Tucker Bickler 8/9/11
ttb265 CS 341
Rich

HW 14 – Due Wednesday, August 11, 2011

1.
   a. Not SD
   b. In D. For any given input, xy can be defined. Thus, once the TM is fed input it should accept anything given to it. Similar to H.
   c. In SD. Using a two state TM (affecting the structure) we can keep track of whether or not the length of a string is even. However, if the string is infinitely long, it may not be decidable.
   d. Not SD
   e.

2. Prove that TM_{REG} is not in SD.

   By reduction of H:
   R(<M,w>) =
   a. Construct the description <M#> of M#(x);
      i. Erase the tape
      ii. Write w on the tape.
      iii. Run M on w.
   b. Return <M#>.

   If Oracle exists and decides TM_{reg}, then C = Oracle(R(<M,w>)) decides H. R can be implemented as a Turing machine. And C is correct. M# ignores its own input. It halts on everything or nothing. So:

      i. <M,w> $\in$ H: M halts on w, so M# halts on all inputs. Oracle accepts.
      ii. <M,w> $\notin$ H: M does not halt on w, so M# halts on nothing. Oracle rejects.

3.