CSc 4330 Programming Language Concepts
Exam 2, Spring 2018, SOLUTIONS
4 April 2018

Problem 1 (10 Points)

Show the result of applying the following substitutions:

```
  (λy.((λf.(f x)) y)) [x := (f y)]
= (λy'.((λf'.(f' (f y))) y'))

  ((f (λx.(x y)))(λz.((x y) z))) [x := g]
= ((f (λx.(x y)))(λz.((g y) z)))

  (λx.(λy.((f x) y))) [y := x]
= (λx'.(λy.((f x') y)))

  ((λx.(f x))(λf.(f x))) [f := (g x)]
= ((λx'.((g x) x'))(λf.(f x)))

  (λf.(λy.((f x) y))) [x := (f y)]
= (λf'.(λy'.((f' (f y)) y')))
```

Problem 2 (40 Points)

Reduce the following expressions as much as possible:

```
  ((λx.(x y))(λz.z))

= (x y) [x := (λz.z)]
= ((λz.z) y)

= z [z := y]
= y
```

```
  ((λx.((λy.(x y)) x))(λz.w))

= ((λy.(x y)) x) [x := (λz.w)]
= ((λy.((λz.w) y)) (λz.w))

= ((λz.w) y) [y := (λz.w)]
= ((λz.w)(λz.w))

= w [z := (λz.w)]
= w
```

```
  (( (λz.(λy.((z y) z))) y) a)

= ( (λy.((z y) z))[z := y] a)
= ( (λy'.((z y') z))[z := y] a)
= ( (λy'.((y y') y)) a)

= ((y y') y) [y' := a]
= ((y a) y)
```

```
  (((λx.(λy.(y x)))(λz.(x z)))(λy.(y y)))

= ((λy.(y x))[x := (λz.(x z))] (λy.(y y)))
= ((λy.(y (λz.(x z)))) (λy.(y y)))

= (y (λz.(x z))) [y := (λy.(y y))]
= ((λy.(y y)) (λz.(x z)))

= (y y) [y := (λz.(x z))]
= ((λz.(x z)) (λz.(x z)))

= (x z) [z := (λz.(x z))]
= (x (λz.(x z)))
```

Problem 3 (20 Points)

Consider the following encodings of Boolean values in λ-Calculus:

```
true = λa.λb.a
false = λa.λb.b
```

and the encoding of “not” and “or” as

```
not = λp.λa.λb.p b a
or = λp.λq.p p q
```

Prove that

```
(a)  (not true) = false

  ((λp.λa.λb.(p b a))(λa.λb.a))
= λa.λb.((λa.λb.a) b a)
= λa.λb.((λb’.b) a)
= λa.λb.b
= false

(b)  (or true false) = true

  ((λp.λq.p p q)(λa.λb.a)(λa.λb.b))
= (( λq.((λa.λb.a)(λa.λb.a) q))(λa.λb.b))
= ((λa.λb.a)(λa.λb.a)(λa.λb.b))
= (((λb.(λa.λb.a))(λa.λb.b))
= λa.λb.a
= true
```

Problem 4 (10 Points)

Consider the following definitions from HW 4c:

```
type Set = Int => Boolean
val bound = 1000
```

Write a Scala function to compute the sum of all values in a set; For example

```
sumElements(add(add(add(emptySet,10),20),30)) should return a value of 60.
```

```
def sumElements(s: Set): Int = {
  def helper(i: Int, sum: Int): Int =
    if (i > bound) sum
    else if (s(i)) helper(i+1,sum+i)
    else helper(i+1,sum)
  helper(-bound,0)
}
```

Problem 5 (10 Points)

Write a Scala function to decode a run-length encoded list. For example

```
decode(List(("a",3),("b",4),("c",2)))
```

evaluates to

```
List("a","a","a","b","b","b","b","c","c")
```

```
def decode(xs: List[(String,Int)]): List[String] =
   xs flatMap (x => List.fill(x._2)(x._1))
```

Problem 6 (10 Points)

Write a Scala function that takes as input a list of integers `xs` and a positive integer `k` and returns a new list obtained from `xs` by removing every `k`th element. For example,

```
dropKth(List(1,2,3,4,5,6,7,8,9,10), 3)
```

should evaluate to

```
List(1,2,4,5,7,8,10)
```

```
def dropKth2(xs: List[Int], k: Int): List[Int] = xs match {
  case Nil => Nil
  case y::ys => xs.take(k-1)++dropKth2(xs.drop(k),k)
}
```

```
def dropKth(xs: List[Int], k: Int): List[Int] = {
   def dkth(ys: List[Int], n: Int): List[Int] = ys match {
     case Nil => Nil
     case z::zs => if (n > 1) z::dkth(zs,n-1) else dkth(zs,k)
   }
   dkth(xs,k)
}

def dropKth3(xs: List[Int], k: Int): List[Int] =
  for (i <- 0 until xs.length if (i+1)%3!=0) yield xs(i)
```

Problem 7 (10 Points)

Consider the following Scala function:

```
def f[T](xs: List[T]): List[T] =
  if (xs.isEmpty)
    xs
  else
    xs.head :: f(for(x <- xs.tail if (x != xs.head)) yield x)
```

(a) What is the value of the following expression:

```
f(List(2, 12, 6, 2, 9, 6, 2))

List(2, 12, 6, 9)
```

(b) What does the function `f` do?

Removes duplicates from xs and returns the new list

Problem 8 (10 Points)

Consider the following Scala function:

```
def g(n: Int): List[Int] =
  for (i <- List.range(1,n+1) if n%i == 0) yield i
```

(a) What are the values of the following expressions:

```
g(7) = List(1,7)

g(12) = List(1,2,3,4,6,12)
```

(b) What does the function `g` do?

Computes all the divisors of n and returns them in a list.

Problem 9 (10 Points)

What are the values of the following Scala expressions:

```
List(1,7,2,9).reduceLeft(_-_)

     -17  (((1-7)-2)-9)

List(1,7,2,9).foldLeft(0)(_-_)

     -19  ((((0-1)-7)-2)-9)

List(1,7,2,9).reduceRight(_-_)

     -13  (1-(7-(2-9)))

List(1,7,2,9).foldRight(0)(_-_)

     -13  (1-(7-(2-(9-0))))
```

Problem 10 (5 Points)

Complete the following definition of finding the maximum value of a list of integers using only `foldLeft`:

```
def maxList(xs: List[Int]): Int =
  xs.foldLeft(Int.MinValue)(scala.math.max(_,_))
```

Problem 11 (5 Points)

Complete the definition of factorial using `foldLeft` and `1 to n` (no recursion/loop)

```
def factorial(n: Int): Int =
  (1 to n).foldLeft(1)(_*_)
```
