HW 17 **Prof. Caldwell**
Due: 11 November 2008 **COSC 3015**

In class we started to present the unification algorithm. I have provided an implementation and some tests (in comments at the end.) Your assignment is to extend the code by adding a product type constructor to the datatype `Type` and then to extend the rest of the functions to handle this new type. Note that this include all functions that do pattern matching on the type constructors for datatype `Type`.

Recall:

Definition 0.1. Type terms t_1 and t_2 are *unifiable* if there is a substitution σ such that $\sigma t_1 = \sigma t_2$.

Thus, a unifier is a substitution that makes the terms equal.

The algorithm is specified as follows:

i.) A variable (say x) unifies with any term (say t) as long as $x \notin FV(t)$.

ii.) If terms t_1 and t_2 are not variables, they unify if they have the same constructor, and if their subterms unify.