CSE 30151 Theory of Computing: Homework 4
GNFAs, Regex, Pumping Lemma

Version 4: Sept. 20, 2017

Instructions

- Unless otherwise specified, all problems from "the book" are from Version 3. When a problem in the International Edition is different from Version 3, the problem will be listed as V3:x.yy/IE:x.zz, where x.zz is the equivalent number. When Version 2 has a different number, it will be listed as V3:x.yy/V2:x.zz. If either IE or V2 does not have a matching number, the problem text will be duplicated.
- You can prepare your solutions however you like (handwriting, LaTeX, etc.), but you must submit them in PDF. You can scan written solutions in the library or using a smartphone (with a scanner app like CamScanner). It is up to you to ensure that submissions are legible.
- Please give every PDF file a unique filename.
  - If you're making a complete submission (all problems), name your PDF file `netid-hw2.pdf`, where `netid` is replaced with your `NetID`.
  - If you're submitting some problems now and other problems later, name your file `netid-hw2-123.pdf`, where `123` is replaced with just the problems you are submitting now.
  - If you use the same filename twice, only the most recent version will be graded.
  - The time of submission is the time the most recent file was uploaded.
- If you use LaTeX and want to draw something like a state diagram, consider using the `tikz` package. A reference document is on the website under "Assignments".
- Submit your PDF file in Sakai. Don't forget to click the Submit (or Resubmit) button!

Practice Problems

These problems are from the book, and most have solutions listed for them. They are listed here for you to practice on as needed and any answers you generate **should not** be submitted. You are free to discuss these with others, but you are not allowed to post solutions to any public forum.

1. 1.10 Regex to NFA
2. 1.20 Regex to strings
3. 1.21a DFA to regex
4. 1.28 regex to NFA
5. 1.29 a,c Pumping Lemma to show non-regularity.

Book Exercises

These problems are found in the text book and are to be answered and submitted by each student. You are to solve them yourself. Use of solution manuals from any source or shared solutions is a violation of the ND Honor Code. You are also not allowed to show your solutions to another student.

1. (5 points) 1.13 Language to DFA and regular expression.
2. (5 points) 1.19a Regular expression to NFA
3. (5 points) 1.21b DFA to regex. Show steps
4. (5 points) 1.29b Pumping Lemma
5. (5 points) 1.37 Show a language is regular. (Hint: remember that if for the binary number x, $mod(x, n) = k$, then the number $mod(2x + b, n)$, where b is another binary digit, is the same as $mod(2k + b, n)$) As an example, $C_3 = \{0,\ 11,\ 110,\ 1001,\ 1100,\ 1111, ...\}$. Your solution should accept the binary digits one at a time from left to right.

Non-book Problems

The following problems are not found in the text book. You are to solve them yourself. Use of any resource you used other than the text book or class notes must be cited. You are also not allowed to show your solutions to another student.

6. (5 points) NOW A PRACTICE PROBLEM SINCE SOLUTION WAS ACCIDENTALLY POSTED (YOU NEED NOT TURN IN ANY SOLUTION). For the following NFA ompute the regular expression accepted by converting it to a GNFA and then using the CONVERT algorithm of from Lemma 1.60 on it. Show work. Feel free to use variables like R_i to stand for intermediate expressions on an edge (with R_i written out below), but show entire final answer. You don't need to simplify.

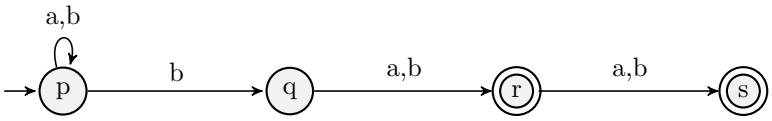
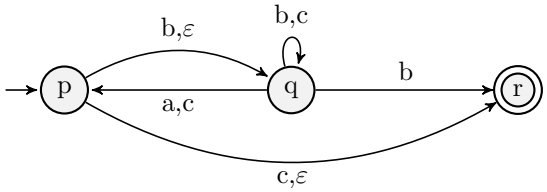


Figure 1: State diagram of NFA for question 6

7. (5 points) Do the same as above for the following NFA, i.e. compute the regex.



8. (5 points) NOW A PRACTICE PROBLEM SINCE SOLUTION WAS ACCIDENTALLY POSTED (YOU NEED NOT TURN IN ANY SOLUTION). Show using the Pumping Lemma that the language $L = \{(ab)^n(cd)^n \mid n \geq 0\}$ is not regular.
9. (5 points) REPLACEMENT PROBLEM: Show that the language generated from $\Sigma = \{a, b, c\}$ where the only constraint is that the number of a's equals the number of b's minus the number of c's, and there are never less b's than c's.

10. (5 points) A *string homomorphism* h is a function that takes each character in its input string some alphabet Σ^*, replaces it by a string (typically from a different alphabet Σ'), and concatenates all resulting substrings together, i.e. $h : \Sigma^* \rightarrow (\Sigma')^*$. As an example if $h(0) = ab$ and $h(1) = \varepsilon$, then $h(00110) = ababab$

    Prove that if L is a regular language over Σ, then the language $L' = \{h(w) | w \text{ is in } L\}$ is a regular language over Σ'. Hint: show by induction over the construction of regular expressions.

    Note: for any string x, $h(x)$ is always a single finite string.