CS 254: Computability and Complexity	**Anonymous submission**
Problem Set #08	*November 7, 2019*

1. Consider the lang XOR = $\{M, x, y|$M is a TM that halts on precisely one of the strings x or y$\}$

   Intuition: We can't determine if one of them is going to loop forever or will eventually halt.

   Proof by mapping reduction:

   Show that co-A$_{TM}$ reduces to XOR. Assume for the sake of contradiction that TM R decides XOR. Then construct a TM S that uses R to decide co-A$_{TM}$

   S = "on input M, x, y:

   1. Simulate M on X

      if M accepts or rejects x, have M loop forever on y

      if M accepts or rejects y, have M loop forever on x

      else: reject
   2. Run R on M, x, y to determine if it accepts one and loops on the other.
   3. if R accepts, M rejects, otherwise M accepts.

   We know this properly maps because either x or y will halt and the other will loop forever. Therefore we we have built a recognizer for co-A$_{TM}$ which we know is not r.e. This is a contradiction, thus XOR is not r.e.

   Now we can prove that co-A$_{TM}$ maps to co-XOR.

   Proof by mapping reduction:

   Show that co-A$_{TM}$ reduces to co-XOR. Assume for the sake of contradiction that TM R decides co-XOR. Then construct a TM S that uses R to recognize co-A$_{TM}$

   S = "on input M, x, y:

   1. Simulate M on x

      if M accepts or rejects x, and accepts or rejects y, accept

      else: loop
   2. Run R on M, x, y to determine if it halts on both or loops on both.
   3. if R accepts, M accepts,

   We know this properly maps because either x and y will halt or both will loop forever. Therefore we have built a recognizer for co-A$_{TM}$ which we know is not r.e. This is a contradiction, thus co-XOR is not r.e.

   Thus we have proven that XOR is neither r.e. or co-r.e.