Zack Austin

CS 4450 HW 8

10/1/13

Homework Chapter 8: 1,3,5

1. Overloaded Operator + with types: int*real → real, int * int → int, real*int → real, and real * real → real.

a) i + r: int * real → **real**.

b) i + r + i: (int*real → real) + i:

r + i: real * int → **real**.

c) i + (r + i): i + (real * int → real):

i + r: int * real → **real**.

d) i + i + r + (r + i): int * int → int + r + (real * int → real):

int * real → real + r:

r + r = real * real → **real**.

3. Language with int and real types where there are legal expressions of 1 + 2, 1.0 + 2, 1 + 2.0, and 1.0 + 2.0.

a) Result of just coercion: In all cases we will make sure the types are implicitly converted to the type of argument our non-overloaded operator uses. If + is used as a real * real → real, then 1 + 2 becomes 1.0 + 2.0, 1.0 + 2 becomes 1.0 + 2.0, etc.

b) Result of just overloading: The + operator is overloaded so it can either a int*int, a real*int, an int*real, or real*real to perform the operation. We overload the operator for every combination for the types involved.

c) Result of both coercion and overloading: Many ways to do this but we could first implicitly convert the types used to the same type and then use that version of the overload on that kind of type. For example: 1 + 2.0 are differing types so we implicitly convert 1 to 1.0 and then use the overloaded operator + as a real*real → real on 1.0 + 2.0.

d) Result of subtype polymorphism: We could treat int as a subtype of real, so any function that takes a parameter of real could of also taken a parameter of type int.

5. ML Function Definitions for following functions:

Before Checking:

a) f(x) = 1: 'a → int

b) f(x,y) = 1: 'a * 'b → int

c) f(x) = x: 'a → 'a

d) f(x,y) = x: 'a * 'b → 'a

e) f(g) = g(1): (int → 'a) → 'a

f) f(g,x) = g(x): ('a → 'b) * 'a → 'b

g) f(g,x,y) = g(x,y): ('a * 'b → 'c) * 'a * 'b → 'c

h) f(g,h,x) = g(h(x)): ('a → 'b) * ('b → 'c) * 'a → 'c. **Wrong.**

i) f(g,x) = g(g(x)): ('a → 'b) * 'b → 'b. **Wrong.**

After:

a) f(x) = 1: 'a → int

b) f(x,y) = 1: 'a * 'b → int

c) f(x) = x: 'a → 'a

d) f(x,y) = x: 'a * 'b → 'a

e) f(g) = g(1): (int → 'a) → 'a

f) f(g,x) = g(x): ('a → 'b) * 'a → 'b

g) f(g,x,y) = g(x,y): ('a * 'b → 'c) * 'a * 'b → 'c

h) f(g,h,x) = g(h(x)): ('a → 'b) * ('c → 'a) * 'c → 'b

i) f(g,x) = g(g(x)): ('a → 'a) * 'a → 'a