EECS 343: Theoretical Computer Science, Homework Exercise 2
due Monday, February 3, 2020 before class

Problem 1: Suppose you have a new "multi-core" Turing machine. The multi-core TM has k tapes, a separate head for each tape, and a separate state for each head. The transition function for the multi-core TM is

$$\delta : Q^k \times \Gamma^k \to Q^k \times \Gamma^k \times \{L, R, S\}^k$$

Initially, the input is on the first tape, and all the other tapes are blank. You can program the machine two ways. You can choose to have the machine accept once all "cores" enter the accept state, or you can have the machine accept once one of the "cores" enters the accept state. Prove that any language that the multi-core Turing machine decides can also be decided by normal, single tape, single head TM.

Problem 2: (Sipser 3.12) Our Turing machine has started malfunctioning! Now, anytime you give it a "left" move, instead of moving one cell to the left, it jumps all the way back to the first cell of the tape. You want to throw it out, but your boss resists because Turing machines are expensive. Can you redesign your programs so that they work on this broken Turing machine? Specifically, prove that any language that can be recognized by a normal Turing machine can be recognized by this broken machine.

Problem 3: (Sipser 3.15(b)/3.16(b)) Given L_1 and L_2, the language $L_1 \circ L_2 = \{ab \mid a \in L_1, b \in L_2\}$. That is, the language is the set of strings that are *concatenations*, and a string w is in $L_1 \circ L_2$ if there exists some way to split w into a prefix and suffix such that the prefix of w is in L_1 and the suffix is in L_2. Prove that (a) if L_1 and L_2 are Turing-decidable, then $L_1 \circ L_2$ is Turing-decidable, and (b) if L_1 and L_2 are Turing-recognizable then $L_1 \circ L_2$ is Turing-recognizable.