R.E. stands for "recursively enumerable", but in the more modern nomenclature, they sometimes use "computably enumerable" for the same concept.

CSE 200
Computability and Complexity
Homework 2
Models of Computation, Efficient Computation, and Undecidability
Due Tuesday, January 26

January 14, 2016

1. Consider the *Dyck language* D_2, with two types of nested parenthesis, $(,),[,]$ (which you can think of as two types of begin statements, such as in latex, with corresponding two types of end statement.) A word w is in D_2 if there is a way of mapping open parenthesis to closed parenthesis of the same type, so that no two edges in the matching cross. (That is, if a loop begins inside another loop, it must also end within that loop, and vice versa. So $([()])([]())$ is in the language, but $[[(])]$ is not.) Prove that this language is decideable in time $O(n^2)$ on a one-tape Turing Machine. Prove a matching lower bound ($\Omega(n^2)$) for all one-tape TMs deciding this language .

2. It would be nice to have a programming language PL where: A) (Recognizability) we could computably tell whether a string was a valid PL program ; B) (Termination guarantee) given a valid PL program and an input, we could simulate the program on the input computably, thus guaranteeing that all programs in our language halt; and C) (Generality) for every recursive language L, there is a program in PL that computes membership in L.

   Show that no programming language exists having all three properties.

3. Give an example of a language L that is neither R.E. nor co-R.E.