

2. Übungsblatt zur Vorlesung Kombinatorische Algorithmen, SS 13

Abgabe: Bis Montag, 06.05.2013, 12:00 Uhr,
in den Abgabekasten vor dem AG-Flur oder per Email an `wild@cs.uni...`

2. Aufgabe

1 + 4 Punkte

In dieser Aufgabe betrachten wir eine Verallgemeinerung des *String Matching Problems*: Wir suchen nicht nach einem exakt festgelegten Pattern-String P , sondern nach einem bestimmten „Muster“, wobei wir annehmen, dass das gesuchte Muster durch eine *reguläre Sprache* $L \subseteq \Sigma^*$ beschrieben ist. In dieser Aufgabe werden wir zeigen, dass man auch für diesen allgemeineren Fall effiziente Algorithmen angeben kann.

Sie dürfen dabei jeweils annehmen, dass L in einer der üblichen endlichen Repräsentationen gegeben ist. (Wie Sie aus der Theorie der formalen Sprachen wissen, lassen sich diese Repräsentationen effektiv ineinander überführen.)

Weiterhin nehmen wir L als *fest gegeben* an, d. h. die folgenden Asymptotiken hängen *nicht* von (der Größe der Repräsentation von) L ab.

- a) Geben Sie einen Algorithmus an, der für ein Wort $w \in \Sigma^n$ in Zeit $\mathcal{O}(n)$ entscheidet, ob w dem Muster entspricht, d. h. ob $w \in L$ gilt.
- b) Entwerfen Sie einen Algorithmus, der das *Regular Substring Matching Problem* löst: Gegeben eine reguläre Sprache L und ein Text $T \in \Sigma^n$, finde alle Paare von Indizes $(i, j) \in [1 : n]^2$ mit $i \leq j$, sodass $T_{i,j} \in L$.

Ihr Algorithmus soll asymptotisch optimale Laufzeit haben, d. h. in Zeit $\mathcal{O}(n + k)$ laufen, wobei k die Anzahl Matches, also Paare (i, j) mit $T_{i,j} \in L$, ist. Weniger effiziente Lösungen erbringen Teilpunkte, je nachdem wie weit sie von der optimalen Laufzeit entfernt sind.