CSC 448: Compiler Design: 2015 Spring, Assignment #1 Solution by Yuancheng Zhang

Purpose:

To:

1. get you acquainted with the C-ish C++ used in the course.
2. go over broad compiler principles
3. go over some scanning theory
4. go over some scanning practice

Computing:

Please ssh into ctilinux1.cstcis.cti.depaul.edu, or use your own Unix machine. Assignment:

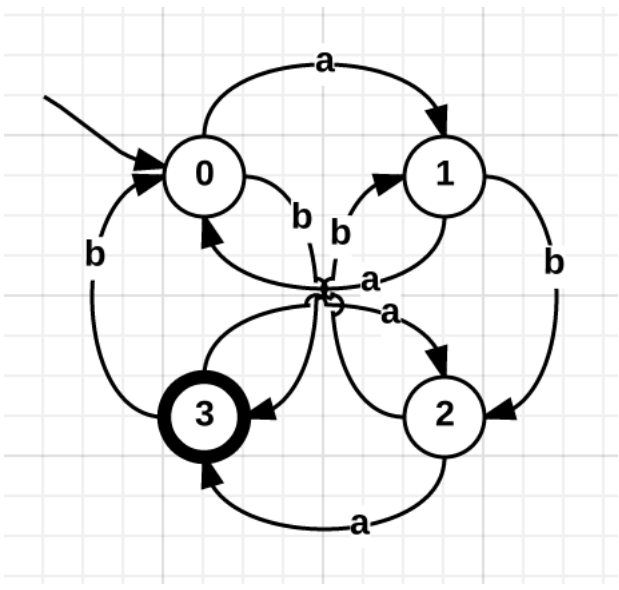
1. acCompiler.cpp (60 points)

Give the acCompiler.cpp program discussed in the first lecture the following functionality:

- Instead of using `i` to declare an integer variable, revise it to use `int`
- Instead of using `f` to declare a floating point variable, revise it to use `float`
- Instead of using `p` as the print command, revise it to use `print`
- Make it *identify by name* the redeclared variables
- Make it *show* the Symbol that cannot be converted in the `convert()` function.

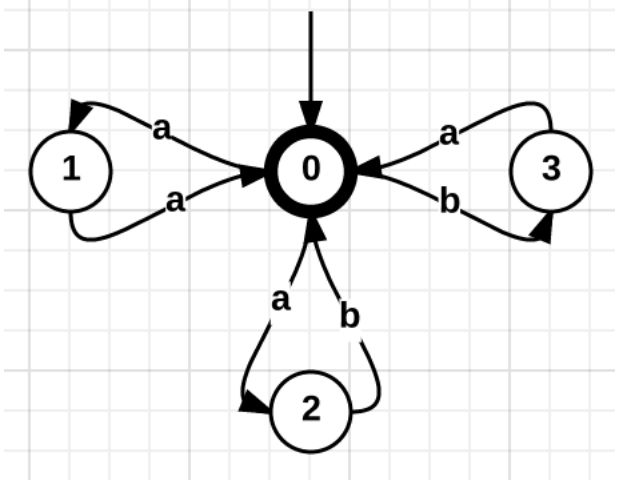
2. Finite Automata (10 points)

a. Create a deterministic finite state automaton that recognizes the subset of strings of (a|b)* with an even number of `a` characters and an odd number of `b` characters



b. Create a deterministic of non-deterministic finite state automaton that recognizes the set of all string in (a|b)* such that every block of four consecutive symbols contains at least two `a` s.

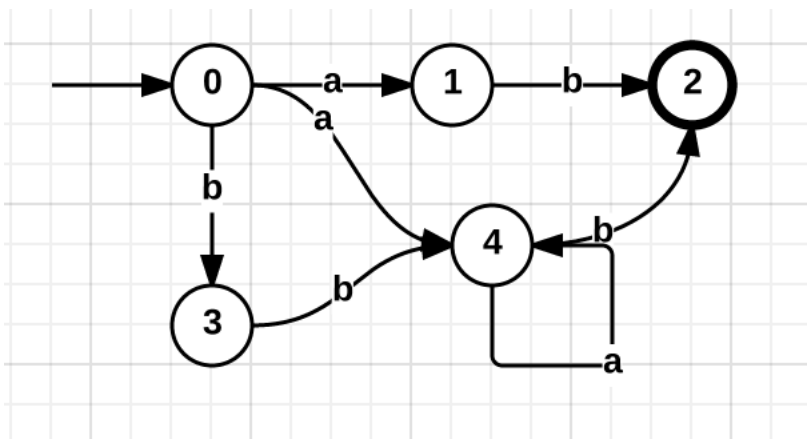
This situation is equal to **(aa|ab|ba)***



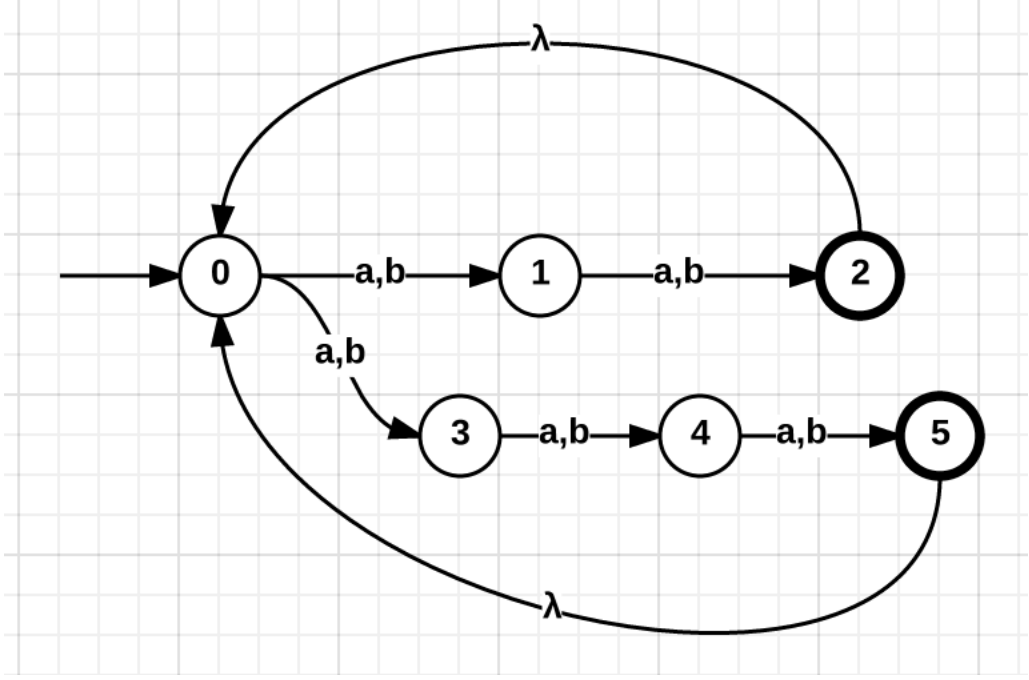
3. Regular Expressions to Finite Automata (10 points)

Please create deterministic or non-deterministic finite automata that would recognize the following regular expressions:

a. (ab | (a | bb)a*b)



b. ((a|b) (a|b))* | ((a|b) (a|b) (a|b))*



4. Regular expression to NFA to DFA (20 points)

a. Construct the NFA that corresponds to the regular expression **(a|b)*aba**

b. Construct the DFA that corresponds to the NFA of part (a). Please show some work.