
Algorithmen auf Sequenzen

Abgabetermin: Donnerstag, den 29. November vor der Vorlesung

Aufgabe (Notenbonus) 1

Sei $t \in \Sigma^*$ und $k \in \mathbb{N}$ gegeben. Entwirf einen Algorithmus mit linearer Laufzeit, der alle kürzesten Teilwörter von t findet, die genau k -mal in t auftreten.

Hinweis: Korrektheitsbeweis und Laufzeitanalyse nicht vergessen!

Aufgabe (Notenbonus) 2

Wende den beschleunigten Algorithmus von Stoye und Gusfield (Abb. 3.20 im Skript) auf das folgende Wort

$$t = t_1 \cdots t_{13} = \text{abbabbabbabba}$$

an. Gib dazu für jeden Knoten v seine Blattlisten (getrennt nach $LL(v')$ und $LL'(v)$), sein DFS-Intervall ($\text{DFS_Int}(v)$) sowie das DFS-Intervall ($\text{DFS_Int}(v')$) für die ausgewählte längste Blattliste an. Gib weiter für jeden Knoten die ausgeführten Tests (basierend auf den DFS-Intervallen) und deren Ergebnis an (und ggf. das ausgegebene rechtsverzweigende Tandem-Repeat).

Aufgabe 3

Sei $\Sigma = \{a, b\}$. Konstruiere eine unendliche Familie $\mathcal{F} \subseteq \Sigma^*$ von Zeichenreihen über Σ , so dass jedes $t \in \mathcal{F}$ mindestens $f(|t|)$ exakte Repeats besitzt, wobei $f(n) = \omega(n)$ ist.