Sotheanith Sok

CECS 424

12/1/18

Assignment-11

1.
```
myfoldr :: (a -> b -> b) -> b -> [a] -> b
myfoldr f acc [ ] = acc
myfoldr f acc (x : xs) = f x (myfoldr f acc xs)

mylengthr :: [a] -> Int
mylengthr = myfoldr (\ _n -> 1 + n) 0

mylengthr [1, 2, 3]
mylengthr     = myfoldr (\ _ n -> 1 + n) 0 [1, 2, 3]
              = (\ _ n -> 1 + n) 1 (myfoldr (\ _ n -> 1 + n) 0 [2, 3])
              = (\ _ n -> 1 + n) 1 ( (\ _ n -> 1 + n) 2 (myfoldr (\ _ n -> 1 + n) 0 [3]) )
              = (\ _ n -> 1 + n) 1 ( (\ _ n -> 1 + n) 2 ( (\ _ n-> 1 + n) 3 (myfoldr (\ _ n -> 1 + n) 0 [ ]) ) )
              = (\ _ n -> 1 + n) 1 ( (\ _ n -> 1 + n) 2 ( (\ _ n-> 1 + n) 3 0) )
              = (\ _ n -> 1 + n) 1 ( (\ _ n -> 1 + n) 2 1)
              = (\ _ n -> 1 + n) 1 2
              = 3
```

2. myfoldl :: (a -> b -> a) -> a -> [b] -> a

myfoldl f acc [ ] = acc

myfoldl f acc (x:xs) = myfoldl f (f acc x) xs

a. mylengthl :: [a] -> Int

mylengthl = myfoldl (\ n _-> 1 + n) 0

b. mylengthl [1,2,3]

mylengthl = myfoldl (\ n _ -> 1 + n) 0 [1, 2, 3]

= myfoldl (\ n _ -> 1 + n) ((\ _ n -> 1 + n) 0 1) [2, 3]

= myfoldl (\ n _ -> 1 + n) ((\ n _ -> 1 + n) ((\ _ n -> 1 + n) 0 1) 2) [3]

= myfoldl (\ n _ -> 1 + n) ((\ n _ -> 1 + n) ((\ n _ -> 1 + n) ((\ n _ -> 1 + n) 0 1) 2) 3) [ ]

= ((\ n _ -> 1 + n) ((\ n _ -> 1 + n) ((\ _ n -> 1 + n) 0 1) 2) 3)

= ((\ n _ -> 1 + n) ((\ n _ -> 1 + n) 1 2) 3)

= ((\ n _ -> 1 + n) 2 3)

= 3

3. myfoldl :: (a -> b -> a) -> a -> [b] -> a

myfoldl f acc [ ] = acc

myfoldl f acc (x:xs) = myfoldl f (f acc x) xs

a. myreverse :: [a] -> [b]

myreverse :: myfoldl (\ m n -> n : m) [ ]

b. myreverse [1,2,3]

myreverse = myfoldl (\ m n -> n : m) [ ] [1,2,3]

= myfoldl (\ m n -> n : m) ((\ m n -> n : m) [ ] 1) [2,3]

= myfoldl (\ m n -> n : m) ((\ m n -> n : m) ((\ m n -> n : m) [ ] 1) 2) [3]

= myfoldl (\ m n -> n : m) ((\ m n -> n : m) ((\ m n -> n : m) ((\ m n -> n : m) [ ] 1) 2) 3) [ ]

= ((\ m n -> n : m) ((\ m n -> n : m) ((\ m n -> n : m) [ ] 1) 2) 3)

= ((\ m n -> n : m) ((\ m n -> n : m) (1 : [ ]) 2) 3)

= ((\ m n -> n : m) (2 : (1 : [ ])) 3)

= (3 : (2 : (1 : [ ])))

= [3, 2, 1]