**Matthew Sokolovsky**

**Homework on SML and PROLOG**

To do alone or in a team of 2 students.

**SML**

1. What are the types of the following expressions?

* [(1,5), (2,3), (5,6)];

int \* int list

* fun f(x:real) = true;

real -> bool

* map f;

real list -> bool list (assuming that it’s using the f function above)

1. Provide expressions of the following types:

* int \* bool

(127,false)

* int list \* bool

([127,255,32767,65535],true)

* int \* real -> bool list

fun oog(n:int,x:real) = if n = 0 then []

else if x > 0.0 then true::oog(n-1,x)

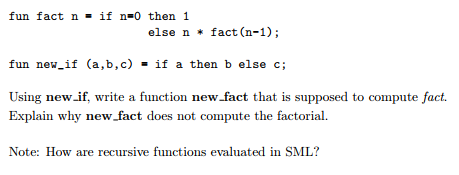
else false::oog(n-1,x);

1. Write the following SML functions:



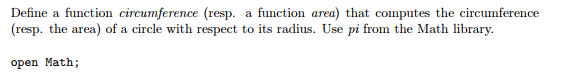
fun powtwo(0) = 1

| powtwo(n:int) = 2 \* powtwo(n-1);



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

I tried running this and it seems that the new\_fact function causes an infinite loop. My guess is that SML uses innermost evaluation, which means that it first evaluates all the arguments of new\_if – calling new\_fact in the process – *before* it calls new\_if. As a result, it just keeps calling new\_fact endlessly without ever checking if n=0.



fun area(r:real) = r \* r \* pi;

fun circumference(r) = sqrt(area(r) / pi) \* pi \* 2.0;

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

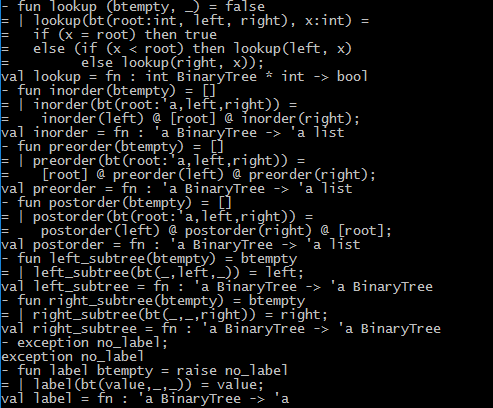
fun add3(x:int) = x + 3;

val add3list = map add3;



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

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



(Ignore the line; it continues on the next page)

