UNIVERSITY OF LIVERPOOL

Department of Computer Science
Dr. Sebastian Wild · Ben Smith

Date: 2022-10-26
Version: 2022-10-26 13:48

# Tutorial 5 for COMP 526 – Efficient Algorithmics, Fall 2022

## Problem 1 (Fibonacci language and failure function)

The sequence of Fibonacci words $(w_i)_{i \in \mathbb{N}_0}$ is defined recursively:

$$\begin{aligned} w_0 &= \texttt{a} \\ w_1 &= \texttt{b} \\ w_n &= w_{n-1} \cdot w_{n-2} \qquad (n \geq 2) \end{aligned}$$

Unfolding the recursion yields $w_2 = \texttt{ba}$, $w_3 = \texttt{bab}$, $w_4 = \texttt{babba}$, an so on.

(Note that the lengths $|w_0|, |w_1|, |w_2|, \ldots$ are *Fibonacci numbers*, hence the name. More precisely, we have $|w_n| = F_{n+1}$, with the Fibonacci numbers defined as $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$, for $n \geq 2$.)

a) Construct the transition function $\delta$ of the string-matching automaton for $w_6$ and draw the string-matching automaton.

b) Construct the failure link array *fail* and the draw the KMP automaton with failure links for $w_6$.

## Problem 2 (How KMP uses itself)

Recall the example $T = \texttt{abababaabab}$ and $P = \texttt{ababaca}$ used in the lecture to illustrate the KMP failure-link automaton.

1. Consider the string $S = S[0..m+n] = P\,\$\,T$ over the extended alphabet $\Sigma' = \Sigma \cup \{\$\} = \{\texttt{a}, \texttt{b}, \texttt{c}, \$\}$ and construct the failure-links array $\mathit{fail}[0..n+m]$.
2. Compare the result with the sequence of states from simulating the failure-link automaton for $P$ on $T$; what do you observe?
3. **Bonus:** Can you compute the values $\mathit{fail}[0..n+m]$ using only $\Theta(P)$ extra space? Here, it is enough to have the values available at some time during the computation; we (obviously) cannot store all of them explicitly in the allowed space.