CSE 200 - Computability and Complexity
Homework 1
Due Wednesday, October 18, 11:59pm

Instructions: You may work individually or in a team of 2 people. You may switch teams for different assignments. Please ensure your name(s) and PID(s) are clearly visible on the first page of your submission, and then upload the PDF to Gradescope. If working in a group, submit only one submission per group: one partner uploads the submission through their Gradescope account and then adds the other group member to the Gradescope submission by selecting their name in the "Add Group Members" dialog box. You will need to re-add your group member every time you resubmit a new version of your assignment.

It is highly recommended (though not required) that you type your answers. It is your responsibility to make any handwriting clear and legible for grading. For ease of grading, please start each new problem on a separate page.

Problems:

1. **Halt! Who goes there?** We proved in lecture that HALT is not computable by Turing Machines (TMs), where $\mathsf{HALT}\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ is defined by

$$\mathsf{HALT}(\langle M \rangle, x) = \begin{cases} 1 & \text{if } M(x) \text{ halts} \\ 0 & \text{otherwise} \end{cases},$$

where $\langle M \rangle$ is an encoding of the TM M. By replacing M with other computational models, the function naturally generalizes. Your goal is to prove whether or not the halting problem is computable when M is some other computational model.

In other words, you are to determine if the problem is computable using our usual definition of a k-tape machine, but the *input* to the halting function is an encoding of one of the Turing machine variants below. You may assume that nothing strange happens with the encodings for all variants—e.g., determining if an encoding is valid is computable.

   (a) TMs where each work tape has $\ell \geq 1$ different heads.

   (b) TMs where the output tape is write only, input tape is read only, and only the first n entries of each non-output tape may be used, where n is the length of the input.

2. **Go bananas!** Define the banana function $B\colon \{0,1\}^* \to \{0,1\}$ by

$$B(x) = \begin{cases} 1 & \text{The first 42 bits of } x \text{ spell "banana" in ASCII} \\ 0 & \text{otherwise} \end{cases}.$$

Prove that determining whether a given TM M computes the banana function is not computable. (You may **not** use Rice's theorem if you are familiar with it.)

3. **Bootstrapping complexity theorems** Let $T_1, T_2, g\colon \mathbb{N} \to \mathbb{N}$ be nondecreasing time-constructible[1] functions. Prove the following theorem:[2]

$$\text{If } \mathsf{TIME}(T_1) = \mathsf{TIME}(T_2), \text{ then } \mathsf{TIME}(T_1 \circ g) = \mathsf{TIME}(T_2 \circ g).$$

4. **Extra space isn't always helpful** In this question, you will prove the following surprising theorem: $\mathsf{SPACE}(o(\log\log n)) = \mathsf{SPACE}(O(1))$. That is, if you have less than $\log\log n$ memory, you might as well not use any memory at all. This was originally proved in the paper Hierarchies of memory limited computations (Stearns, Hartmanis and Lewis, 1969). Below is a guided proof, where you will need to prove various claims along the way. You may look at the original paper if that helps you.

   Let M be a 3-tape TM which has a read-only input tape, a write-only[3] output tape, a read-write work tape. By assumption, M uses $S(n) = o(\log\log n)$ work memory. We can make two simplifying assumption about M without loss of generality: M halts on all inputs; M cannot write blank symbols on the work tape. The latter condition implies that at every given point in time, the work tape content is

a_1	a_2	$\ldots$	a_{k-1}	a_k	$\sqcup$	$\sqcup$	$\ldots$

   where $\sqcup$ is a blank symbol, and $a_1, \ldots, a_k$ are non-blank symbols.

   For an input $x \in \{0,1\}^*$ define $s(x)$ to be the number of non-blank symbols on the work tape at the end of the computation of $M(x)$. By the assumption, $s(x) \le S(|x|) = o(\log\log|x|)$ for all inputs x. Define the *work configuration* to be a snapshot[4] of the TM that contains the content of the work tape, the location of the head in the work tape, and the TM state. Notably, this snapshot does not contain the location of the head on the input tape.

   The proof will require us to track all the work configurations obtained while running $M(x)$. To this end, let $W(x)$ denote the set of *all* work configurations obtained while running $M(x)$.

   (a) Prove that $|W(x)| = o(\log n)$ for all $x \in \{0,1\}^n$.

   Given input $x \in \{0,1\}^n$ and coordinate[5] $i \in [n]$, define $W(x,i) \subseteq W(x)$ to be the set of all work configurations of $M(x)$ for which the input tape head is at position i.

   (b) Prove that for any $x \in \{0,1\}^n$, the number of distinct $W(x,i)$ is $o(n)$. That is, $|\{W(x,i) \mid i \in [n]\}| = o(n)$. Note that this is better than the trivial bound of n.

[1] Recall a function $g\colon \mathbb{N} \to \mathbb{N}$ is *time constructible* if $g(n) \ge n$ and there is a TM that on input 1^n outputs $g(n)$ (say, suitably encoded in binary) in time $O(g(n))$.

[2] Here we use "$\circ$" for function composition.

[3] In class, we said that the output tape was read-write. In general, this distinction doesn't matter, but in these small space regimes we don't want the Turing machine to sneak computation onto the output, so we will just stipulate that it is write-only.

[4] See Figure 2.2 of the textbook for the idea of a snapshot. However the snapshot in that diagram is not exactly the type used here since it does not contain the location of the head on the work tape.

[5] Here, and throughout, we use $[n]$ do denote the set $\{1, \ldots, n\}$.

(c) Prove that there exists $n_0 \geq 1$ such that the following holds: for any $n \geq n_0$ and any $x = x_1 \cdots x_n \in \{0,1\}^n$, there exist $i < j$ for which $W(x,i) = W(x,j)$ and $x_i = x_j$.

Let B denote the maximum amount of memory used by any input of length at most n_0. That is, $B = \max\{s(x) \mid x \in \{0,1\}^n, n \leq n_0\}$.

(d) Prove that unless M uses $O(1)$ memory for all inputs of all lengths, there must be input length $n > n_0$ and an input $x \in \{0,1\}^n$ for which $s(x) > B$.

Fix the first (minimal) such n^* and $x \in \{0,1\}^{n^*}$. By (c), there exist coordinates $i < j$ for which $W(x,i) = W(x,j)$ and $x_i = x_j$. Define a new word:

$$y = x_1 \cdots x_i x_{j+1} \cdots x_{n^*} \in \{0,1\}^{n^*-j+i}$$

To conclude the proof, we will show that $M(y)$ never halts, which is a contradiction.

(e) Prove that $W(y,k) \subseteq W(x,k)$ for $k \in [i]$ and $W(y,i+k) \subseteq W(x,j+k)$ for $k \in [n^* - j]$.

Hint: Reason by induction on the number of steps in the computation. The only places where the work configuration of $M(y)$ and $M(x)$ might diverge is in the crossover between x_i and x_{j+1}. Explain why the condition in (c) guarantees that it doesn't.

(f) Prove that $s(y) < s(x)$.

(g) Prove that $M(y)$ never reaches a work configuration where the state is q_{halt}, the halting state. For this, note that when $M(x)$ halts, it used $s(x)$ memory.

(h) Complete the proof.