CSE 200 - Winter 2020
Homework 1
Due Monday, January 27th, 11:59pm

<u>Question 1: Equivalent models of Turing Machines</u>

A RAM Turing Machine has the following tapes:

- Input tape
- Work tape
- RAM address tape

The input tape is read-only. The rest are both read and write. There are two special instructions (i.e. states) that a RAM Turing Machine has, in addition to a standard Turing Machine.

- **Read from memory:** The content the work tape, at the address given in the RAM address tape, is read and written in the current head location on the work tape.
- **Write to memory:** the content of the work tape, at the address given in the address tape, is rewritten to be the content under the current head location on the work tape.

Both operations do not change the location of the head in the work tape.

(a) Show that a standard TM (with a read-only input tape, and a read-write work tape) can simulate a RAM TM.
(b) If the time complexity in the RAM TM is T(n), what is the time complexity of its simulation in the standard TM ? prove your answer.

Hint: Mimic the structure of the proof that the number of tapes doesn't matter.

<u>Question 2: Rice's theorem</u>

Two TMs M1,M2 are called <u>equivalent</u> if they behave the same on all inputs. Namely, for every input x, they both accept, both reject or both loop.

A <u>property of TMs</u> is a function $f : \{0,1\}^* \to \{0,1\}$ that satisfies the following condition: for any equivalent TMs M1 and M2 it holds that f(<M1>)=f(<M2>). Here we assume that <M> is some fixed encoding of TMs.

Prove Rice's theorem: if f is a <u>decidable property</u> of TMs then f is constant. Namely either f(<M>)=0 for all M, or f(<M>)=1 for all M.

Hint: Reduce from the halting problem.

Question 3: Time vs space

We proved in class that $SPACE(S) \subset TIME(n2^{O(S)})$. Prove that this exponential gap is tight.

Hint: give a problem which can be solved in constant space, but that cannot be solved in sub-linear time.

Question 4: below log space

We will prove the following theorem: $SPACE(o(log\ log\ n)) = SPACE(O(1))$. Namely, giving a TM access to memory of size $o(log\ log\ n)$ is not more powerful than giving it access to a constant size memory. In other words, the TM can be simulated by a DFA and hence computes a regular language. 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. Feel free to look at the original paper if you feel that it may help you. Your goal is to give the proofs of claims (a)-(h) given below.

Let M be a TM which has a read-only input tape and a read-write work tape. By assumption, M uses $S(n) = o(log\ log\ n)$ work memory. We assume M halts on all inputs. We make a simplifying assumption: M cannot write blank symbols on the work tape. This implies that at every given point in time, the work tape content is $w_1 w_2 ... w_k bbbb..$ where b is a blank symbol, and $w_1, ..., w_k$ are non-blank symbols.

For an input $x \in \{0,1\}^*$ define $s(x)$ to be the number of non-blank symbols at the end of the computation of $M(x)$. That is, $s(x)$ is the number of work tape cells used when running M on x. By our assumption, $s(x) \leq S(|x|) = o(log\ log\ |x|)$ for all inputs x.

A <u>work configuration</u> is a <u>snapshot</u> of the TM (see page 11 of the textbook), except for the location of the head on the input tape. That is, it includes the content of the work tape, the location of the head in the work tape, and the TM state.

The proof will require us to track all the work configurations obtained while running M(x). Let $W(x)$ denote the <u>set</u> 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 $i \in [n]$, define $W(x,i) \subset W(x)$ to be the set of all the work configurations obtained when running M(x), when the input tape head is at coordinate i. Observe that $w \in W(x,i)$ together with i give a complete snapshot of the TM.

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

(c) Prove that there exists $n_0 \geq 1$ such that the following holds: for any $n \geq n_0$ and any $x = x_1 \ldots x_n \in \{0,1\}^n$, 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 $\leq n_0$. That is, $B = max\{s(x') : x' \in \{0,1\}^{n'}, n' \leq n_0\}$.

(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:

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

To conclude the proof, we will show that M(y) never halts, which is a contradiction, as M should halt on all inputs.

(e) Prove that $W(y,k) \subseteq W(x,k)$ for $k = 1,..,i$ and $W(y,i+k) \subseteq W(x,j+k)$ for $k = 1,..,n-j$.

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 an accept or a reject state. For this, note that when M(x) terminates, it used $s(x)$ memory.

(h) Complete the proof.

Advice: if you get stuck at stages (g)-(h), read the original paper. It is explained well there.