



Blatt 7

Aufgabe 7.1

- (a) Überlegen Sie sich einen in der Grösse eines Suffixbaums linearen Algorithmus, der für jeden inneren Knoten entscheidet, ob dieser links-divergent ist oder nicht und diesen entsprechend dieser Eigenschaft markiert. (3 Punkte)
- (b) Beschreiben Sie eine Methode, mit der alle maximalen Paare eines Strings S der Länge n gefunden werden können. Diskutieren Sie die Korrektheit und Laufzeit dieser Methode. Welchen Speicherplatzbedarf hat Ihr Algorithmus (abgesehen vom Bedarf für den üblichen Suffixbaum)? Modifizieren Sie den Algorithmus so, dass der Speicherplatzbedarf $\mathcal{O}(n)$ nicht übersteigt. (5 Punkte)

Hinweis: Für jeden inneren Knoten v des Suffixbaums von S sollen maximal soviele Listen erzeugt werden, wie es Buchstaben im Alphabet gibt. Die Liste im Knoten v , die zum Buchstaben x gehört, soll alle Anfangspositionen von Teilstrings des Strings S enthalten, die mit der Pfadbeschriftung von v im Suffixbaum übereinstimmen und als linkes Zeichen ein x haben. Aus diesen Listen können nun die maximalen Paare konstruiert werden.

- (c) Was müsste an Ihrem Algorithmus zum Finden aller maximalen Paare geändert werden, wenn nur maximale Paare einer minimalen Länge m gesucht sind? Welchen Einfluss hat das auf die Laufzeitabschätzung? (2 Punkte)

Aufgabe 7.2 Gegeben sei ein Alphabet A und das Metazeichen \star mit $\star \notin A$. Es gelte die Konvention, dass das Metazeichen \star einen beliebigen String aus A^* repräsentiert. Nun sei das Alphabet $\bar{A} = A \cup \{\star\}$ gegeben. Jeder String aus \bar{A} repräsentiert somit eine Menge von Strings aus A^* , wir bezeichnen sie als Suchmuster. Wir wollen im folgenden solche Muster in einem String T mit Hilfe des Suffixbaums $ST(T)$ suchen.

- (a) Gegeben sei das Suchmuster $S_1 \star S_2 \in \bar{A}$ mit $S_1, S_2 \in A^*$. Konstruieren Sie einen Algorithmus, der das Muster einmal in T findet (bzw. das Nichtauftreten feststellt). (3 Punkte)
- (b) Gegeben sei nun das allgemeine Suchmuster $S_1 \star S_2 \cdots \star S_k$. Konstruieren Sie einen Algorithmus, der das Muster einmal in T findet (bzw. das Nichtauftreten feststellt). (2 Punkte)
- Schzen Sie die Laufzeit ihres Algorithmus ab. (2 Punkte)