ComS 331 Spring 2024 Name: Aren Ashlock

1. Are the following languages Turing-decidable? Turing-acceptable but not Turing-decidable? Not even Turing-acceptable? For each answer, give an explanation of your reasoning (just as in class, M is a generic deterministic Turing machine, w a generic input string to it, and ρ is an encoding function).

   - $L_1 = \{\rho(M) : |L(M)| > 10\}$
   - $L_2 = \{\rho(M) : |L(M)| \leq 10\}$
   - $L_3 = \{\rho(M)\rho(w) : M \searrow w \text{ in 10 steps or less}\}$
   - $L_4 = \{\rho(M)\rho(w) : M \searrow w \text{ in more than 10 steps}\}$

   L_1 is **Turing-decidable** because we can build a TM that keeps track of each string encountered (to prevent double counting) and the # of unique strings in $L(M)$. By doing so, if (1) the TM halts and the tracker is ≤ 10, it says "N" or (2) the tracker reaches 11 (> 10), the TM halts and says "Y."

   L_2 is **Turing-decidable** for a similar reason to L_1. We can build a TM that keeps track of each string encountered (to prevent double counting) and the # of unique strings in $L(M)$. By doing so, if (1) the TM halts and the tracker is ≤ 10, it says "Y" or (2) the tracker reaches 11 (> 10), the TM halts and says "N."

   L_3 is **Turing-acceptable but not Turing-decidable**.
   First, it is not Turing-decidable because we can reduce the Halting problem to it ($M_H \to M_T$). Assume the convergence in 10 steps or less is decidable, that would mean the Halting problem is decidable. Use $\rho(M)\rho(w)$ as input to M_T, which would say "Y" if it halts in 10 steps or less and "N" otherwise. However, "N" encompasses the cases where M_T may not halt, meaning it cannot be decidable.
   However, L_3 is Turing-acceptable and we can come up with a TM that behaves accordingly based on the outcomes of M. (1) If $M \searrow w$, we only want the TM to halt if it is done in 10 steps or less. (1a) So, if the # of steps is more than 10 steps, we simply have the TM go to an infinite loop and never halt. (1b) Otherwise, if if it is done in 10 steps or less, the TM will continue to halt. (2) If $M \nearrow w$, the TM continues to not halt. Therefore, the TM is capable of halting when $M \searrow w$ in 10 steps of less and not halting otherwise.

   L_4 is **Not even Turing-acceptable**. If it were Turing-acceptable, then it would have to halt for all w where M halts in more than 10 steps. Since there is no upper bound to when M should halt, as long as the TM that runs L_4 isn't halting, there is still uncertainty as to whether $\rho(M)\rho(w) \in L_4$, so the TM can't know whether to halt (and accept it) or if it'll be rejected. We know about the halting problem, which explains how it's not possible to know if a TM will halt for every w, so it's impossible to have the TM halt for every $\rho(M)\rho(w)$ where $M \searrow w$ in more than 10 steps. Thus, it is not even Turing-accptable.