Space-efficient CSA construction

Nicola Prezza

References and Reading

[1] Section 11.3 of: Navarro, Gonzalo. Compact data structures: A practical approach. Cambridge University Press, 2016.

Exercises

1 CSA construction Our goal in this exercise is to build space-efficiently a compressed suffix array. Working space will be measured considering only the data structures used *on top* of the read-only input text T (which can stay on disk). As usual, we assume that T ends with a special character \$ lexicographically smaller than all other alphabet characters and not appearing elsewhere in T.

1.1 Consider the Burrows-Wheeler Transform $BWT(T)$ of a text $T \in \Sigma^n$, and let $c \in \Sigma$. What are the differences between $BWT(T)$ and $BWT(cT)$? Hint: think of the BWT as the last column in the matrix of the sorted circular permutations of T. How does this matrix change if we add a character c at the beginning of T?

1.2 Based on the solution of exercise 1.1, propose a fast algorithm that builds the BWT in $O(nH_0(T) + \sigma w)$ bits of working space, where $\sigma = |\Sigma|$.

1.3 Show how to obtain the suffix array and the Ψ array of T given $BWT(T)$. Can we build directly a compressed Ψ array from $BWT(T)$? (i.e. without building the uncompressed Ψ and then compressing it).