Homework 1

due Tuesday, Jan 11, 2022

Directions: Do any 4 problems.

Turing Machines

From First Principles

Problem 1. For a string $\mathbf{x} \in \{0,1\}^*$, let $\mathsf{int}(\mathbf{x}) \in \mathbb{N}$ denote the integer whose binary representation is $\mathbf{x}$; so if $\mathbf{x} = x_n x_{n-1} \cdots x_1 x_0$, then

$$\mathsf{int}(\mathbf{x}) = x_0 + 2x_1 + 4x_2 + \cdots + 2^{n-1}x_{n-1} + 2^n x_n.$$

Let $f_{\mathsf{add}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be the "addition" function which maps the pair of strings $(\mathbf{x}, \mathbf{y})$ to the string $\mathbf{z}$ which satisfies $\mathsf{int}(\mathbf{z}) = \mathsf{int}(\mathbf{x}) + \mathsf{int}(\mathbf{y})$. Write down the transition function for a Turing Machine which computes f_{add}. Your TM should use the standard alphabet $\{0, 1, \triangleright, \Box\}$; it should have two input tapes ($\mathbf{x}$ will be written on one, $\mathbf{y}$ on the other), in addition to the usual work and output tapes; it should have state set $\{\mathsf{begin}, \mathsf{running}, \mathsf{halt}\}$. Finally, the tapes of your TMs should have their starting cells on the right and continue infinitely off to the left (so the opposite of how we have been drawing them in class). This is for convenience since it will mean x_0 and y_0 are aligned and they are the first you encounter as you move left from the starting position.

Problem 2. Describe a TM which computes the function $f : \{0,1\}^* \to \{0,1\}$ defined by $f(\mathbf{x}) = 1$ if the integer $\mathsf{int}(\mathbf{x})$ is divisible by 3; $f(\mathbf{x}) = 0$ otherwise. Your TM should use the standard alphabet and have the standard three tapes (input/work/output). As in Problem 1, you may assume the tapes of your TM start on the right and continue off infinitely to the left.

Bipartite Graphs

Definition (Graph and Adjacency Matrix). A *graph* G is a pair of finite sets $G = (V, E)$. We call V the set of *vertices* and $E \subset V \times V$ the set of *edges*. The *adjacency matrix* of G, denoted T_G is a $|V| \times |V|$ matrix with 0/1 entries where for any $u, v \in V$, the $(u,v)-$th entry is 1 if and only if $(u,v) \in E$. In this way, the graph $G = (V, E)$ can be represented by a string $T_G \in \{0,1\}^{|V|^2}$.

Definition (Bipartite Graph). We say that a graph $G = (V, E)$ is *bipartite* if V can be partitioned into two sets A and B such that all edges in E connect a vertex in A with a vertex in B. Formally, G is bipartite if there exist $A, B \subsetneq V$ such that $A \cup B = V$ and $A \cap B = \emptyset$ such that for every $(u,v) \in E$ either $u \in A, v \in B$ or $u \in B, v \in A$. Let $f_{\mathsf{bipartite}} : \{0,1\}^* \to \{0,1\}$ be such that $f_{\mathsf{bipartite}}(\mathbf{x}) = 1$ if $\mathbf{x}$ is the adjacency matrixx of a bipartite graph; $f_{\mathsf{bipartite}}(\mathbf{x}) = 0$ otherwise.

Problem 3. Describe part of a TM which computes $f_{\mathsf{bipartite}}$. For simplicity, let us assume your TM is always given input $\mathbf{x} \in \{0,1\}^{n^2}$ for some integer n (so in particular, the input can be parsed as the adjacency matrix of a graph). Thus, your TM is not responsible for rejecting inputs which are not of this form (this would be trivial but tedious). Also, assume your TM has two 2-dimensional work tapes called the "neighbor tape" and the "partition tape". Assume that both 2d tapes have been populated as dictionaries with one key for every vertex in the graph; the values in the neighborhood tape corresponding to the vertex v is a list of all vertices $w \in V$ which are *neighbors* of v (*i.e*, $\{w \in V : (v,w) \in E\}$). Write down a transition function which greedily iterates through the vertices of the graph and attempts to put each vertex into either A or B so that all edges traverse from A to B. Your TM can then output 1 if this process terminates successfully; it can output 0 if not.

Non-Deterministic Turing Machines

Definition (Non-Deterministic Turing Machine). A *non-deterministic Turing Machine* is the same as a regular Turing Machine except that has two different transition functions δ_0 and δ_1. In particular, just like a regular Turing Machine, a non-deterministic Turing Machine has three tapes (input, work, output) and a register which keeps track of state. During each time step, the non-deterministic Turing Machine reads from the tape heads, checks the state and makes its decision on the new state, what to write, and where to move the heads according to one of its transition functions.

Definition (Non-Deterministic Decision). For a language $\mathsf{L} \subset \{0,1\}^*$ and a time function $T : \mathbb{N} \to \mathbb{N}$, we say that a non-deterministic Turing Machine M *decides* L *in time* T if for all $\mathbf{x} \in \{0,1\}^*$, there exists a sequence of bits $b_1, b_2, \dots, b_{T(|\mathbf{x}|)}$ such that:

$$\mathbf{x} \in \mathsf{L} \iff \mathsf{M}^{\mathbf{b}}(\mathbf{x}) = 1 \text{ in at most } T(|x|) \text{ steps},$$

where $\mathbf{b} = (b_1, b_2, \dots, b_{T(|x|)})$ and the shorthand "$\mathsf{M}^{\mathbf{b}}(\mathbf{x}) = 1$" indicates that if M is initialized with $\mathbf{x}$ on its input tape and run so that in time step i, M uses the transition function δ_{b_i}, then M eventually halts with 1 on its output tape.

Problem 4. Let $\mathsf{L} \in \{0,1\}^*$ be a language and $T : \mathbb{N} \to \mathbb{N}$ a time function. Prove that $\mathsf{L} \in \mathsf{NTIME}\big(T(n)\big)$ if and only if there exists a non-deterministic Turing Machine M which decides L in time T. Deduce that $\mathsf{L} \in \mathsf{NP}$ if and only if there exists a polynomial time, non-deterministic Turing Machine which decides L.

A Non-Deterministic Time Hierarchy Theorem

Problem 5. Prove the two claims below to complete the proof of the theorem via the technique of "lazy diagonalization".

Theorem. $\mathsf{NTIME}(n) \subsetneq \mathsf{NTIME}(n^2)$.

Notation. If M is a Turing Machine we let $\mathsf{str}(\mathsf{M}) \in \{0,1\}^*$ be the string representation of M. For a string $\alpha \in \{0,1\}^*$, M_α denotes the Turing Machine described by α, and $\mathsf{int}(\alpha) \in \mathbb{N}$ denotes the integer whose binary representation is α.

Proof. Define a function $\phi : \mathbb{N} \to \mathbb{N}$ recursively: $\phi(1) = 2$ and $\phi(i+1) = 2^{\phi(i)^{1.2}}$ for $i \geq 1$. Let us now define the function $f : \{0,1\}^* \to \{0,1\}$ as follows: $f(\mathsf{str}) = 0$ unless str can be parsed as a pair $(\alpha, \mathbf{x})$ such that $\mathbf{x} = 1^n$ (*i.e.*, the string of n consecutive 1s) for some $n \in \mathbb{N}$ such that $\phi(i) < n \leq \phi(i+1)$, where $i = \mathsf{int}(\alpha)$, and furthermore such that one of the following two points holds:

1. $n < \phi(i+1)$ and there exists $\mathbf{w} \in \{0,1\}^{n^{1.5}}$ such that $\mathsf{M}_\alpha(1^{n+1}, \mathbf{w}) = 1$ in at most $n^{1.5}$ steps;
2. $n = \phi(i+1)$ and for all $\mathbf{w} \in \{0,1\}^{(\phi(i)+1)^{1.1}}$, $\mathsf{M}_\alpha(1^{\phi(i)+1}, \mathbf{w}) \neq 1$ in $(\phi(i)+1)^{1.1}$ steps.

Claim. $f \in \mathsf{NTIME}(n^2)$. (**Hint:** Prove this claim by explicitly writing down an n^2-time NDTM which computes f.)

Claim. $f \notin \mathsf{NTIME}(n)$.

Hint: Suppose M is an $\mathcal{O}(n)-$time NDTM which computes f. Let $\alpha = \mathsf{str}(\mathsf{M})$ be the string representation of M, let $i = \mathsf{int}(\alpha)$, and consider the family of unitary strings

$$\mathcal{F} := \{\mathbf{x} = 1^n : \phi(i) < n \leq \phi(i+1)\}.$$

Show that there must exist some $\mathbf{x} \in \mathcal{F}$ such that $\mathsf{M}(\alpha, \mathbf{x}) \neq f(\alpha, \mathbf{x})$. $\square$