Q1 Prefix-free Definition

1 Point

When is an encoding $E : \mathcal{O} \to \{0,1\}^*$ prefix-free? That is, write down the definition of prefix-free encoding.

Definition 2.17 — Prefix free encoding. For two strings y, y', we say that y is a *prefix* of y' if $|y| \leq |y'|$ and for every $i < |y|$, $y'_i = y_i$.

Let $\mathcal{O}$ be a non-empty set and $E : \mathcal{O} \to \{0,1\}^*$ be a function. We say that E is *prefix-free* if $E(o)$ is non-empty for every $o \in \mathcal{O}$ and there does not exist a distinct pair of objects $o, o' \in \mathcal{O}$ such that $E(o)$ is a prefix of $E(o')$.

Q2 Converting to Prefix-Free

2 Points

In class we discussed a way to convert any encoding $E : \mathcal{O} \to \{0,1\}^*$ to be a prefix-free encoding.

Describe a way to do this in two or three sentences. [It is recommended to type your response here.]

Given any element $a \in O$

1. Compute $E(a)$
2. Convert all $1 \to 11$, $0 \to 00$
3. Add 01 to the end

Q3 Size?

1 Point

Is the following statement true or false: Every function $f : \{0,1\}^n \to \{0,1\}$ has a Boolean circuit of size at most $2^n/n^2$.

We showed $\frac{2^n}{8n}$ false*, so $\frac{2^n}{n^2}$ should be false as well (for large enough n)

*Theorem 5.3 Lower Bound

- ○ True
- ◉ False.

Q4 Equivalence?

1 Point

Which of the following circuit models (given by the gates specified in the choices) are equivalent to Boolean circuits?

Here, 0 denotes the constant 0 and 1 denotes the constant 1 function.

- ☑ $\{NAND\}$
- ☐ $\{AND, OR, 0, 1\}$ ← We showed this in homework!
- ☑ $\{AND, NOT\}$
- ☑ $\{NAND, AND, OR, NOT\}$

Q5 Code as data

1 Point

In class, we saw a way to encode NAND circuit programs as elements of $\{0,1\}^*$. What was the number of bits needed to encode a NAND circuit program of size s?

[You can use big-Oh notation.]

→ $O(s \log(s))$

★Recall how this comes from encoding a list of tuples

Q6 Circuits

1 Point

Which of the following are true:

Theorem from lecture 4

- [x] Every function $f : \{0,1\}^n \rightarrow \{0,1\}$ has a Boolean circuit of size at most $O(n2^n)$ computing it.
- [ ] Every Boolean circuit of size s has an equivalent NAND circuit of size at most s.

Size at most 3s

- [x] Every NAND circuit of size s has an equivalent Boolean circuit of size at most $2s$.

NOT(AND) produces NAND

- [x] There exists a NAND circuit for $Eval_{n,m,s}$ (the evaluation function as defined in class that takes the encoding of a NAND circuit, input and returns the evaluation of the circuit on the input) of size at most $O(s^2 \log s)$.

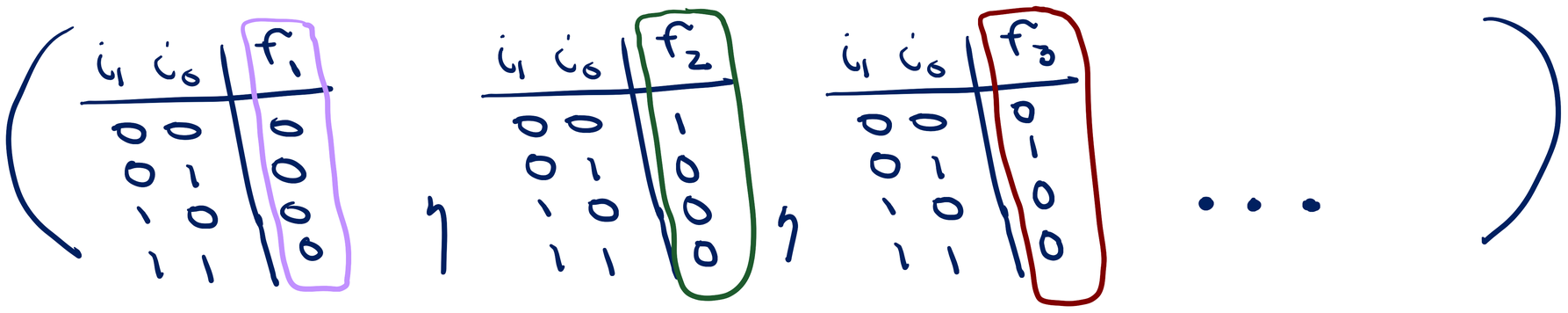
Universal Circuit

Q7 Function Count

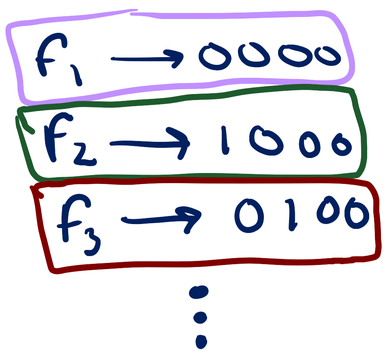
1 Point

How many functions are there from $f : \{0,1\}^n \to \{0,1,2,\ldots,n-1\}$?

- We know # functions is same as # truth tables
  - → e.g. $n=2$



- Each truth table has a unique combination of outputs
  - → e.g. $n=2$



- Therefore our problem becomes counting # n-ary strings of length $L = 2^n$

$$\Rightarrow \#\text{ functions} = n^L = \boxed{n^{2^n}}$$

Q8 State PECTT

1 Point

State the Physical Extended Church-Turing Thesis as discussed in class.

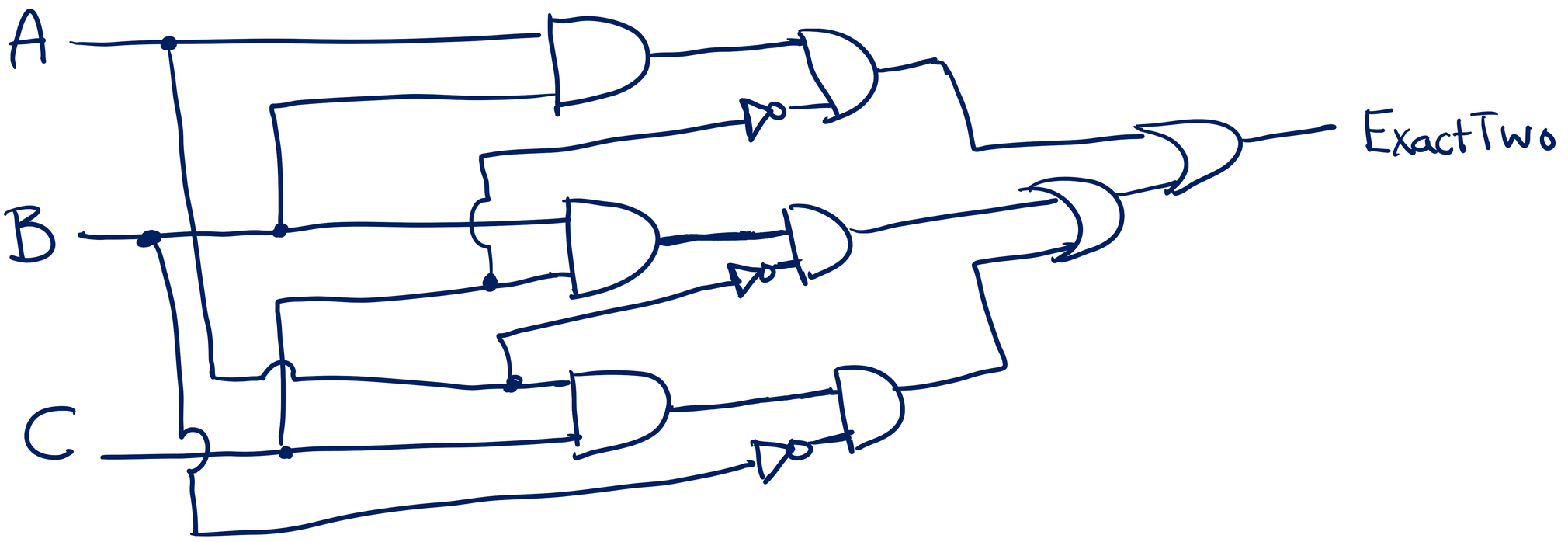
"Physical Extended Church-Turing Thesis" (PECTT): *If a function $F : \{0,1\}^n \to \{0,1\}^m$ can be computed in the physical world using s amount of "physical resources" then it can be computed by a Boolean circuit program of roughly s gates.*

Q9 Computing Exactly Two using AON

3 Points

Design a AND/OR/NOT circuit to compute the function $ExactTwo : \{0,1\}^3 \to \{0,1\}$ such that $ExactTwo(a,b,c) = 1$ if and only if exactly two of a, b, c are 1 and 0 otherwise. In otherwords, $ExactTwo(0,1,1) = ExactTwo(1,0,1) = ExactTwo(1,1,0) = 1$ and the function evaluates to 0 on all other inputs.

You can draw the circuit or write it as a AND/OR/NOT program.

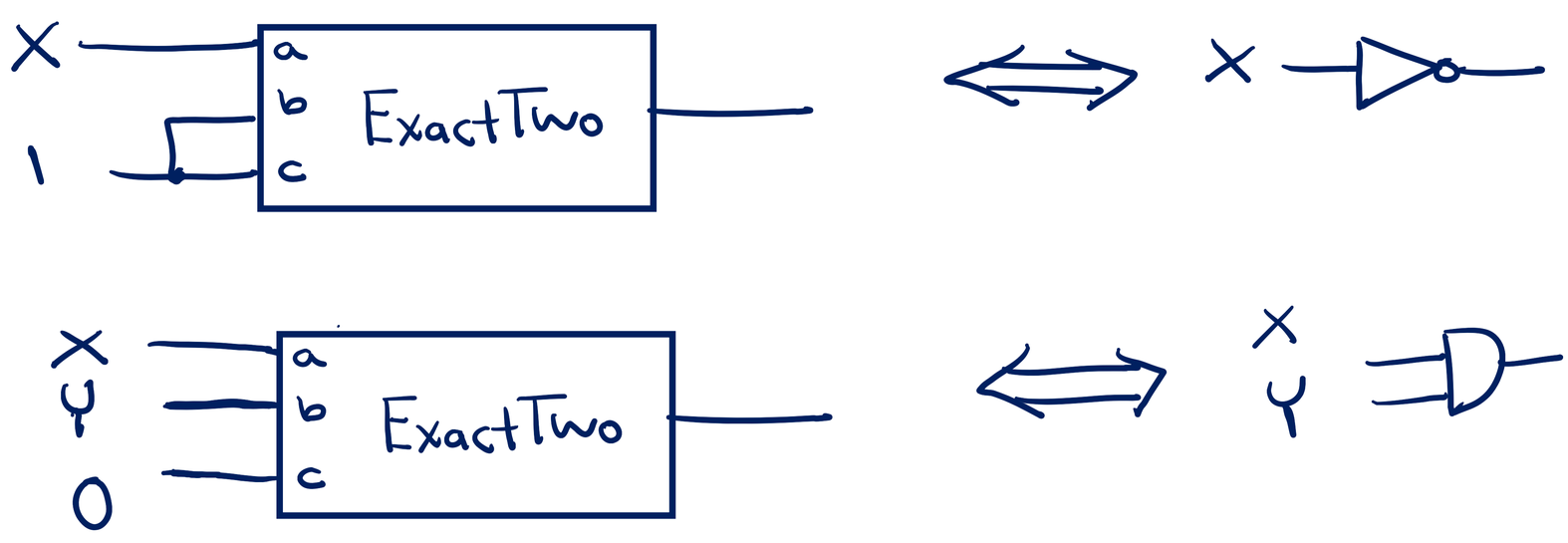


Q10 ExactTwo is Universal

3 Points

Show that $\{ExactTwo, 0, 1\}$ is a universal set of gates. Here, $ExactTwo : \{0,1\}^3 \to \{0,1\}$ is the function defined above and 0 is the constant zero function and 1 denotes the constant one function.

You can draw or write down the formulas for the computations as we did in class.

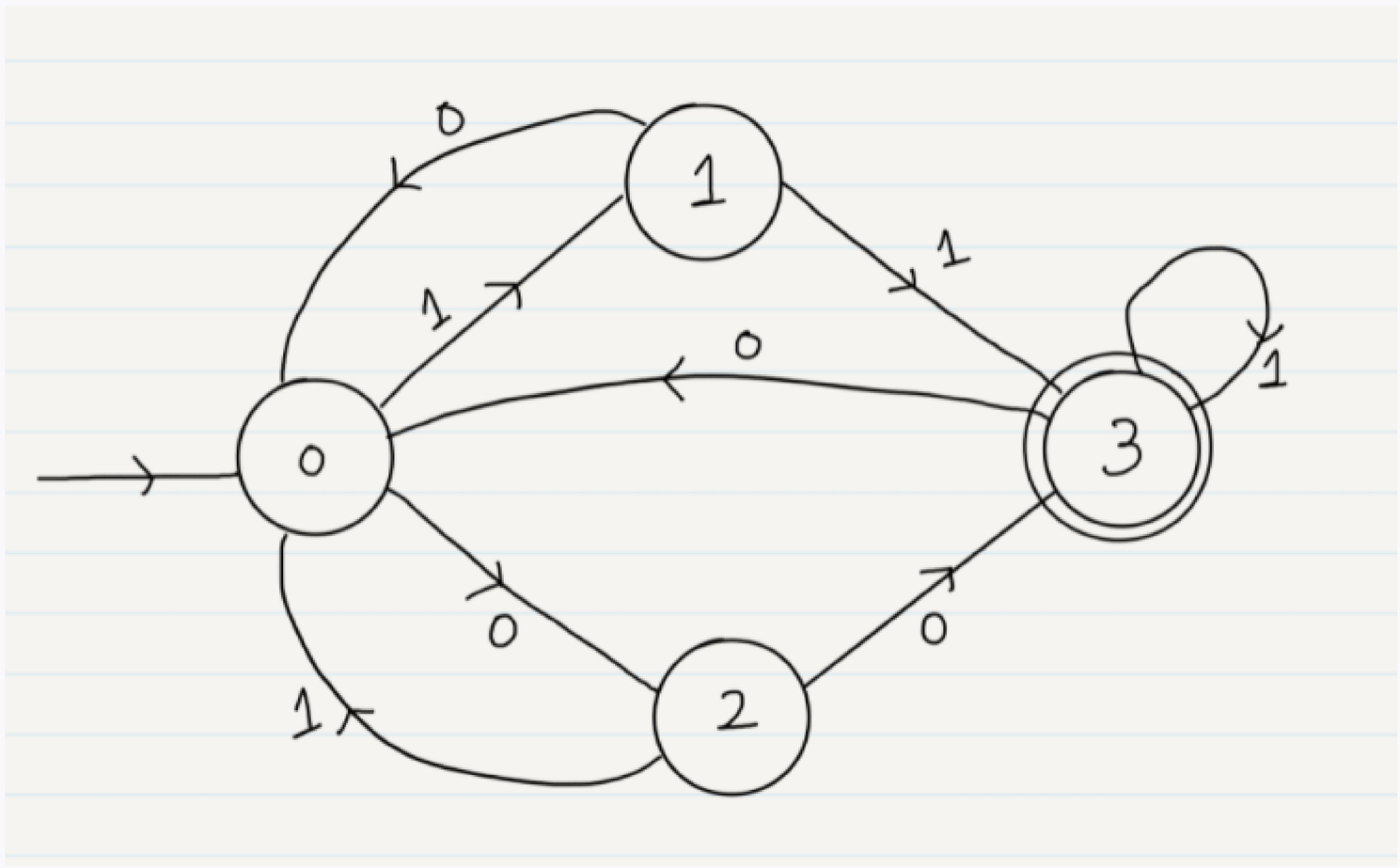


Chaining these together gives us NAND ⇒ universal circuit

Q11 Follow automata

2 Points

Write down the sequence of states visited by the automata D shown in the diagram on input $x = 0101001$. What is $D(x)$?



STATE	INPUT
0	0
2	1
0	0
2	1
0	0
2	0
3	1
3	

Because our last state = 3 ∈ Accept $\Rightarrow D(x)=1$

Q12 Designing a DFA

3 Points

Draw a DFA that accepts strings that contain 101 as a substring. That is, draw a DFA D such that $D(x) = 1$ if and only if x contains 101 as a substring. You don't have to prove that your DFA works.

For instance, $1010, 10001010, 11101011, \ldots$ should be accepted by the DFA whereas strings such as $01001, 11000110, 11001000, \ldots$ should not lead to an accepting state.

It would be best to draw the DFA and upload the image.