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.

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.]

[redacted]

[redacted] [redacted]

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.

- [x] $\{NAND\}$
- [ ] $\{AND, OR, 0, 1\}$ ← We showed this in homework!
- [x] $\{AND, NOT\}$
- [x] $\{NAND, AND, OR, NOT\}$

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.]

[redacted]

Q6 Circuits

1 Point

Which of the following are true:

Theorem from lecture 4

- [x] Every function $f : \{0,1\}^n \to \{0,1\}$ 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 3 s

- [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 $f:\{0,1\}^n \to \{0,1,2,\dots,n-1\}$?

[redacted]

Q8 State PECTT

1 Point

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

[redacted]

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.

[redacted]

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.

[redacted]

[redacted]

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.

[redacted]