Homework 2. Due October 25, 9:59PM.

CS181: Fall 2021

GUIDELINES:

- Upload your assignments to Gradescope by 9:59 PM.

- Follow the instructions mentioned on the course webpage for uploading to Gradescope very carefully (including starting each problem on a new page and matching the pages with the assignments); this makes it easy and smooth for everyone. As the guidelines are simple enough, bad uploads will not be graded.

- You may use results proved in class without proofs as long as you state them clearly.

- Most importantly, make sure you adhere to the policies for academic honesty set out on the course webpage. The policies will be enforced strictly. Homework is a stepping stone for exams; keep in mind that reasonable partial credit will be awarded and trying the problems will help you a lot for the exams.

- All problem numbers correspond to our text 'Introduction to Theory of Computation' by Boaz Barak. So, exercise a.b refers to Chapter a, exercise b.

---

1. Exercise 5.4. You can assume that $m \leq 2^n$ for the problem. [1 point]

   [Hint: We already had an upper bound on the number of circuits of size s in class (this also applied to multi-output functions so you can use that as is). Next, what is the total number of functions $f : \{0,1\}^n \to \{0,1\}^m$? To get an upper bound use the reasoning we used in class: Think of the large truth table containing the outputs for every possible input string and the number of ways to fill the table. (How many options per cell and how many cells?)

   Now compare the number of circuits to the number of m-bit output functions as we did in class. You don't have to redo the arithmetic calculations we did in class - you can use those simplifications as is.]

2. Answer the following for the DFA shown in the figure [.75 point]:
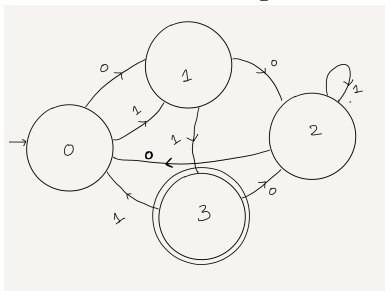
(a) What is the set of accept states?

(b) What sequence of states does the machine go through on input 1010101?

(c) Write down an accepting input for the machine.

3. Draw a DFA that accepts strings which contain 001 as a substring. For instance, 001100, 00110, 00000110111 would be accepted whereas 010100 should not be. [1 point]

[Hint: Think along the following lines: As you go over the input, you should initially skip all 1's. If you see a 0, then you may be onto something - it could be the first 0 of your sequence. If you see a 1 now, you can 'reset'. If you see a 0, you know you've seen two consecutive 0's. What should you look for next? You don't have to prove that you are correct - just the diagram with some explanations to make the grading easier would be enough.]

4. Use the construction we did in class to build a NFA for the Kleene star operation to build a NFA that accepts the strings $(\{01, 101\})^*$. [.75 point]

[Hint: Start with a DFA that only accepts $\{01, 101\}$ and then use the construction in class. You don't have to prove your construction works - just a diagram is enough (with some verbal explanations for clarity).]

5. For a language L, define $DROPFIRST(L)$ to be the language containing all strings that can be obtained by removing the first symbol of a string in L. Thus, $DROPFIRST(L) = \{x : bx \in L, \text{ for some } b \in \{0, 1\}\}$. Show that if L is computable by a DFA, then DROPFIRST(L) is computable by a NFA. [.5 points]

[Hint: Think of adding appropriate ε transitions from the start. Also, be careful and add a new start state instead of reusing old start state. You don't have to prove your construction works - just a diagram is enough (with some verbal explanations for clarity).]