

**Moti Begna**  
**CSCI 5106**  
**Homework 7**

## Problem 2

- Type for the *insert* function =  $\text{fn: } ('a * 'a \rightarrow \text{bool}) * ('a * 'a \rightarrow \text{bool}) * 'a * 'a \text{ tree} \rightarrow 'a \text{ tree}$
- Type for the *member* function =  $\text{fn: } ('a * 'b \rightarrow \text{bool}) * ('a * 'b \rightarrow \text{bool}) * 'a * 'b \text{ tree} \rightarrow \text{bool}$

The *insert* and *member* function defined in problem 1 ensure at compile time that whatever arbitrary value types are being parametrized in the function, they stay consistent with the other parameters that are being passed into the function. For example, in the *insert* function, the compiler understands that whatever type 'a ends up being, it will remain consistent with the other parameter values to the function. Namely, the equality relation will take 2 values of type 'a as well as the ordering relation, the element inserted is of type 'a, and the parametrized binary search tree as well as the tree being returned are of type 'a tree. Thus, if the type for 'a is set as an int at runtime, all of the parametrized values will treat 'a as such. By doing this, we ensure that at compile-time that type errors do not arise at run-time.

Now, this implementation of polymorphism in sml differs from c in a substantial way. Primarily, the differences arise in the simplicity that sml offers in achieving polymorphism with the any type declaration 'a. While c may have to rely on the use of a union or by explicitly declaring parameterized values to have a void\* type, sml bypasses these restrictions by making use of inductively defined types. One similarity, however, is that it is up to the programmer to ensure that the *insert* and *member* functions are receiving the correct types when called after their definition. If for example, a string was trying to be added to an integer tree, the compiler would issue a tycon mismatch where the operator domain expected 'a to be defined as an integer type for all of its parameters, but was given an operand where one of the parameters (namely the inserted value) was of type string.