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

Aufgabe 5

a)

b)

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:

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

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

[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)
```

[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

```
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
```