Vladimir Hugec

Question 1:

Input: 01#01

$q_1 01\#01$
$xq_2 1\#01$
$x1q_2\#01$
$x1\#q_4 01$
$x1q_6\#x1$
$xq_7 1\#x1$
$q_7 x1\#x1$
$xq_1 1\#x1$
$xxq_3\#x1$
$xx\#q_5 x1$
$xx\#xq_5 1$
$xx\#q_6 xx$
$xxq_6\#xx$
$xq_7 x\#xx$
$xxq_1\#xx$
$xx\#q_8 xx$
$xx\#xq_8 x$
$xx\#xxq_8$
$xx\#xxq_{accept}$

Input: 00#0
$q_1 00\#0$
$xq_2 0\#0$
$x0q_2\#0$
$x0\#q_4 0$
$x0q_6\#x$
$xq_7 0\#x$
$q_7 x0\#x$
$xq_1 0\#x$
$xxq_2\#x$
$xx\#q_4 x$
$xx\#xq_4$
$xx\#xq_{reject}$

Question 2:

The extra movements for a Turing machine: stay, double left, and double right, are no more powerful than a Turing machine with only the options to move left and right since those extra movements are in reality just extensions of the original left and right movements. For example, the new stay option can be simulated by a left movement followed by a right movement or a right movement by a left movement depending on implementation, and double left and double right are simply two consecutive left and right movements. So in reality, the new Turing machine with the extra movement options is no more powerful than the stock Turing machine. In order to simulate these extra movements, we simply need to add another transition state that move the tape head the extra movement necessary. For the transition $\delta(q_{start}, x) \rightarrow \delta(q_2, y, S)$ we add the intermediary q_i to simulate the transition by moving the head at first to the right or left, then in the opposite direction of the first one used. I will use right. So the transition now becomes $\delta(q_{start}, x) \rightarrow \delta(q_i, y, R) \rightarrow \delta(q_2, x, L)$. The exact same procedure is done for double left and double right. Therefore these extra movements are no more powerful than just using the normal ones since they can be easily replicated by the normal ones.

Question 3:

Since we know that for any DFA D, we can construct another that accepts the complement language and for any DFA's D_1 and D_2, we can construct others that accept the union and intersection of those languages. To show, $L(A) \subset L(B)$ is decidable, we can construct a TM M that decides $L(A) \subset L(B)$. Using the first fact above, since for $L(A)$ to be a proper subset of $L(B)$, the intersection of $L(A)$ and $L(B)$ should contain all of $L(A)$ but only a part of $L(B)$. Therefore if we have a string w, that accepts in $L(A) \cap L(B)$ then it must have been in $L(A)$ and vice-versa.

$M =$ "On input $\langle A, w \rangle$, where A is a $L(A) \cap L(B)$ and w is a string."

1. Simulate A on input w
2. If A accepts then accept, if A rejects then reject.