

HW8 – DUE 10 AM 25.Feb.2013

Instructions:

Submit **HW8_[lastname].lisp** by dropping it into your Assignments folder on Google Drive. Please document your code.

1. We can represent a binary tree in LISP as follows:

- nil will represent the empty tree; otherwise
- Each node of a binary tree non-empty tree will be of the form (atom list1 list2), where atom represents the nodes value and list1 and list2 are lists (possibly nil) representing the left and right subtrees respectively; for example

(a (b nil (c nil nil)) (d nil (e (f nil nil) nil)))

- a. Define a function treep that accepts a list as an argument and returns T if the list is a tree and NIL otherwise.
- b. Define a function preorder that accepts a tree as a parameter and returns a list containing the node values of the tree based on a preorder traversal. When applied to the above tree, preorder would return the list (a b c d e f)
- c. Same as b. except give an inorder traversal. When applied to the above tree, inorder would return the list (b c a d f e)
- d. Same as b. and c. except give a postorder traversal. When applied to the above tree, preorder would return the list (c b f e d a)

2. Extend the definition of the symbolic differentiation function d we developed to include differences between functions and the quotient rule.