Homework Assignment #3

Due Thursday 10/30

1. In homework #1, we saw a representation of binary trees as lists and a corresponding `sum_tree` function. Why can't you write this function in ML?

2. Write a `reduce_tree` function for the tree datatype on slide 75. Now write a `sum_tree` function in terms of `reduce_tree`.

3. Write a function `compose_list` that takes a list of functions and returns a function that is their composition. For example:

```
compose_list [] [1,2,3] = [1,2,3]
compose_list [tl,tl,tl] [1,2,3,4,5] = [4,5]
compose_list [(fn x=>x+1), (fn x=>x+2)] 4 = 7
```

4. Simulate the type inference of your `reduce_tree` and `compose_list` functions, as was done in class. Show the initial types of each argument and the result, and then accumulate constraints on these types, unifying types until you satisfy all of the constraints.

5. Write a signature and a corresponding structure which implements a symbol table. The symbol table maintains a mapping from identifiers (i.e. strings) to data. The symbol table should support the operations `lookup`, which finds an identifier in the symbol table, returning the associated data, and `insert`, which adds an identifier and its associated data to the symbol table. The `lookup` function should raise an exception when the given identifier is not found.