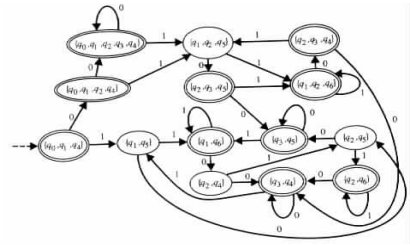

Theory of Automata
Fall 2014-Project

Due Date: 24th Nov, 2014 **Due Time: 11:00 a.m.**

In this project, you will implement a "**RecM- Regular Expression Combo Machine**". RecM will be used for many tasks related to REs, most important of which is searching acceptable strings in a given text file. Your program will take two inputs: a regular expression and a text file. As output, your program will print following files

NFA.txt	This file will have the complete NFA for RE, i.e. Alphabet, Start states, Final states and transition table.
DFA.txt	This file will have the complete DFA specification of the RE.
MinDFA.txt	This file will have the transition table of minimum DFA of the RE.
CFG.txt	Contains context free grammar for the given RE.
Output.txt	Contains those lines of the input text file, which contain strings that match the given RE.

Definition of a Regular Expression

For definition of the RE we will assume that the alphabet, Σ, will consists of all printable ASCII characters except the Meta characters. Meta characters are: (,), |, *, and &.

Use the following operators for RE: + for union, * for kleen star, and | for concatenation. Also note that * is a unary operator.

For simplicity you may assume that regular expression do not contain the symbols ∧ and ∅.

Project Parts

The project will consist of Multiple parts:
Part I: Construction of an NFA from a given regular expression.

Part II: DFA construction and minimization using partition method.

Part III: Processing the given input text file with the NFA/DFA.

Process the input text file line by line. That is, begin with the first line of text and check whether the NFA accepts it or not. If it accepts then print that line in output.txt, if it does not accept then print nothing. Proceed in this fashion until the end of the text. .

Part IV: Generate CFG for the given regular expression. You can generate the CFG easily using the resultant DFA of part II.

Bonus: Give a user-friendly interface, stretch your imagination and come with a creative output, input layout.

Jumbo Bonus: Given a CFG your RecM should be able to tell whether it accepts a regular language or a non-regular language.

Hint: It is easier to process a regular expression if it is in postfix form. For example, the postfix equivalent of the regular expression "(a+b)*x|z*" is given by the expression "ab+*xz*|". So, before processing the regular expression, you can choose to convert it into postfix form.

Motivation: A similar but much powerful tool **JFLAP** (Java Formal Languages and Automata Package) was developed by Susan Rodger and her students. This tool has been downloaded over 64 000 times by people in 161 of the 193 countries in the world - this was already a few years ago. http://www.jflap.org/tutorial/

Instructions: (Please read very carefully!)

1. Submit your group members name by 1st Nov, 2014 to your course instructor.
2. The project is in a group of two and no group of three will be allowed, as long as there is a genuine case of a person getting left out.
3. Start early so that you could finish in time.
4. You must provide source code in Java or C/C++. You are not allowed to use libraries or APIs for project tasks. Submit your code in a zip file.
5. Comment and indent your code clearly.
6. You must prepare a clear detailed report of your work including your "reasoning" (design and implementation details) and at least "5 test runs" on large files with regular expressions of different complexities. Present this report both as hardcopy and also soft copy in the zip file.
7. Creative and Interesting work will be given more credit then others.
8. Copy cases as always will not be tolerated at all. Projects have a higher weight in the grading, so you guys should make sure that you don't get a zero due to plagiarism.

9. Demos will be taken in groups, but there will be individual grading as well to insure that there are no free riders and both the group members have significantly contributed in the project.
10. Once the groups are submitted, there will not be any shifting in the groups (people moving from one group to another). You have to do your work in the same group that you have submitted to course Instructor.
11. The Project is to be submitted on slate before the submission time is over.

Good Luck