Homework 1

Due On: February 16, 2021 3:30PM (CST)

Problem 1 (Search) Given a sequence of n bits $(x_1, \ldots, x_n)$ construct a circuit implimenting the function $f_{search} : (x_1, \ldots, x_n) \mapsto (x_1 \vee x_2 \vee \ldots \vee x_n, i_b)$. Here i_b is the binary representation (of size $\lceil \log(n+1) \rceil$) of the least index i (between 1 and n) such that $x_i = 1$, if $x_1 \vee x_2 \vee \ldots \vee x_n = 1$. Otherwise $i = 0$. That is, f_{search} is a search function for a list of 1-bit binary numbers. For example, $f_{serach}(001011) = (1, 011)$. Determine the size (number of gates) and the depth of your circuit. You may assume your basis set is $\{\text{NAND}\}$.

Problem 2 (BPP) In the definition of the complexity class BPP we required that $0 \leq \epsilon < \frac{1}{2}$. Suppose we relax this restriction to $0 \leq \epsilon \leq \frac{1}{2}$ instead. Let this new class be PP. Check if the proof given in class showing $BPP \subseteq P_{/poly}$ still holds if we replace BPP with PP. If not, then why not?

Problem 3 (Experimentation) Write a Python program that given an integer n, outputs the circuit for $f_{maj} : \mathbb{B}^n \to \mathbb{B}$. Where f_{maj} is the majority function, which returns 1 if the majority of the input bits are 1, otherwise returen 0. Use $\{\text{NAND}\}$ as the basis set. Test your program on all strings upto $n = 5$ (however your program should be able to generate f_{maj} for an arbitrary n). You should submit a single python file and may not use any external libraries that make this implimentation trivial. Upload the python file directly on Canvas along with the text/pdf file showing your program output. Determine the entropy loss for computing f_{maj} when $n = 5$. Take $k_B = 1.380649 \times 10^{-23} JK^{-1}$.

Problem 4 (Minsky machines) Design a Minsky machine (see problme 3.1 in the textbook for definition of a Minsky machine) that given two non-negative numbers a and b, computes $a \times b$.