## Homework 8 - Due *Tuesday, November 22, 2022* at 11:59 PM

Reminder Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write "Collaborators: none" if you worked by yourself. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Note You may use various generalizations of the Turing machine model we have seen in class, such as TMs with two-way infinite tapes, stay-put, or multiple tapes. If you choose to use such a generalization, state clearly and precisely what model you are using.

Problems There are 5 required problems.

1. (Individual Review: NO COLLABORATION PERMITTED.) Consider the following computational problem. Given two TMs $M_{1}$ and $M_{2}$, is it the case that the language recognized by $M_{1}$ is exactly the complement of the language recognized by $M_{2}$ ?
(a) Formulate this problem as a language $C_{\mathrm{TM}}$.
(b) Use a reduction to prove that $C_{\mathrm{TM}}$ is undecidable. Explain (briefly) why the TM computing your reduction is correct.
2. (Odd-Length TM) Let $O L_{\mathrm{TM}}=\{\langle M\rangle \mid M$ is a TM that accepts all strings whose length is an odd number and rejects all other strings $\}$.
(a) Your first goal in this problem is to use a mapping reduction from $\overline{A_{\mathrm{TM}}}$ to show that $O L_{\mathrm{TM}}$ is not Turing-recognizable. Describe what the inputs and outputs of this mapping reduction should be.
(b) Describe a TM computing a mapping reduction from $\overline{A_{\mathrm{TM}}}$ to $O L_{\mathrm{TM}}$ and explain why this TM is correct.
(c) Explain why part (b) implies that $O L_{\mathrm{TM}}$ is not Turing-recognizable.
(d) Use a (different) mapping reduction to prove that $\overline{O L_{\mathrm{TM}}}$ is not Turing-recognizable (i.e., $O L_{\mathrm{TM}}$ is not co-Turing-recognizable). Explain briefly why the TM computing your mapping reduction is correct.
3. (Review of Logs and Asymptotic Notation) Use the formal definitions of $O$ and $o$ notation to prove the following statements.
(a) Let $x, y, z$ be variables representing non-negative numbers. Simplify the following expression so it is of the form $a \log _{2} x+b \log _{2} y+c \log _{2} z$, where $a, b, c$ are constants: $\log _{2}\left(\frac{\sqrt{x} \cdot y}{z^{2}}\right)+$ $\log _{4}\left(16^{\log _{2} y}\right)$.
(b) Prove that $n^{2}\left(3 \log _{7} n+n\right)=O\left(n^{3}\right)$ by showing that there exists a constant $c>0$ and a natural number $n_{0}$ such that $n^{2}\left(3 \log _{7} n+n\right) \leq c n^{3}$ for every $n \geq n_{0}$. (Hint: You can use without proof the fact that $\log _{2} n \leq n$ for every $n \geq 1$.)
(c) Prove that $3 n=o\left(n^{2}\right)$ by showing that for every constant $c>0$, there exists a natural number $n_{0}$ such that $3 n \leq c n^{2}$ for every $n \geq n_{0}$.
(d) Prove that $3^{\sqrt{n}}=2^{o(n)}$ by using the fact that $f(n)=o(g(n))$ if $\lim _{n \rightarrow \infty} f(n) / g(n)=0$.
4. (Asymptotic Notation Practice) This problem will be graded automatically by Gradescope. Please enter your answers manually by completing the assignment Homework 8-Problem 4. For each of the following, select true or false using the radio buttons on Gradescope. All logarithms are base 2 unless otherwise stated. (1 point each)
(a) $1337=O\left(n^{2}\right)$
(k) $10^{6}=o(n)$
(b) $n^{10}=O\left(n^{9} \log n\right)$
(1) $2 \log n=o(\log n)$
(c) $n \log n+10 n=o\left(n^{2}\right)$
(m) $\frac{1}{3}=o(1)$
(d) $4^{n}=O\left(2^{n}\right)$
(n) $2^{n}=o\left(4^{n}\right)$
(e) $4^{n}=2^{O(n)}$
(o) $n^{5}=O\left(32^{\log _{2} n}\right)$
(f) $2^{2^{n}}=O\left(2^{n^{2}}\right)$
(p) $\log n=O(\log (\log n))$
(g) $2 n=o\left(n^{2}\right)$
(q) $3^{2^{n}}=o\left(2^{3^{n}}\right)$

## 5. (Polynomial-Time Algorithms)

(a) An undirected graph $G=(V, E)$ is triangle-free if for every triple of vertices $u, v, w$, it is not the case that $(u, v),(v, w)$, and $(w, u)$ are all edges in the graph. Let $T F=\{\langle G\rangle \mid$ $G$ is triangle-free $\}$. Show that $T F \in \mathrm{P}$ by i) giving a high-level description of a polynomialtime algorithm deciding $T F$, ii) analyzing the correctness of your algorithm, and iii) explaining why your algorithm runs in polynomial time.
You don't need to specify the exact polynomial runtime that your algorithm runs in, since this may depend on implementation details that are suppressed in a high-level description. Just give a convincing argument that the runtime is polynomial as in the examples in Chapter 7.2 of Sipser.
(b) The Fibonacci sequence is defined by the following recurrence: $F_{0}=0, F_{1}=1$, and $F_{n}=$ $F_{n-1}+F_{n-2}$ for every $n \geq 2$. Prove that there is no polynomial-time algorithm that takes as input a natural number $n$ (written in binary) and outputs (i.e., writes to its tape) the number $F_{n}$ (again, written in binary).
Hint: How big can the numbers $F_{n}$ get as a function of $n$ ?
(c) Give a high-level description of a polynomial-time algorithm that takes as input a natural number $n$ (written in unary, i.e., as the string $1^{n}$ ) and outputs the number $F_{n}$ (written in binary). Explain why your algorithm is correct and why it runs in polynomial time.
Hint: You can use without proof the fact that Turing machines can perform basic arithmetic operations on binary numbers, like addition, in polynomial time.

