COT 4210 DAILY PF#5 SOLUTIONS SUMMER 2012

1) a) q_1000
$\sqcup q_2 00$
$\sqcup 0 q_3 0$
$\sqcup 0 0 q_4$
$\sqcup 0 \sqcup q_{reject}$

b) $q_1 0000$
$\sqcup q_2 000$
$\sqcup 0 q_3 00$
$\sqcup 0 0 q_4 0$
$\sqcup 0 0 q_3 \sqcup$
$\sqcup 0 0 q_5 0 \sqcup$
$\sqcup 0 q_5 0 0 \sqcup$
$\sqcup q_5 0 0 0 \sqcup$
$q_5 \sqcup 0 0 0 \sqcup$
$\sqcup q_2 0 0 0 \sqcup$
$\sqcup 0 q_2 0 0 \sqcup$
$\sqcup 0 0 q_3 0 \sqcup$
$\sqcup 0 0 0 q_4 \sqcup$
$\sqcup 0 0 0 0 \sqcup q_{accept}$

c) $q_1000000$
$\sqcup q_2 00000$
$\sqcup 0 q_3 0000$
$\sqcup 0 0 q_4 000$
$\sqcup 0 0 0 q_3 00$
$\sqcup 0 0 0 0 q_4 0$
$\sqcup 0 0 0 0 q_3 \sqcup$
$\sqcup 0 0 0 q_5 0 \sqcup$
$\sqcup 0 0 q_5 0 0 \sqcup$
$\sqcup 0 q_5 0 0 0 \sqcup$
$\sqcup q_5 0 0 0 0 \sqcup$
$q_5 \sqcup 0 0 0 0 \sqcup$
$\sqcup q_2 0 0 0 0 \sqcup$
$\sqcup 0 q_2 0 0 0 \sqcup$
$\sqcup 0 0 q_3 0 0 \sqcup$
$\sqcup 0 0 0 q_4 0 \sqcup$
$\sqcup 0 0 0 0 q_4 \sqcup$
$\sqcup 0 0 0 0 \sqcup q_{reject}$

2) The flaw in step 2 is that we commit to running M on s_i until it finishes. The problem is that it may never terminate while running on a particular string, meaning that certain strings in L never get tried. If they never get tried, our proposed enumerator won't ever produce them. Thus, the proposal doesn't satisfy the requirement of an enumerator to print out any string in L. For example, if $1 \in L$ and $0 \notin L$ and M loops on 0, assuming we test 0 before 1, our proposed enumerator will NEVER print 1, even though $1 \in L$.

3) We will prove that the two models are equivalent by showing that a Stay TM can simulate any steps that a regular TM can, and that a regular TM can simulate any steps that a Stay TM executes.

The first direction of the proof is fairly simple. A Stay TM can simulate a regular TM simply by not using its Stay option, since any transition allowed on a regular TM is allowed on a Stay TM.

For the other direction, each move left or right in a Stay TM can be replicated identically on a regular TM. But, we must find a way to simulate the move:

$\delta(q, a) = (r, b, S)$

on a Stay TM on a regular TM. We will replace each move of this type in a regular TM as follows:

$\delta(q, a) = (\text{temp-state}_r, b, R)$
$\delta(\text{temp-state}_r, X) = (r, X, L)$, for each character $X \in \Gamma \cup B$.

Note that the the temporary state is different for each destination state r that has a stay transition in the stay machine. Without this distinction, there's no way of knowing which state to proceed to from the temp-state. Also, note that you need several transitions, for each alphabet character and the blank symbol, because the transitions must handle whatever character appears on the square to the right. Finally, it's important to do the right transition before the left transition instead of the other way around because we're only guaranteed that a square will be available on the right.

Creating this substitution will allow a regular TM to simulate the behavior of a Stay TM so that outcome of all input strings in the Stay machine is preserved in this regular TM that simulates it.

4.) a) Let TM_1 decide L_1 and TM_2 decide L_2. We describe how to build a TM TM_3 that decides $L_1 \cup L_2$. Let TM_3 run the same steps as TM_1. If this accepts, accept. Otherwise, run the input on the same directions as TM_2. If this accepts, accept. Otherwise, reject. For specifics, describe TM_3 as a multitape machine that first copies its input onto its second tape before running TM_1's steps. After finishing those steps, if necessary, copy the contents from the second tape to the first. Assume these details for parts (b), ~~(c)~~ and (d).

b) Keep everything the same as part (a) except if the steps of TM_1 reject, reject. If they accept, continue and ~~the~~ run the steps of TM_2 on the input. Accept if this accepts. Reject otherwise.

c) Let TM_1 decide L. We describe how to build TM_2 that decides $\overline{L}$. Run the steps of TM_1. If they accept, reject. Otherwise, accept.

d) Make TM_3 a multi-tape machine to decide $L_1 \circ L_2$. Use one tape as a counter that starts at 0 and ends at n, where n is the length of the input. Use another tape to store a copy of the original input.