Theoretische Informatik 2

Sommersemester 2018

Elizaveta Kovalevskaya
Georg Schnitger
Philipp Tertel

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

Übungsblatt 3

Ausgabe: 02.05.18
Abgabe: 08.05.18

Aufgabe 3.1 ***Beschreibungslänge*** (3+2+3+2+3 = 13 Punkte)

Für einen regulären Ausdruck R bezeichne $|R|$ die Anzahl der Symbole – inklusive aller Klammern – von R. Wie groß können die Unterschiede in der Beschreibungslänge von DFAs, NFAs und regulären Ausdrücken werden?

a) Sei R ein regulärer Ausdruck. Zeigen Sie: Es gibt einen zu R äquivalenten ε-NFA N mit höchstens $\mathcal{O}(|R|)$ Zuständen. (Ein NFA mit ε-Übergängen heißt ein ε-NFA. N ist äquivalent zu R, wenn $L(N) = L(R)$ gilt.)

   Fazit: NFAs besitzen mindestens die Beschreibungskraft regulärer Ausdrücke.

b) Der reguläre Ausdruck $r := (0|1)^* \cdot 1 \cdot (0|1)^{k-1}$ hat die Länge $\mathcal{O}(k)$, während DFAs für $L(r)$ mindestens 2^k Zustände besitzen. Dieses Beispiel zeigt bereits einen fast größtmöglichen Unterschied in der Beschreibungslänge auf.

   Sei R ein regulärer Ausdruck. Zeigen Sie: Es gibt einen DFA für $L(R)$ mit höchstens $2^{\mathcal{O}(|R|)}$ Zuständen.

c) Wie groß können zu einem DFA äquivalente reguläre Ausdrücke im schlimmsten Fall werden?

   Sei A ein DFA mit Zustandsmenge Q. Zeigen Sie: Das dynamische Programmierverfahren im Beweis des Satzes von Kleene produziert einen regulären Ausdruck der Länge höchstens $|\Sigma| \cdot |Q| \cdot 2^{\mathcal{O}(|Q|)} = |\Sigma| \cdot 2^{\mathcal{O}(|Q|)}$.

   Hinweis: Wie sieht die Rekursionsgleichung aus, die das dynamische Programm für $R^k_{p,q}$ benutzt?

d) Gegeben seien reguläre Ausdrücke R_1 und R_2. Zeigen Sie, dass es einen regulären Ausdruck R gibt, sodass $L(R) = L(R_1) \cap L(R_2)$ und $|R| \leq |\Sigma| \cdot 2^{\mathcal{O}(|R_1| \cdot |R_2|)}$.

   Hinweis: Wie kann man aus zwei NFAs A_1, A_2 einen NFA A mit möglichst wenigen Zuständen bauen, so dass $L(A) = L(A_1) \cap L(A_2)$ gilt?

e) Wir definieren die Sprache W_n, die für DFAs sehr leicht, für reguläre Ausdrücke aber sehr schwierig ist. Das Alphabet ist die Menge $\Sigma_n = \{1, \ldots, n\} \times \{1, \ldots, n\}$. W_n besteht aus allen Worten der Form

   $$(1, i_1)(i_1, i_2)(i_2, i_3) \cdots (i_{k-1}, i_k)$$

   über Σ_n mit $k \geq 1$. Man kann zeigen, dass jeder reguläre Ausdruck für W_n mindestens die Länge $\Omega(2^n)$ besitzt. Konstruieren Sie einen DFA für W_n mit möglichst wenigen Zuständen.

   Fazit: Eine Explosion der Länge regulärer Ausdrücke (im Vergleich zur Zustandszahl von DFAs) ist also unausweichlich.

Bitte wenden!

Aufgabe 3.2 *Abschlusseigenschaften* (3+4 Punkte)

Sei L eine reguläre Sprache über dem Alphabet Σ.

a) Sei $K \subseteq \Sigma^*$ eine *beliebige*, nicht notwendigerweise reguläre Sprache. Zeigen Sie, dass der Quotient

$$L/K := \{\, u \in \Sigma^* \;:\; \text{es gibt } v \in K, \text{ so dass } uv \in L \,\}$$

eine reguläre Sprache ist.

b) Zeigen Sie, dass die Sprache $\frac{1}{2}L := \left\{\, u \;:\; \text{es gibt } v \text{ mit } uv \in L \text{ und } |u| = |v| \,\right\}$ regulär ist.

Aufgabe 3.3 *Entscheidungsprobleme* (2+2 Punkte)

a) Gegeben sind zwei DFAs $A_1 = (\Sigma, Q_1, \delta_1, q_0^1, F_1)$ und $A_2 = (\Sigma, Q_2, \delta_2, q_0^2, F_2)$. Geben Sie einen Algorithmus an, der in Zeit $\text{poly}(|Q_1| + |Q_2|)$ entscheidet, ob $L(A_1) \subseteq L(A_2)$ gilt.

Hinweis: Bauen Sie einen DFA A aus A_1 und A_2, sodass $L(A) = \emptyset \iff L(A_1) \subseteq L(A_2)$.

b) Zeigen Sie, dass das *Wortproblem für reguläre Ausdrücke* effizient lösbar ist: Für einen regulären Ausdruck R über dem Alphabet Σ und ein Wort $w \in \Sigma^*$ ist in Zeit $\text{poly}(|R| + |w|)$ zu entscheiden, ob $w \in L(R)$ gilt.