22c:111 (CS:3820) Programming Language Concepts
Fall 2014
Homework Assigment 1
Due: Wednesday, Sep 24 at 11.59pm

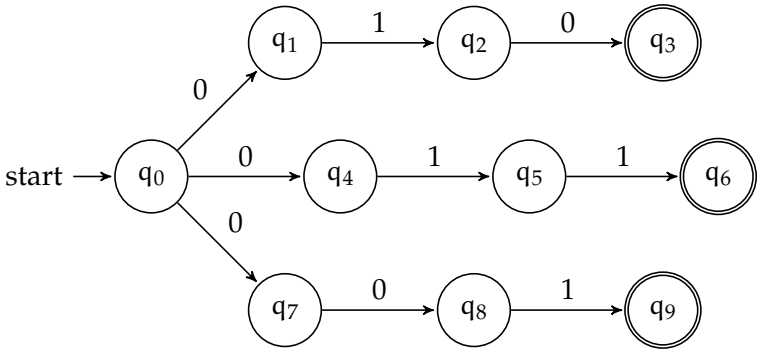
Solutions must be uploaded to the respective Dropbox section on ICON before the due date and time. Typed solutions can be submitted as one file in either of the following formats: PDF (strongly recommended), plain text, OpenOffice/LibreOffice, RTF or MS Word. Handwritten solutions are perfectly fine, but must be scanned and submitted as a *single PDF file* (no JPEGs). *Please make sure your handwriting is clearly legible on the scanned file*. Whatever the format, every page should clearly contain the name of the author and a page number.

1 Finite State Automata

1. Draw a DFA that accepts only those strings over $\Sigma = \{0, \ldots, 9, ,\}$ without leading zeroes that represent a natural number $\geqslant 0$ with the thousands separated by a comma.

   Your automaton must accept the strings "1,000", "1,000,000", and "42" and reject "9,87", "007", and ",101". Show a run of your automaton on each of those strings as a sequence of states. (8 points)

2. The following is an NFA. Why is it non-deterministic? Using the subset construction, give a DFA that accepts the same language. You are free work on the graph of the NFA or its transition table. (8 points)



2 UNIX Regular Expressions for Text Classification

Write a regular expression accepted by the UNIX tool `egrep`. We will use this tool to distinguish job adverts from other adverts. The job adverts must contain a salary figure that your regular expression must look for. Salaries may be given either on a per hour, week, month or year basis. They must appear preceded by a dollar sign, be given either as a dollar amount only, or in dollars and cents. There must be some word or words nearby (for our purposes on the same line) that identify a salary as opposed to a rental rate or a sale price.

Enter your regular expression between the quotes of the expression `REGEXP=""` in the fourth line of the script `hw1.sh` downloadable in the dropbox. Do not modify any other line. The script will call the `egrep` tool with the `-i` option to interpret your regular expression without regard to the case of letters in it. Therefore, the expression `word` will also match `Word` and `WORD`. While testing you will find it useful to call the script with the `-v` option to see which line your regular expression matches.

Run the script in a terminal on a Linux or Mac computer (or from the Cygwin environment in Windows) on the given test files `hw1-job-1.txt` to `hw1-job-3.txt` as well as `hw1-nojob-1.txt` to `hw1-nojob-3.txt`. It must return `Found` on the job adverts and `Not found` otherwise. Test your script on additional or modified adverts. In order to get full credit for this question, your script has to correctly classify all of our (hidden) test cases.

We will auto-grade your submission, so please make sure that exactly this one line is output and that your submission returns the correct answer on the test cases. Make sure that it does not output `Error` due to an invalid regular expression, because then we cannot give you any points for this question.

Hint: To use the dollar sign as a literal in a UNIX regular expression, you have to wrap it in square brackets as `[$]` to make it lose its special meaning as the end-of-line marker. (12 points)

3 Context-Free Grammars

1. Find a grammar that produces the language of the correctly parenthesized strings over the alphabet $\Sigma = \{(,)\}$. Show that it contains $()$, $(()())$, and $((())())$ by drawing a parse tree each.

   Define your grammar as the tuple $G = (N, T, P, S)$, and precisely state the set of non-terminal symbols N, the set of terminal symbols T, the set of productions P and the non-terminal symbol which is the start symbol. (8 points)

2. The grammar $G = (\{S, I\}, \{a, b, c\}, P, S)$, where P consists of the production rules

   $$S \to I \mid S + S \qquad\qquad I \to a \mid b \mid c$$

   is ambiguous.

   Demonstrate this, by giving a string that is the yield of two different parse trees. Then modify the grammar to be unambiguous, but so that it still accepts the same language. (8 points)