**Homework 7 (SML)**

**Programming Principles and Implementation**

**Yevgeniy Shatrovskiy**

**Homework on SML**

What are the types of the following expressions?

* [(1,5), (2,3), (5,6)];
  + (int \* int) list
* real -> bool
  + real -> bool
* fun f(x:real) = true;
  + real -> bool
* map f;
  + a list -> b list

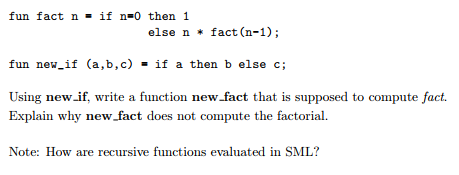
1. Provide expressions of the following types:

* int \* bool
  + (1,true)
* int list \* bool
  + ([1,2,3,4], true)
* int \* real -> bool list
  + fun yes((x:int, y:real)) = [false];

1. Write the following SML functions:



fun power(0) = 1 | power(n) = 2 \* power(n-1);



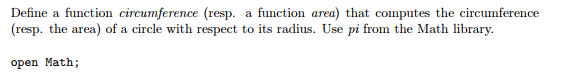
After completing the function, the code does not throw an error, but it also does not complete. I tried to use a print statement but it also did not return any sort of print command. This is the modified functions that I used:

fun new\_if(a,b,c) = if a then b

else( print("Here"); c);

fun new\_fact(n) = new\_if(n = 0, 1, n \* new\_fact(n-1));

The only explanation that I can think of is that it is looping over and over again. I believe that it may be calling the inner function instead of evaluating the base case condition.



open Math;

fun circumference(r) = Math.sqrt(Math.pi \* Math.pow(r,2.0) / Math.pi) \* Math.pi \* 2.0;

circumference (2.0);

How to use map to add 3 to each elements of a list

map (fn x => x+3) [1,2,3,4,5];

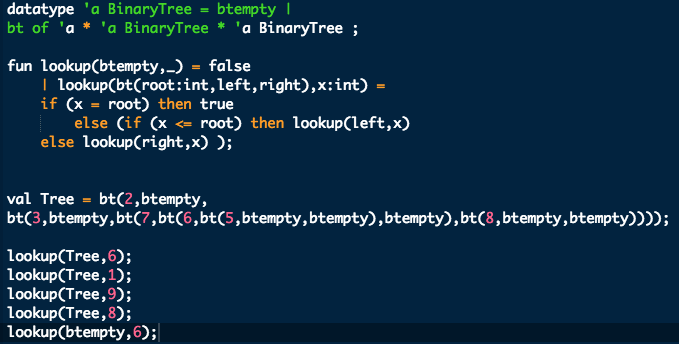
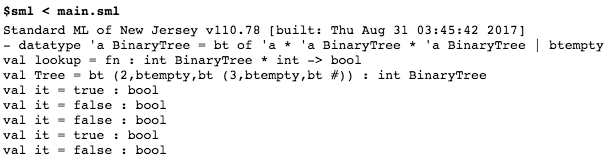
You can also predefine a function beforehand in order to apply it to every element in the array.

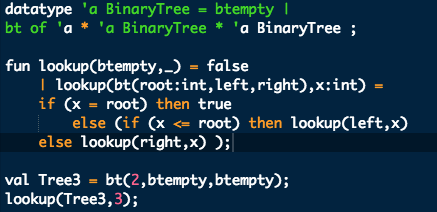


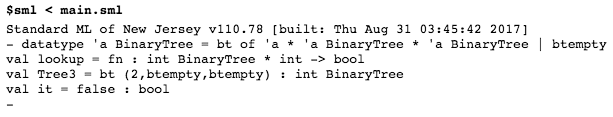
fun move(l) = tl(l) @ [hd(l)];

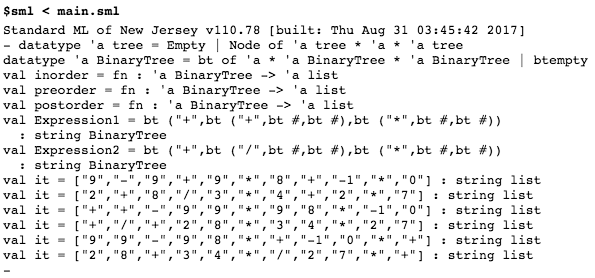
1. Implement the datatype BinaryTree and all the functions that are provided in the lecture notes: lookup, inorder, preorder, postorde, left\_subtree, right\_subtree and label. Provide screenshots to show that your code is correct. Provide 2 tests for each function.

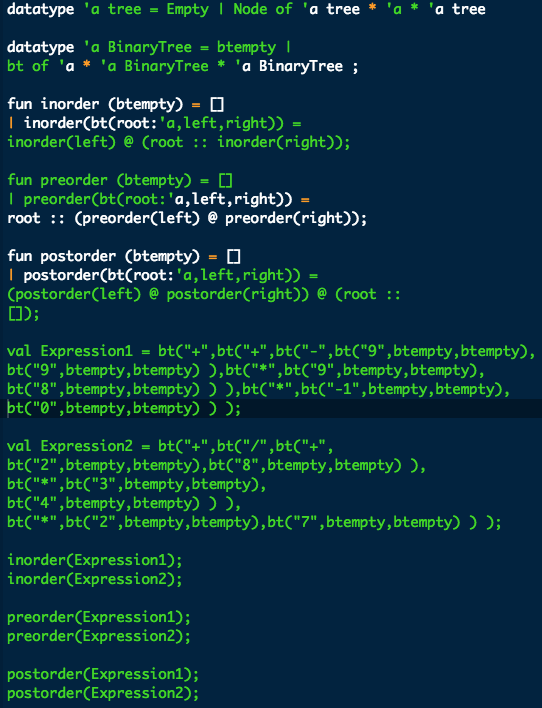
Lookup(Tree and Tree3):

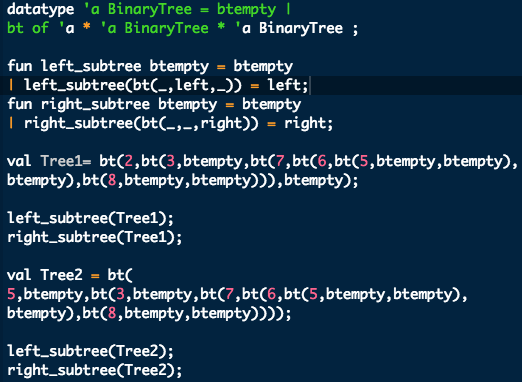
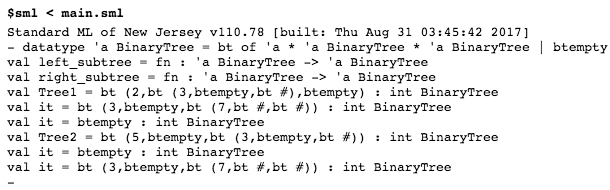


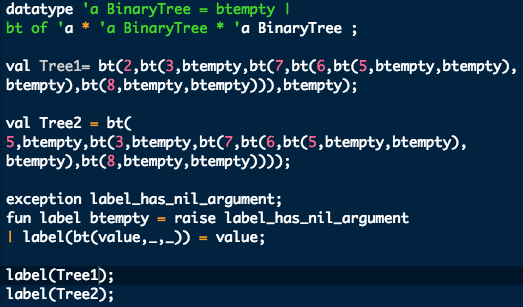
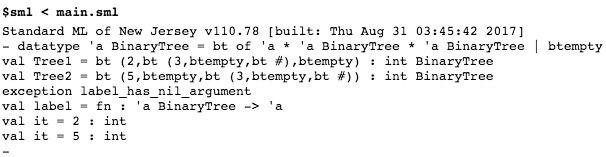




In-order, Pre-order, Post-order (Expression1 and Expression2):



Left\_subtree, Right\_subtree (Tree1 and Tree2):

Label(Tree1 and Tree2):