HW 10 **Prof. Caldwell**
Due: 5 October 2008 **COSC 3015**

1 Modelling Finite Functions as Lists

Exercise 1.1. Based on the code provided on the HW web-page you must write the following functions.

```
update :: Eq a => (a,b) -> FinFun a b -> FinFun a b
functional :: Eq a => [(a,b)] -> Bool
domain :: (Eq a, Eq b) => FinFun a b -> [a]
range :: (Eq a, Eq b) => FinFun a b -> [b]
apply :: (Eq a, Eq b) => FinFun a b -> a -> Maybe b
```

1.1 `update :: Eq a => (a,b) -> FinFun a b -> FinFun a b`

This function should be called as `update(i,v) f` where `f` is a finite function with domain `a` and codomain `b`. If `i` is the first element of any pair in `f` return the finite function that is just like `f` except that it maps `i` to `v`. If `i` is not the first element of any pair in `f`, return the finite function the behaves just like `f` but also contains the pair `(i,v)`.

1.2 `functional :: Eq a => [(a,b)] -> Bool`

Call this predicate as `functional m` where `m` is a list of pairs. The predicate returns `True` iff for every pair `(i,v)`$\in$ `m`, there is no pair `(j,v')` $\in$ `m` with `i=j`.

1.3 `domain :: (Eq a, Eq b) => FinFun a b -> [a]`

This function returns a list of values in the domain `a` that the function is actually defined for.

1.4 `range :: (Eq a, Eq b) => FinFun a b -> [b]`

This function returns a list of values in the codomain `b` that is the range of the finite function.

1.5 `apply :: (Eq a, Eq b) => FinFun a b -> a -> Maybe b`

`apply f x` :: apply the finite function `f` to argument `x` and return `Nothing` if `x` $\not\in$ `(domain f)` and return `Just y` when the pair `(x,y)` $\in$ `f`.

Exercise 1.2. Instantiate the type of finite functions `FinFun` as an instance of the `Eq` type class by defining `==` to be the extensional equality on functions. You can use the equality in the module `Sets` as a model, or you may even use it in your definition.

1.6

Recall that extensional equality for finite functions means the list representing the function is functional *and* they have the same elements.