Marburg University

Department of Computer Science
Prof. Dr. Sebastian Wild

Date: 2025-11-21
Version: 2025-11-20 17:22

Exercise Sheet 6 for Effiziente Algorithmen (Winter 2025/26)

Hand In: *Until 2025-11-28 18:00, on ILIAS.*

Disclaimer: English translations of our exercise sheets are provided as as best-effort service; in case of doubt, the German versions take precedence.

Problem 1

30 points

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

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

For example $w_2 = \mathtt{ba}$, $w_3 = \mathtt{bab}$, $w_4 = \mathtt{babba}$, etc. (The lengths of the words $|w_0|, |w_1|, |w_2|, \ldots$ are *Fibonacci numbers*, thus $|w_n| = F_{n+1}$, where the Fibonacci numbers are defined by $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$, for $n \ge 2$.)

- Construct the transition function δ of a string matching automaton for w_6, and draw the corresponding string matching automaton.
- Construct the failure link Array *fail* and draw the KMP automaton with failure links for w_6.

Problem 2

30 points

Apply the Boyer-Moore algorithm for the pattern $P = \mathtt{dacbdc}$ and the text $T = \mathtt{eacbdecbdcdabbccacbdc}$. As in the lecture, draw a table in which each row aligns the pattern with the substring of T being compared. Justify which rule was applied step-by-step, and indicate how many characters were compared in that step.

Problem 3

20 + 20 points

We consider a pattern P and a text T, where $|T| = n$, $|P| = m$, and $n \geq m \geq 1$.

a) Prove that every pattern matching algorithm must examine at least $\lfloor n/m \rfloor$ characters.

b) For any $m \geq 1$ and $n \geq m$, construct a pattern P and a text T so that the Boyer-Moore algorithm examines precisely $\lfloor n/m \rfloor$ characters. Justify your solution.

Problem 4

30 points

Suppose that a pattern P may contain a wildcard symbol τ. A wildcard symbol may match any substring (including the empty substring). For example, the pattern `ab`τ`ba`τ`b` matches the string `cabcdbab`: the first wildcard matches `cd` while the second one matches the empty string.

Describe, in words, an algorithm which, given a pattern $P[0..m)$ that can contain wildcards, produces a finite automaton A. The automaton A must find an occurrence of P in a text $T[0..n)$ in $\mathcal{O}(n)$ time. Justify the correctness of your solution.

Problem 5

20 + 20 + 10 points

In this exercise you will implement the Knuth-Morris-Pratt algorithm in Java and then extend it. Solve the following subexercises.

a) Create the method `boolean kmp(String text, String pattern)`, implementing the Knuth-Morris-Pratt algorithm as shown in the lectures. Test your implementation in a `main` method using the text `ababbababa` with the following patterns:

   1. `abab`
   2. `aba`
   3. `aab`

b) Compare the runtime of your implementation from part a) with the substring search algorithm from Java (i.e., the `contains` method for strings). Can you construct inputs that show the advantages of each algorithm? If so, design an appropriate experiment and highlight your reasoning.

c) Write a method `int[] kmpWithPositions(String text, String pattern)`. Extend the implementation from part a) so that it returns every position in the text where a pattern begins. Repeat the test in the `main` function to output all such positions.