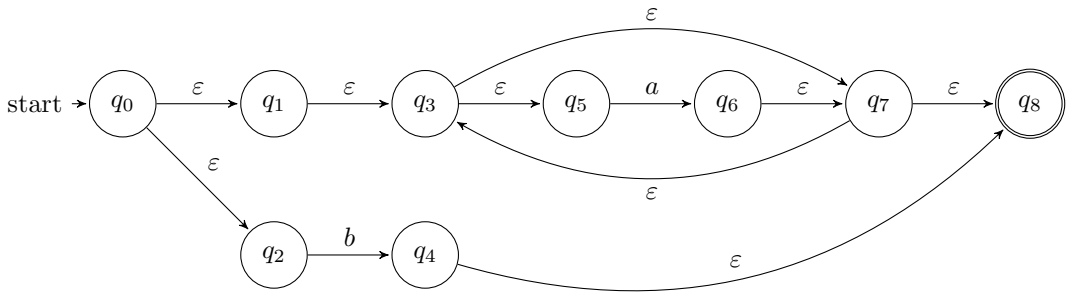

Abbildung 6: Thompson-Konstruktion: $a^* + b$ Eingabe

$r = (a^* + b)^*$

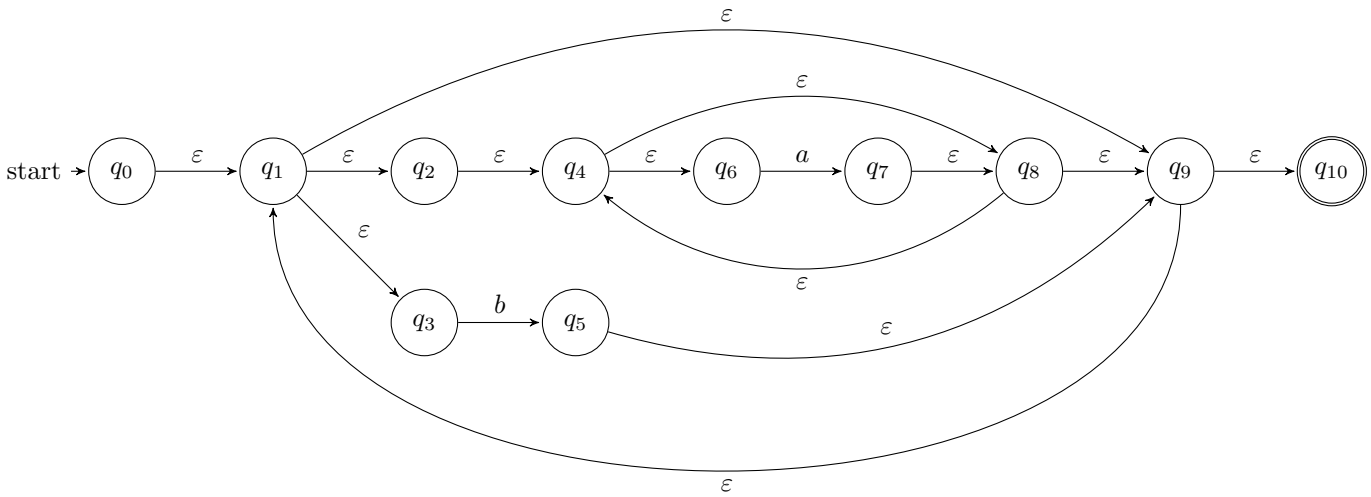

Abbildung 7: Thompson-Konstruktion: $(a^* + b)^*$ Eingabe

b)

Es ist zu erkennen, dass der Automat aus Abbildung 7 eine ε-Kreis über die Zustände q_1, q_2, q_4, q_8, q_9 besitzt. Dieser Kreis sorgt dafür, dass zum ein beliegt oft gelesen werden kann, so dass gewisse Teile des Autoatmens

Georg C. Dorndorf Matr. Nr. 366511
Adrian C. Hinrichs Matr. Nr. 367129
Jan Bordihn Matr. Nr. 364705

übersprungen / beliebig oft gelesen werden kann. Dieser Kreis kann in einem Zustand vereinfacht werden, da die ganzen ε-Transitionen nicht relevant für die Eingabe sind. Somit sind nur die Transitionen von Bedeutung, welche vom ε-Kreis abgehen. Dies können aber auch direkt von einem Zustand alle abgehen. Beim Zusammenfassen des Kreises muss naben den abgehenden Transitionen, überprüft werden ob durch die ε-Transitionen ein Enzustand erreicht werden kann. Ist das der Fall, ist der neue Zustand - für dne Kreis - auch ein Endzustand.
Beim Kontrahieren des Autoatmens aus Aufgabenteil a (Abbildung 7) entfallen die Zustände des ε-Kreise. Zudem entfallen fast alle Transitionen des ε-Kreises, außer eine, um nach gelesene as oder bs zum Anfang des Autoatmens zu kommen. Weiter wird der Typ des Zustandes q_{10}, welcher nach dem Ende des ε-Kreises in q_9 folgt auch der Typ des neu einzufügen Zustandes q_c, Es ergibt sich folgender Automat.

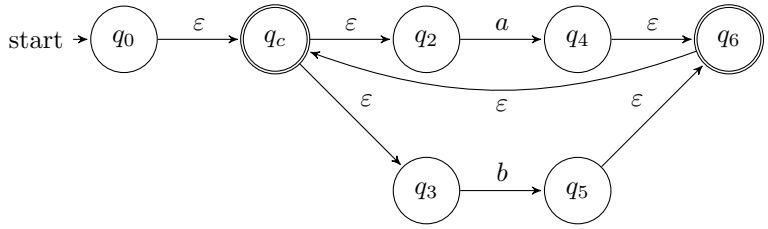

Abbildung 8: NFA ohne ε Kreis

Um die Äquivalenz beider Automaten zu zeigen, betrachten wir erst die möglichen Eingaben. So akzeptiert der Automat das leere Worte, über die Zustände q_0, q_1, q_9, q_{10}, als auch eine beliebige Folge von as oder bs. In dem Automat (Abbildung 8) kann auch nur diese Eingabe, ohne einen geeichten Kreis zu besitzen, gelesen werden. Denn dadurch das der neu eingefügt Zustand q_c ein Endzustand ist, wird dass leere Worte erkannt. Eingaben von mehren as oder bs werden durch die ε-Transition von q_6 zu q_c erkannt.

c)

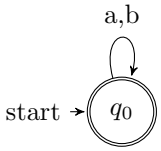

Abbildung 9: reduzierter NFA ohne ε-Transitionen