Problem Set 5

Due: **10:00pm, Friday, 28 February**

This problem set focuses on the circuit size hierarchy, finite automata and regular expressions (Chapter 5 and 6 in TCS Textbook). Write your answers in the ps5.tex LaTeX template. You will submit your solutions in GradeScope as a PDF file with your answers to the questions in this template.

There are two "required" problems and three "optional" practices. There are no extra points from optional practice. If you work on them, we will grade your solution, and you will learn more.

Collaboration Policy: Some problems in this assignment have different policies, and they overrides the policies here.
You may discuss the problems with anyone you want. You are permitted to use any resources you find for this assignment **other than solutions from previous/concurrent CS3120 courses**. You should write up your own solutions and understand everything in them, and submit only your own work. You should note in the *Collaborators and Resources* box below the people you collaborated with and any external resources you used. You shall explicitly state the *content*, e.g., the main message in your collaborated discussion, the search keywords, the LLM/AI prompts, or the section in a book.

Collaborators and Resources: TODO: replace this with your collaborators and resources (if you did not have any, replace this with *None*)

To do this assignment:

1. Open this read-only Overleaf project located at https://www.overleaf.com/read/wqcqvybwfqcy#3ef453, and then copy this project.

2. Open your copy of the project and in the left side of the browser, you should see a file directory containing ps5.tex. Click on ps5.tex to see the LaTeX source for this file, and enter your solutions in the marked places. (You will also see the uvatoc.sty file, a "style" file that defines some useful macros. You are welcome to look at this file but should not need to modify it.)

3. The first thing you should do in ps5.tex is set up your name as the author of the submission by replacing the line, `\submitter{TODO: your name}`, with your name and UVA id, e.g., `\submitter{Haolin Liu (srs8rh)}`.

4. Write insightful and clear answers to all of the questions in the marked spaces provided.

5. Before submitting your ps5.pdf file, also remember to:

   – List your collaborators and resources, replacing the TODO in `\collaborators{TODO: replace ...}` with your collaborators and resources. (Remember to update this before submitting if you work with more people.)

   – Replace the second line in ps5.tex, `\usepackage{uvatoc}` with `\usepackage[response]{uvatoc}` so the directions do not appear in your final PDF. Starting from this Problem Set, we may deduct 3pt if you forgot this step.

Problem 1 *Random functions are hard. (TCS Exercise 5.8)*

Suppose $n > 1000$ in this problem. Suppose that we choose a function $F : \{0,1\}^n \to \{0,1\}$ at random, choosing for every $x \in \{0,1\}^n$ the value $F(x)$ to be the result of tossing an independent unbiased coin. Prove that the probability that there is a $2^n/(1000n)$-line program (namely, a NAND-straightline program of $2^n/(1000n)$ many lines) that computes F is at most 2^{-100}. (If you are stuck, see this exercise in the book for a hint.)

Note: This is another *existential* proof showing that there exists a hard-to-compute function. It is existential because we do not know whether the sampled function F satisfies the above circuit size.

Answer:

Problem 2 *Regular expressions*

For each of the following languages, give a regular expression that describes the language or explain why no such regular expression exists.

(a) $\{x \in \{0,1\}^* \mid x$ as interpreted as a binary representation of a natural number is divisible by $8\}$ (note that the empty string is interpreted as a binary representation of 0, which is divisible by 8).

(b) $\{x \in \{0,1\}^* \mid x$ does not contain the substring $011\}$

(c) $\{x \in \{0,1\}^* \mid$ every odd position of x is a $1\}$ (we will say the first bit is indexed by 0)

Answer:

Practice 1 *Equal to constant function. (TCS Exercise 5.3)*

For every $k \in \mathbb{N}$ and $x' \in \{0,1\}^k$, show that there is an $O(k)$ line NAND-CIRC program that computes the function $\mathrm{EQUALS}_{x'} : \{0,1\}^k \to \{0,1\}$ that on input $x \in \{0,1\}^k$ outputs 1 if and only if $x = x'$. After you finish, count the number of NAND gates in your circuit.

This is an *optional exercise*.

Note: Construct a NAND circuit (or NAND straightline program) that uses only 'NAND'. That is, encode the given constant x' using NAND gates. (Food for thought: what's the point of requiring this?)

Answer:

Practice 2 *Kleene*

One of our components for regular expressions is the *Kleene star*. Describe precisely the set of all languages that can be defined using regular expressions *without* a Kleene star. Provide an informal justification of your answer.

This is an *optional exercise*.

Note: You probably also learned another similar syntax of Kleene plus '+'. That is also excluded in this problem (just like Kleene star).

Answer:

Practice 3 *Closure properties of regular functions. (TCS Exercise 6.1)*

Suppose that $F, G : \{0,1\}^* \to \{0,1\}$ are regular functions. For each of the following definitions of H, either prove that H is always regular or provide a counterexample for regular F, G that would make H not regular.

- $H(x) = F(x) \wedge G(x)$.
- $H(x) = NAND(F(x), G(x))$.
- $H(x) = F(x^R)$, where x^R is the reverse of x, i.e., $x^R = x_{n-1}x_{n-2} \dots x_1x_0$.

This is an *optional exercise*.

Note: A regular function can be computed using either a DFA or a regular expression. Sometimes it is easier to think about the DFA, but sometimes it is easier to think about the regular expression of the function.

Answer:

Do not write anything on this page; leave this page empty.

This is the end of the problems for PS3. Remember to follow the last step in the directions on the first page to prepare your PDF for submission.

