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

ASSIGNMENT 1. Due Thursday, April 17th, in class (or via email to cse431-staff@cs before class starts)

1. The usual Turing machine has a 1-dimensional infinite tape. Suppose we invent a new model: the **2-dimensional Turing machine.** This Turing machine has an entire infinite grid of tape cells (one tape cell for every (x, y) where x and y are non-negative integers). It operates just like a 1D Turing machine, except at each step, the tape head can move up, down, left, or right. If the tape head sits a site $(i, 0)$ and tries to go down it stays in place. If it sits at $(0, i)$ and tries to go left then it stays in place. Show that this type of Turing machine recognizes exactly the class of Turing-recognizable languages.

   [Hint: In problems like this, it is nice to remember the power we already have. We know that a multi-tape TM recognizes exactly the same language as a single-tape TM, so your simulation of the grid TM could use multiple tapes. But it cannot use infinitely many tapes! Find a mapping between the grid and a single 1D tape, and use a second "work tape" to compute the correspondence.]

2. Suppose we have a finite alphabet Σ and a language $\mathcal{L} \subseteq \Sigma^*$. In this case, we define another language

$$\mathcal{L}^* = \{ w_1 w_2 w_3 \dots w_k : w_i \in \mathcal{L} \text{ for } i = 1,2, \dots, k \text{ and some } k \in \mathbb{N}\}$$

   a) Prove that if $\mathcal{L}$ is Turing-recognizable, then so is $\mathcal{L}^*$.
   b) Prove that if $\mathcal{L}$ is Turing-decidable, then so is $\mathcal{L}^*$.

3. Suppose that $\Sigma = \{a, b, c, d, \dots, z\}$ (the usual 26-letter Roman alphabet). Show that a language $\mathcal{L} \subseteq \Sigma^*$ is Turing-decidable if and only if there is an enumerator that enumerates $\mathcal{L}$ in **lexicographic order.**

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

Suppose that we change the Turing machine model so that it cannot alter the input (the input symbols must remain in place), but it can write anywhere else on the tape. Give a complete proof that such a Turing machine cannot even recognize the language of palindromes over the alphabet $\{0,1\}$.