CSE 431 – Theory of Computation. Spring, 2014. Instructor: James R. Lee

ASSIGNMENT 2. Due Thursday, April 24th, in class (or via email to cse431-staff@cs before class starts)

1. A **useless state** in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states. Formulate this problem as a language and show that it is undecidable.

2. Recall the definition of a computable function (Section 5.3 of the book). A language A is **mapping-reducible** to a language B, written $A \leq_m B$ if there is a computable function $f : \Sigma^* \to \Sigma^*$ where for every $w \in \Sigma^*$,

$$w \in A \Leftrightarrow f(w) \in B$$

   The function f is called a **reduction** of A to B. One can easily check that if $A \leq_m B$ and B is decidable, then A is decidable as well.

   a) Consider the language

   $L = \{ \langle M, w \rangle$: M is a TM that on input w ever attempts to move its tape head left when it is on the left-most cell of the tape $\}$.

   Show that L is undecidable by giving a reduction from A_{TM} to L.

   b) Give an example of an undecidable language L where $L \leq_m \overline{L}$.

3. **Uncomputable numbers.** Consider Turing machines with alphabet $\Sigma = \{0,1\}$ and tape alphabet $\Gamma = \{0,1,\blacksquare\}$ (here $\blacksquare$ is meant to be the blank symbol). Define the function $B : \mathbb{N} \to \mathbb{N}$ as follows. For each value k, consider all the k-state TMs that halt when started with a blank input tape. Let $B(k)$ be the maximum number of 1s that remain on the tape among all these machines.

   a) Show that B is **not** a computable function.

   b) Show that B grows faster than any computable function in the following sense: If $f : \mathbb{N} \to \mathbb{N}$ is computable, then for some value k we must have $B(k) \geq f(k)$. [This means that B grows super fast! Note that functions like $f(k) = 2^{2^{2^{2^{2^{\cdots}}}}}$ where there are k 2's in this tower **are** computable. $B(k)$ grows faster than any such function.]

OPTIONAL PROBLEM (You may do this problem for extra credit, OR you can do it instead of the first three problems!)

Let A and B be two disjoint languages. Say that a language C **splits** A and B if $A \subseteq C$ and $B \subseteq \bar{C}$. The language C here acts as a **proof** that A and B are disjoint. Describe two Turing-recognizable languages that cannot be split by any decidable language.