Yash Shah

Homework 1. Due October 10, 9:59PM.

CS181: Fall 2022

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. Let S, T be two finite sets such that there is a one-to-one mapping from S to T and a one-to-one mapping from T to S. Show that $|S| = |T|$ (i.e., the two sets have the same number of elements). [1 point]
2. Exercise 2.4 [1 point]. (Note: The constant 1000 is there for slack and not a specific one.)
3. Prove that the set $\{AND, NOT\}$ is universal. [1 point]
4. Exercise 3.4. To be more precise, the problem is asking you to show that there is a function that **cannot** be computed by a Boolean circuit that is only allowed to use AND/OR (so NOT gates not allowed). As a further hint, you can show that there is a function that takes two inputs and has one output that cannot be computed in such a way (no matter how many AND/OR gates you use). [1 point]

ADDITIONAL PROBLEMS. DO NOT turn in answers for the following problems - they are meant for your curiosity and understanding.

1. Exercises 2.3, 3.1, 3.9, 3.10, 3.13, 3.14.

Problem 1

1. Let S, T be two finite sets such that there is a one-to-one mapping from S to T and a one-to-one mapping from T to S. Show that $|S| = |T|$ (i.e., the two sets have the same number of elements). [1 point]

Given that S, T are one-to-one mappings from S to T and T to S, we can deduce

$F: S \to T \implies x \in T$ has a single $F(x) \in T$ where $x \neq x' \in S \implies F(x) \neq F(x') \in T$

$G: T \to S \implies y \in T$ has a single $G(y) \in S$ where $y \neq y' \in T \implies G(y) \neq G(y') \in S$

By definition, $F: S \to T$ being a one-to-one mapping means that $|S| \leq |T|$.

Equivalently, $G: T \to S$ being a one-to-one mapping means that $|T| \leq |S|$

The 2 equalities $|S| \leq |T|$ and $|T| \leq |S|$ can only be satisfied under one condition as $|S| < |T|$ and $|T| < S$ cannot simultaneously be true. That leaves $|S| = |T|$ as the only possible cardinalities of S and T.

Therefore,

$\therefore |S| = |T|$

Problem 2

Exercise 2.4 — Representing graphs: upper bound. Show that there is a string representation of directed graphs with vertex set $[n]$ and degree at most 10 that uses at most $1000n \log n$ bits. More formally, show the following: Suppose we define for every $n \in \mathbb{N}$, the set G_n as the set containing all directed graphs (with no self loops) over the vertex set $[n]$ where every vertex has degree at most 10. Then, prove that for every sufficiently large n, there exists a one-to-one function $E : G_n \to \{0,1\}^{\lfloor 1000 n \log n \rfloor}$.

Given G_n is the set $[n]$ where every vertex has degree at most 10, we can define a function E that represents a circuit C as a DAG with $n+s$ vertices where n = input vertices, s = gate vertices and m of these labled as output

For this graph, let A_G be the adjacency list between vertices n and s, that is

$$a_{ns} = \begin{cases} 1 & \text{if } n \to s \in G_n \\ 0 & \text{otherwise} \end{cases}$$

For each $a_{n,s}$ where $s = m$, there at most 10 s gates for each n vertex. Therefore, each (n has 10 · s has 10) at most 10 times $\Rightarrow 10 \cdot 10 \cdot 10 = 1000$ for each n. However, since not all of these paths will lead to an output m as the graph is a DAG and each vertex has an input that leads to only one output (or the graph is cyclic or a vertex has multiple edges by definition), there is a binary multiplicate of paths for each n, that is, $\log(n)$. Therefore all paths can be determined in such a graph binarily with $\{0,1\}^{1000 n \log(n)}$

Henceforth, there is at most $1000 n \log(n)$ graph of vertices, and can be represented as a one-to-one function (as described above with properties of a DAG) $E : G_n \to \{0,1\}^{1000n\log n}$

Problem 3

3. Prove that the set $\{AND, NOT\}$ is universal. [1 point]

A universal gate is a set of gates such that every boolean function can be implemented with gates in this set. As proven in discussion 1D, $\forall f : \{0,1\}^n \rightarrow \{0,1\}^m \; \exists$ AND/OR/NOT Boolean circuit of size. Hence, the set {AND, OR, NOT} is universal.

We must prove that the set {AND, NOT} can construct an OR gate. For that will mean that {AND, NOT} = {AND, NOT, OR}, and thus universal.

Using de morgan's law, we can construct

$$\text{NOT}\Big(\text{AND}(\text{NOT A}, \text{NOT B}) = \text{NOT}(\text{OR}(A,B))\Big)$$

$$\downarrow$$

$$\text{NOT}(\text{AND}(\text{NOT A}, \text{NOT B})) = \text{NOT}(\text{NOT}(\text{OR}(A,B)))$$

$$\downarrow$$

$$\text{NOT}(\text{AND}(\text{NOT A}, \text{NOT B})) = \text{OR}(A,B)$$

Henceforth, using AND and NOT gates, we were able to create an OR gate. That implies the set {AND, NOT} contains the gate OR. Therefore, by transivity, since {AND, NOT, OR} is a universal gate, so is {AND, NOT}

$$\therefore \{\text{AND}, \text{NOT}\} = \{\text{AND}, \text{OR}, \text{NOT}\} \Rightarrow \text{universal}$$

Problem 4

Exercise 3.4 — AND,OR is not universal. Prove that for every n-bit input circuit C that contains only AND and OR gates, as well as gates that compute the constant functions 0 and 1, C is *monotone*, in the sense that if $x, x' \in \{0,1\}^n$, $x_i \leq x'_i$ for every $i \in [n]$, then $C(x) \leq C(x')$.

Conclude that the set $\{AND, OR, 0, 1\}$ is *not* universal.

For all n-bit input circuit C, we can deduce the following:

1) 2^n possible input circuits C → where $i \in \{0,1\}^{2^n}$ in range $[0, 2^n - 1]$
2) $2^{(2^n)}$ possible functions in C → where $f \in \{0,1\}^{2^{2^n}}$ in range $[0, 2^{2^n} - 1]$

If a circuit C only contains AND and OR gates, then the function must be non-negative in order to maintain the monotony of the circuit function and input states.

Say a function $f \in \{0,1\}^{2^{2^n}}$ is not monotonic. That would imply $i, i' \in \{0,1\}^{2^n}$, such that $f(i) = 1$ AND $f(i') = 0$, and $f(i \text{ OR } i') = i'$. This function could only maintain monotony if the NOT gate altered the 0 state with 1 — this cannot be achieved by the constant functions 0 and 1 because it would require a fundamental reconfiguration of the input — however, the use of the NOT gate would render the function negative function as the sign bit would flip.

However, such a function f could not be true because if $f(i) = 1$ AND $f(i') = 0$, $f(i \text{ OR } i')$ returns i not i'. Therefore, with AND and OR gates only in a circuit C, all n-bit input circuit C for any size will have C as monotone.

Since C is monotone, we know the NOT gate cannot be used. That means the NAND and NOR gates cannot be constructed, which by definition means the the set $\{AND, OR, 0, 1\}$ cannot be universal.