CMSC 330 Fall 2007
Homework 1: Ruby regular expressions, formal languages, and formal regular expressions

Posted on: Wednesday, Sep. 12, 2007
Due: Thursday, Sep. 20 9:20AM, 2007

Homeworks should be submitted in my office (4115 AVW) BEFORE 9:20AM (slip them under my door if it is closed). Homeworks will not be accepted after 9:20AM as the homework problems may be reviewed in class that day. No late homework is accepted. **Late homework will receive no credit.** Be sure to label the problem you are solving clearly with the problem number and subsection. Typing your homework is not required, but homework should be legible. **Illegible solutions will receive no credit.** Be sure to put your **name** on the homework. Note that this homework covers some material that you have not been taught as of Wednesday, Sep. 12. The lecture will have covered all relevant material by Thursday, Sep. 13 at the latest.

Remember, you must do this homework **on your own**, without any external sources. If you are unsure what is allowed, please contact the instructor.

1. **Ruby Regular Expressions**: Write a regular expression in Ruby to match each of the following sets of strings (or lines). Hint: You can check all your work by creating programs.

   (a) $\{w|w$ ends with your first name $\}$

   (b) $\{w|w$ contains exactly two a's $\}$

   (c) $\{w|w$ is valid output from the `pwd` command $\}$
   Sample outputs:
   `/`
   `/a/asteria.cs.umd.edu/junkfood/s/sorelle`
   Hint: If you're unsure about what a standard UNIX system would allow for directory names which could appear in this path, log on to your glue account and try making directories (`mkdir`) remembering that in most shells `\` can be used to escape special characters.

   (d) $\{w|w$ such that w contains only a's and b's and all characters appear in adjacent pairs $\}$

Sample strings in the language: “”, “aa”, “aabb”

Sample strings **not** in the language: “abba”, “b”, “cc”

2. **Formal Languages**: This problem tests your understanding of the formal language operations defined in lecture, which are necessary for understanding regular expressions.

   Consider the languages $A = \{a, aa\}$ and $B = \{bb, bbbb\}$. Show the languages denoted by each of the following:

   (a) A^3

   (b) AB^3

   (c) $A \cup B^*$

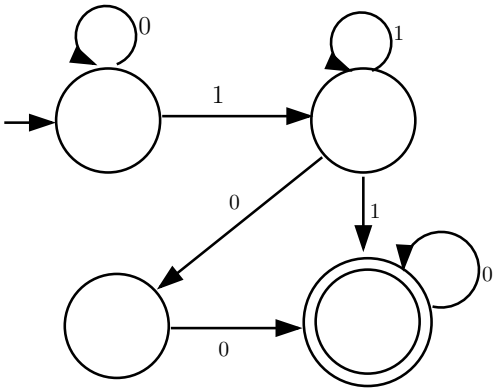
   (d) $(BB)^*$

3. **Formal Regular Expressions**: Give regular expressions for the following languages (represented by their description or by an NFA which accepts them). You should use only the formal regular expressions as defined in the notes.

   (a) Positive even decimal integers (initial 0's are not allowed). Example strings in the language: “0”, “12”, “254”. Example strings **not** in the language: “02”, “3”, “00”, “”, “-6”

   (b) Binary strings containing an odd number of 1's

   (c) Strings accepted by the NFA given below:



4. **NFAs**: Give NFAs for the following regular expressions.

   (a) $(0|1|2)^*10$

   (b) $(0^*1)^*|112$