Problem Set 4

Due: **10:00pm, Friday, 14 February**

This problem set focuses the Boolean circuit model for finite computation (Chapter 3 and 4 in TCS Textbook). Write your answers in the ps4.tex LaTeX template. You will submit your solutions in GradeScope as a PDF file with your answers to the questions in this template.

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/ggsvhxdwnxbg#34b764, 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 ps4.tex. Click on ps3.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 ps4.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 ps4.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 ps4.tex, `\usepackage{uvatoc}` with `\usepackage[response]{uvatoc}` so the directions do not appear in your final PDF.

Problem 1 $\{\mathrm{IF}, \mathrm{ZERO}, \mathrm{ONE}\}$ *is universal.*

Let $\mathrm{IF} : \{0,1\}^3 \to \{0,1\}$ be the function defined in textbook (Section 4.1.3), and let $\mathrm{ZERO} : \{0,1\} \to \{0\}$ be the constant function that always outputs 0, and similarly let $\mathrm{ONE} : \{0,1\} \to \{1\}$ be the constant function outputs 1. Consider the set of gates $\mathcal{G} = \{\mathrm{IF}, \mathrm{ZERO}, \mathrm{ONE}\}$.

Show that (1) NAND can be implemented by $\mathcal{G}$, and (2) $\mathcal{G}$ can be implemented by NAND.

Note: in Class 7, when we use LOOKUP_n to compute a function $f : \{0,1\}^n \to \{0,1\}$, we need to represent f as a 2^n-bit string s_f, and then the constant functions $(\mathrm{ZERO}, \mathrm{ONE})$ are needed to encode the string s_f.

Answer:

Problem 2 *XOR is not universal.*

Prove that for every n-bit input circuit C that contains only XOR gates, as well as gates that compute the constant functions ZERO and ONE, C is affine or linear modulo two, in the sense that there exists some $a \in \{0,1\}^n$ and $b \in \{0,1\}$ such that for every $x \in \{0,1\}^n$, $C(x) = \sum_{i=0}^{n-1} a_i x_i + b \mod 2$.

Conclude that the set $\{\mathrm{XOR}, \mathrm{ZERO}, \mathrm{ONE}\}$ is not universal.

Answer:

Problem 3 *Threshold function.*

Prove that there is some constant c such that for every $n > 1$, and integers $a_0, \dots, a_{n-1}, b \in \{-2^n, -2^n + 1, \dots, -1, 0, +1, \dots, 2^n\}$, there is a NAND circuit with at most cn^4 gates that computes the threshold function $f_{a_0,\dots,a_{n-1},b} : \{0,1\}^n \to \{0,1\}$ that on input $x \in \{0,1\}^n$ outputs 1 if and only if $\sum_{i=0}^{n-1} a_i x_i > b$ (that is, output 0 otherwise).

Note: You likely need to implement the circuit to prove the number of gates. Instead of directly using NAND, we recommend using circuits that compute addition, multiplication, and comparison of integers. You shall clearly describe your assumed circuit sizes, up to a constant factor. For example:

1. Circuit ADD that adds two n-bit integers: $\Theta(n)$ gates.
2. Circuit MUL that multiplies two n-bit integers: $\Theta(n^2)$ gates.
3. Circuit CMP that compares two n-bit integers (a, b) and outputs 1 iff $a > b$: $\Theta(n)$ gates.

If you use other circuits discussed in this class, you shall state their functionality and size clearly. If you use circuits NOT discussed in this class, you must fully implement them.

Note: this threshold function can be useful in neural networks. See textbook, Section 3.5.5: neural networks.

Answer:

Problem 4 *Prove that a perceptron cannot compute* XOR.

A *perceptron* is a single layer neural network (Section 3.4.5) that can be modeled by the following function:

$$f(x_0, x_1, \ldots, x_{k-1}) = \sigma(\sum_i w_i x_i)$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function. For this question, you may assume the activation function is a rectified linear unit (ReLU), commonly used in deep learning:

$$ReLU(x) = \max(0, x)$$

Prove that there is no way to define *XOR* using a perceptron. That is, show that there is no way to assign the values of w_i such that $f(x_0, x_1) = ReLU(w_0 x_0 + w_1 x_1)$ implements the *XOR* function. You can interpret the output of f as a Boolean value with values below 0.5 interpreted as False and values ≥ 0.5 interpreted as True.

Resource policy: The proof that a perceptron cannot compute *XOR* is of some historical importance, and it doesn't take much cleverness to find proofs of this. ***You should not search for solutions to this problem*** since the goal of it is for you to think about this yourself and come up with a proof. We think it will be pretty obvious if you write-up a found proof, so expect everyone to be able to explain their proof and how they derived it to us orally if asked. ***Do NOT click the links in this problem until you submit your solutions.***

Historical note: The historical significance of this problem, which is often overblown to the point where some refer to it as the "XOR affair", is that it has been attributed by some as one of the reasons why research in neural networks mostly ceased in the 1980s, except for a few die-hard believers who kept working on it, eventually leading to the explosion of "deep learning" over the past decade, and being awarded the Turing Award in 2018 and Nobel Prize in 2024.

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.

