RMCS 2019 Assignment 2
Implementing a Regular Expression Engine

SCIS UoH

Submit By: 02/11/2019
Honour Code: You are not expected to discuss any part of this assignment with your collegues.

In this assignment you have to implement the entire pipeline of a regular expression engine. Your implementation should consider the following regular expression constructs.

- Literals: Lower/upper case roman literals and numerals ([a-zA-Z0-9])
- Operations: concatenation eg. **ab**, union eg. **a|b**, repetition eg. **a*** and **a+**, grouping eg. **(a|b)***

The regular expression engine will proceed in 4 phases. Each phase will be realized by a corresponding program.

- PHASE1 - RE to NFA
- PHASE2 - NFA to DFA
- PHASE3 - DFA to Minimized DFA
- PHASE4 - Minimized DFA to String Acceptor

A program for a phase will take input from a file and write output to another file. For example the program for phase1 will read a regular expression as input from a file named 'p1.in' and write output to 'p1.out' the corresponding NFA (format as specified in **minor-1**). Then the program for phase2 will read input from 'p1.out' and write output to 'p2.out' and so on. The phase4 program will read the minimized DFA representation from 'p3.out' and will also read from 'p4.in' strings to be tested for accept/reject. 'p4.in' may contain multiple strings, separated by a newline, for each of which the phase4 program will output to stdout(screen) either *accepted* or *rejected*. **You will then write a shell script to glue together these separate programs, taking input from stdin for both the regular expression and the string to be checked.**

You may then go ahead and replace *grep* in your OS with your new implementation! ☺