Advanced Text Indexing Techniques

Besprechung in den Übungen am 28.4.2009

Exercise 1
Give the suffix- and LCP-array for $T = \text{CTTCTTCTCTA}$. Also draw the corresponding suffix tree!

Exercise 2
We saw in the lecture that there are two possibilities for storing links to the children of a node in the suffix tree: as fixed arrays of length $\sigma$, and as arrays of variable length ($O(n\sigma)$ versus $O(n)$ total space for the tree).

a) Draw the arrays next to the nodes from the suffix tree of Exercise 1.

b) Show the class-definition (in Java, C++, ...) of a node for each of the two cases. Do not forget that the second possibility should support $O(\log \sigma)$-search for the correct child, given a letter $a \in \Sigma$.

Exercise 3

a) Look again at the first implementation of exercise 2 (fixed-length lists): give an example (as general as possible; i.e., possibly valid for all text lengths $n$) of a text $T$ where $O(n\sigma)$ space is used.

b) Look again at the second implementation of exercise 2 (variable-length lists): give an example (as general as possible) of a text $T$ and a pattern $P$ where search for $P$ takes $O(m \log \sigma)$ time.

Exercise 4
Show how suffix trees can be used to find efficiently:

a) The longest repetition in a text $T$ (possibly overlapping, e.g. $aba$ in $T = \text{ababaa}$).

b) The longest \textit{non-overlapping} repetition.

c) \textit{All} non-overlapping repetitions in $T$.

d) The \textit{longest string} occurring in both $T^1$ and $T^2$ (e.g. $aba$ in $T^1 = \text{babaa}$ and $T^2 = \text{aaaba}$).