Brandon Cao

CIS 425

Spring 16’

Assignment 5

* 1. val a = fn : int \* int -> int

The two parameters x,y go through an arithmetic operation by adding x to 2y, which would return an integer.

* 1. val b = fn : real \* real -> real

x and y must be real numbers because both of them added together are being divided by a real number, 2.0.

* 1. val c = fn : (‘a -> ‘b) -> ‘a -> ‘b

uncurried function

* 1. val d = fn : ( ( ‘a -> ‘a ) \* ‘a ) -> ‘a )

\*This is assuming that there is no error because in the book there is an extra parenthesis.

d takes in a function and a parameter (x) and the function being passed in recursively calls itself passing x into the function. X in this case can be a function since it is recursively calling itself or a value depending on what the function is.

* 1. val e = fn : 'a \* 'a \* ('a -> bool) -> 'a

Because there is a condition, there will be an arrow with ‘a pointing to a Boolean. The function is also of type y because it is a function of y. y is also one of the possible return values of function e and the returned values must be of the same type. Thus that is why function e returns of type ‘a.

2)

Sort function is of type (‘a \* ‘a -> bool) \* ‘a list -> ‘a list. Insert function is of type ‘a \* ‘a list -> ‘a list.Sort takes in two arguments (less and list). This makes insert have a list type because it is not bound to any other type. A programmer would expect this because they can pass a list and a comparison operator to sort any type of list.

3) (‘a -> int) \* ’a -> int

4) There would be an error for the output because the parse tree has a circular loop and cannot make the g’s the same type.

5) The append function takes in two lists which should be of the same type. The function must return a list of type ‘a list since the operation is to combine the two lists. Knowing the function is of this type, a programmer can easily find a bug since the function works with lists containing items of the same type.

6) ML type checker states that the function should return one value of type ‘a. The types do not match so that is why an error occurs.

The type checker states that the function returns a list, but the function should return a single value of type ‘a, not a list. The type of function is wrong (x should not be a list) and this points you to an error.

7)

a) f: ‘a -> bool

b) g:a’->int

c) These functions have the same run-time behavior because they are both recursive.

d) The designers didn’t work harder to make this possible because when running the two functions it’s hard to determine the differences because they never terminate.

8)

see attached file: typechecker.sml