Saarland University
Jun.-Prof. Dr. Tobias Marschall
Dr. Marcel Schulz

1.Assignment - Algorithms for Sequence Analysis, SS 2018

Exercise 1: DFA & NFA (1+1+1=3 Theory)

(a) Draw an NFA over $\Sigma = \{A,C,G,T\}$ that accepts all strings $s \in \Sigma^*$ that fit to the pattern $P = $ `AT*G*TC`. The `*` symbol can be replaced by any sequence $s', s'' \in \Sigma^*$. For example, if $s' = $ `ACCGT` and $s'' = $ `A` , then $P = $ `ATACCGTGATC`.

(b) Draw a DFA that accepts the same language (i.e. the same set of strings).

(c) If M is an NFA that recognizes language C, does swapping the accept and non-accept states in M necessarily yield a new NFA that recognizes $\bar{C}$? Explain why or why not.

Exercise 2: Another DFA (1+1=2 Theory)

(a) Draw a DFA over $\Sigma = \{A,C\}$ that accepts all strings $s \in \Sigma^*$ that match the pattern $P = \mathtt{A}^{10}\mathtt{C}^{10}$. That is, ten times `A` followed by ten times `C`.

(b) Does a DFA that accepts all strings that match $P = \mathtt{A}^n\mathtt{C}^n$ for any $n \geq 1$ exist? Explain why or why not.

Exercise 3: Shift-And Algorithm (2+2=4 Theory, 3 Programming)

(a) A wildcard character # is a special character which matches any single character.
Example: the pattern `ATT#CG` matches `ATTTCG`, `ATTCCG`, `ATTGCG`, `ATTACG`.
Describe how the Shift-And algorithm can be modified to handle such wildcard characters.

(b) In multiple string matching, we are given a text T and set of patterns

$$P = \{p_1, p_2, p_3, ..., p_m\}$$

The goal is to find all matches between any of the patterns and the text.
Describe how the Shift-And algorithm can be modified so as to solve the multiple string matching problem.

(c) Implement the Shift-And algorithm for *multiple patterns*. Your program should take two parameters.

   (i) the name of an input file with the set of patterns. Patterns are line separated.

   (ii) the name of an input file with the text T to be searched.

   For each pattern, the output should contain a new line, start with the name of the pattern and followed by a comma-separated list of all positions in the text where the pattern occurrences end. Positions should be 0-based, i.e., counting starts at 0. **Example:** Assume a file `input_text.txt` exists and contains the text `GGAAGGAA` and also a file `patterns.txt` exists and contains the pattern `AA`, `GG` and `TT`. Then your program should work as follows:

```
$ ./program patterns.txt input_text.txt
AA 3,7
GG 1,5
TT
```

Remarks:

- There are 12 points to be earned on each assignment sheet.
- 50% of programing points and 50% of theoretical exercises are necessary to take the exam.
- You are allowed to work in groups of two.
- Hand in your solutions on paper (except for source code) by putting it in the letter box of Tobias Marschall in E2.1 (ground floor).
- Programming code is to be sent as a *tar.gz* package by mail to:
  - aryan.3264@gmail.com
- Source code is only considered if
  - it is in one of the languages Python, C, C++, or Java,
  - it is reasonably documented, commented, and readable,
  - command-lines for compiling and calling are provided,
  - compilation does not fail, and
  - executables are named according to:

    `lastname1_lastname2_assignment1_exercise3`
- For each programing task, there will be three input files. One is for you for testing and two are for us for grading. The latter two are kept secret and points are awarded based on whether your program computes the right answer for these two input files.
- Do not forget to mention your names and matriculation numbers on your solutions!

- Copying between groups will result in zero points for all involved groups!

Hand in date: Friday, May 4, before 10:00