

Programming/Theory HW6

CS 169 S19, Linnell

Due Fri. 5/17 in class

Submission instructions at the end

1. (5 points) In Scala, write a more general version of the `sum_fun` function called `apply_combine` that applies a function `f` (taken as a parameter) to the integers from 1 to `x` (taken as a parameter), then combines those values using another function `g` (taken as a parameter). So `combine` takes in three parameters: a function `f` that takes in an `Int` and returns an `Int`, an `Int x`, and a function `g` that takes in two `Ints` and returns an `Int`. It then returns the result of using the function `g` to combine the values gained by applying `f` to 1, 2, ..., `x`.

For example, if I define

```
def add(x:Int, y:Int) :Int = x+y
def square(x:Int):Int = x*x
```

then

```
combine(square, 4, add)
would return 30
```

And if I define

```
def mult(x:Int, y:Int) :Int = x*y
```

then

```
combine(x=>x, 4, mult)
would return 24
```

2. (10 points) Write a curried version of the function from problem 1. This function, `apply_combine2`, takes in one parameter, the function `f: Int=>Int`. `apply_combine2` then returns a function that takes in one `Int` parameter, and returns yet another function which in turn takes in a function `g: (Int, Int)=>Int` and returns the same `Int` value as `apply_combine`. It's up to you whether you wish to use named or anonymous functions.

For example, we could use this function in the following ways:

```
def apply_square = apply_combine2(x=>x*x)
def apply_square_5 = apply_square(5)
println(apply_combine2(x=>x*x) (5) ((x, y)=>x*y))
```

Hint: When figuring out your types, start by figuring out the (overall) type of *the function that takes in `g`* (it's NOT `(Int, Int)=>Int`) and use that to figure out the return type of the function that takes in the `Int` parameter.

3. Theory (5 points) a. Reduce the following expression in the λ -Calculus:

$((\lambda m. \lambda n. \lambda f. \lambda x. ((m f) ((n f) x)) \lambda f. \lambda x. (f x)) \lambda f. \lambda x. (f (f x)))$

Note: there are several bound `x`s and `f`s with different scopes. You may wish to rewrite some of the expressions with different variable names.

b. What arithmetic operation do you think is represented in the λ -Calculus by $\lambda m. \lambda n. \lambda f. \lambda x. ((m\ f)\ ((n\ f)\ x))$? Wait until after class Monday to try to answer part b. **Hint:** Given our discussion on Monday, how would you interpret the two parameters sent in to this function in part a?

See the syllabus for instructions on how to start a Scala project in Eclipse.

Submission instructions: You will print out your code for each problem, stapling together multiple sheets (there will be deductions for unstapled homework!). Turn in the hardcopy at the beginning of class. You will ALSO submit all of your .scala files as attachments, to cs169@math.scu.edu (NOT Dr. Linnell's email!!) The subject line of the email should be "CS169 HW6 YourLastName YourIDNumber "