# Homework 8

This homework is due Thursday, May 3.

For programming problems, hand in a print-out and mail a single file containing all the modules to `coleman` on snowhite.

1. (10 points) Define a module STRING-WITH-ERROR with sort `String` in OBJ3 that models the semantic algebra of Schmidt's example 3.3. (Note that there is an error element.) You may use the module CHAR found in

   `~leavens/cs541public/obj/char.obj`

   as a definition of character strings. Is it possible to use subsorts to dispense with the error string?

2. (10 points) Define a parameterized module PRODUCT with sort `Product` in OBJ3 that models the product denotational semantics (see Schmidt's 3.2.1). Can you generalize your module to work for a finite product $A_1 \times \cdots \times A_n$?

3. (15 points) Define a parameterized module DISJOINT-UNION in OBJ3 that is like the disjoint union domain described in Schmidt's 3.2.2.

   (a) What problems do you encounter in trying to make the domain match Schmidt's definition? Separate out the syntactic problems (e.g., names of operations) from semantic ones (e.g., OBJ3 won't let you do something you need to do).

   (b) Which problems go away if you try to model a specific instance of such a module (that is a particular disjoint union such as "Int + Error")?

4. (15 points) Compare the relative merits of inheritance (as in Smalltalk) with importing modules (as in OBJ3).

5. (20 points) Compare the relative merits of subtype polymorphism (the polymorphism achieved by message passing) with the generic modules and subsorting of OBJ3.

6. (15 points) Based on your experience with Scheme and functional programming, argue against Goguen's assertion that higher-order functions are unnecessary for higher-order programming (see section 4.6 of "Introducing OBJ3").