

# Homework 6

meghan  
doughtin

1. substitute(x, y, L1, L2):

(lists are empty base)

substitute(x, y, [], []).

substitute(x, y, [x|L1], [y|L2]) ←

substitute(x, y, L1, L2).

substitute(x, y, [z|L1], [z|L2]) ←

substitute(x, y, L1, L2), x ≠ z.

\* substitute(a, x, [a, b, a, c], [x, b, x, c]) ; x, b, x, c is  
result of sub  
x for all a in  
a, b, a, c

sub(a, x, [a|b, a, c], [x|b, x, c])  
sub(a, x, [b|a, c], [b|x, c])  
sub(a, x, [a|c], [x|c])  
sub(a, x, [c], [c])  
sub(a, x, [], [])

checking

2. no-doubles(L1, L2)

\* member prev. defined  
in text book

no-doubles([], []).

\* nonmember prev. defined  
in text book

no-doubles([x|L1], L2) ←

member(x, L1), no-doubles(L1, L2).

no-doubles([x|L1], [x|L2]) ←

nonmember(x, L1), no-doubles(L1, L2).

3. sum-tree(TreeOfIntegers, sum)

\* tree prev. defined  
in text book

sum-tree(void, 0).

sum-tree(tree(x, L, R), sum) ←

sum-tree(L, sumL), sum-tree(R, sumR),

plus(x, sumL, s), plus(s, sumR, sum).

4. path(x, Tree, Path)

path(x, void, Path) :

path(x, Tree, Path) ←

edge(x, Node), path(Node, Tree, Path) . .