

Übungen zur Vorlesung
Effiziente Algorithmen
SS 2002
Blatt 5

Aufgabe 5.1 (4 Punkte)

1. Sei h eine beliebige natürliche Zahl. Wie viele Befehle sind bei der UNION-FIND-Datenstruktur mit Bäumen mindestens nötig, damit ein Baum mit Tiefe von mindestens h entsteht? Geben Sie eine Folge von möglichst wenigen Befehlen an, so dass ein Baum mit Tiefe h entsteht.
2. Für die UNION-FIND-Datenstruktur mit Bäumen und Path-Compression sei eine Befehlsfolge von insgesamt m UNION- und FIND-Befehlen gegeben, wobei jedoch zuerst alle UNION- und dann erst alle FIND-Befehle ausgeführt werden.

Zeigen Sie, dass der Algorithmus diese Befehlsfolge in Zeit $O(m)$ ausführt. Wo scheidert diese Analyse, wenn die UNION- und FIND-Befehle in beliebiger Reihenfolge ausgeführt werden dürfen?

Aufgabe 5.2 (4 Punkte)

Gegeben seien zwei DFAs $M = (Q, s, F, \Sigma, \delta)$, $M' = (Q', s', F', \Sigma, \delta')$, wobei Q und Q' die Zustandsmengen, s und s' die Startzustände, F und F' die Mengen der akzeptierenden Zustände, Σ das Eingabealphabet und δ, δ' die Zustandsüberföhrungsfunktionen bezeichnen. Es sei $n := |Q| + |Q'|$. Wir nehmen an, dass $|\Sigma|$ konstant ist. Zwei DFAs heißen äquivalent, wenn sie die gleiche Sprache akzeptieren.

Entwerfen Sie einen Algorithmus, der in Zeit $O(n \log^* n)$ bestimmt, ob M und M' äquivalent sind.

Lösungshinweise:

Wir erlauben die Schreibweise $\delta(q, x)$ für $q \in Q$ und $x \in \Sigma^*$. Dabei ist $\delta(q, \varepsilon) = q$, $\delta(q, x_1 x_2 \dots x_m) = \delta(\delta(q, x_1), x_2 \dots x_m)$. Zwei Zustände $q \in Q$, $q' \in Q'$ heißen äquivalent, wenn für alle $x \in \Sigma^*$: $\delta(q, x) \in F \Leftrightarrow \delta'(q', x) \in F'$ gilt. Offensichtlich sind M und M' genau dann äquivalent, wenn s und s' äquivalent sind. Außerdem gilt, dass falls zwei Zustände q und q' äquivalent sind, dann auch $\delta(q, a)$ und $\delta'(q', a)$ für alle $a \in \Sigma$ äquivalent sind.

Aufgabe 5.3 (4 Punkte)

Das Problem des String Matching kann als reguläre Sprache beschrieben werden. Sei Σ das Alphabet. Das Problem, in einem Text das Muster $p_1p_2 \dots p_m \in \Sigma^m$ zu finden, wird durch die Sprache SM , definiert durch

$$SM := \{xp_1p_2 \dots p_my \mid x, y \in \Sigma^*\},$$

beschrieben. Aus der Vorlesung GTI ist bekannt, dass es einen DFA für SM gibt.

1. Zeigen Sie, dass jeder DFA für SM mindestens $m + 1$ Zustände hat.
2. Beschreiben Sie explizit einen DFA für SM , der mit $m + 1$ Zuständen auskommt.

Aufgabe 5.4 (4 Punkte)

Sei Σ ein Alphabet.

Geben Sie einen Algorithmus an, der zu zwei Zeichenfolgen

$$a_0a_1 \dots a_{n-1} \in \Sigma^* \text{ und } b_0b_1 \dots b_{n-1} \in \Sigma^*$$

gleicher Länge in Rechenzeit $O(n)$ bestimmt, ob es ein k mit $0 \leq k \leq n - 1$ gibt, so dass $a_i = b_{(k+i) \bmod n}$ für $0 \leq i < n$.

Ausweichtermin für die Übungsgruppe 5 (Hubert Wagner), die am Donnerstag, den 9. Mai wegen des Feiertags nicht stattfinden kann: Montag, 13. Mai von 08:15-10:00 in OH 16, 205.