St. Francis Xavier University
Department of Computer Science

CSCI 541: Theory of Computing
Assignment 1
Due October 5, 2023 at 1:30pm

Assignment Regulations.

- This assignment may be completed individually or in a group of two people. If you are collaborating on an assignment as a group, your group must submit exactly one joint set of answers.
- Please include your full name and email address on your submission. For groups, every member must include their full name and email address on the joint submission.
- You may either handwrite or typeset your submission. If your submission is handwritten, please ensure that the handwriting is neat and legible.

[10 marks] 1. A deterministic singly infinite k-tape Turing machine can recognize exactly the same class of languages as an ordinary deterministic Turing machine with one doubly infinite tape. In this question, we will prove this fact in two separate parts.

(a) Prove that a deterministic one-tape Turing machine $\mathcal{M}_1$ is capable of recognizing every language that a deterministic k-tape Turing machine $\mathcal{M}_k$ can recognize. You must give a complete description of exactly how $\mathcal{M}_1$ simulates the computation of $\mathcal{M}_k$, including how the contents of all k tapes are stored on the single tape of $\mathcal{M}_1$.

(b) Prove that a deterministic Turing machine with one singly infinite tape $\mathcal{M}_S$ is capable of recognizing every language that a deterministic Turing machine with one doubly infinite tape $\mathcal{M}_D$ can recognize. You must give a complete description of exactly how $\mathcal{M}_S$ simulates the computation of $\mathcal{M}_D$, including how $\mathcal{M}_S$ handles the situation where the input head of $\mathcal{M}_D$ moves off the leftmost symbol of the tape.

[8 marks] 2. A deterministic k-tape Turing machine without leftward moves has its transition function restricted in the following way: in any computation step, the input head and the work tape heads can either make a rightward move or remain stationary (i.e., make no move). Each tape head can make its move independently of the others. Formally, the transition function is defined to be

$$\delta\colon Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times \{R, N\}^k.$$

(a) Prove that deterministic k-tape Turing machines without leftward moves *cannot* simulate the computation of an ordinary deterministic k-tape Turing machine.

(b) What class of languages does such a model recognize: regular, context-free, decidable, or semidecidable? Justify your answer.

[8 marks] 3. Prove that $\bigcup_{k \geq 0} \mathsf{DSPACE}(n^k) \subseteq \bigcup_{c \geq 1} \mathsf{DTIME}\left(2^{n^c}\right)$.

In your proof, you must show that an arbitrary deterministic Turing machine $\mathcal{M}$ recognizing its language in space $O(n^k)$ can be simulated by some deterministic Turing machine $\mathcal{N}$ in time $O\left(2^{n^c}\right)$.

Hint. Which complexity classes does this question refer to? Consider the number of configurations that $\mathcal{M}$ may find itself in, for some given state set Q and tape alphabet Γ.

[8 marks] 4. Define a function pad: $\Sigma^* \times \mathbb{N} \to \Sigma^*\texttt{\#}^*$ in the following way: $\text{pad}(w, \ell) = w\texttt{\#}^j$, where $j = \max\{0, \ell - m\}$ and $|w| = m$. In other words, the function $\text{pad}(w, \ell)$ pads the word w with enough copies of the symbol # so that the length of the padded word is at least ℓ.

We can extend this padding function from words to languages in the usual way: for any language L and any function $f\colon \mathbb{N} \to \mathbb{N}$, let

$$\text{pad}(L, f) = \{\text{pad}(w, f(m)) \mid w \in L \text{ and } |w| = m\}.$$

Prove that for every language L and for every $k \in \mathbb{N}$, $L \in \mathsf{P}$ if and only if $\text{pad}(L, n^k) \in \mathsf{P}$.

[6 marks] 5. Show that the function n^3 is time constructible. To do this, you must explain how to construct a Turing machine that takes as input the word $\texttt{1}^n$ and writes the word $\texttt{1}^{n^3}$ to its output tape in time $O(n^3)$.