

# Functional Programming (Sheet #1)

Marius Gavrilescu

1. 

```
areaCircle :: Double -> Double
areaCircle r = pi * r^2
areaCone :: (Double, Double) -> Double
areaCone (r,h) = pi * r * (r + sqrt(r^2 + h^2))
```
2. Three:
  - (a) `square 7 = 49`
  - (b)  $(3+4) * (3+4) = (3+4) * 7 = 7 * 7 = 49$
  - (c)  $(3+4) * (3+4) = 7 * (3+4) = 7 * 7 = 49$
3. `multiply(0, infinity)` will return 0, while `multiply(infinity, 0)` will not end.
4. A function `f` is strict if  $f(\perp) = \perp$ .  
`g` is strict, so  $g(\perp) = \perp$ . `f` is strict, so  $f(g(\perp)) = f(\perp) = \perp$ . Since  $h(x) = f(g(x))$ , we get  $h(\perp) = \perp$ . Therefore `h` is strict.
5. (a) `((integral f) a) b * ((deriv g) x)`  
(b) `((f 3)*2)*4`
6. 

```
fib :: Integer -> Integer
fib 0 = 0
fib 1 = 1
fib x = fib (x-1) + fib(x-2)
```
7. 

```
is_leap :: Int -> Bool
is_leap x
| x `rem` 4 > 0 = False
| x `rem` 100 > 0 = True
| x `rem` 400 > 0 = False
| otherwise = True
```
8. (a) No. Counterexample: it cannot be applied to the function `mult (a,b) = a * b`. More generally, it can only be applied to a function of the type `(a -> a)`.  
(b) `twice twice twice f` will apply `f` 16 times. More generally,  $\underbrace{\text{twice twice} \dots \text{twice}}_{n \text{ times}} f$  applies `f`  $2 \uparrow n$  times.