

CMSC330 Spring 2015 Quiz #3

Name _____

Discussion Time (circle one): 10am 11am 12pm 1pm 2pm 3pm

Discussion TA (circle one): Amelia Casey Chris Mike Elizabeth Eric Tommy

Instructions

- Do not start this test until you are told to do so!
- You have 15 minutes for this quiz.
- This is a closed book exam. No notes or other aids are allowed.
- Answer essay questions concisely in 2-3 sentences. Longer answers are not needed.
- For partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

1. (6 pts) Lambda Calculus

Evaluate the following λ -expressions as much as possible. Show each alpha conversion and/or beta-reduction.

Recall that application is left-associative, i.e., $x y z$ is equivalent to $(x y) z$

a.) (1 pt) $(\lambda f. \lambda x. f) a$
 $\Rightarrow (\lambda x. a)$

b.) (1 pt) $(\lambda x. x) (\lambda x. x) b$
 $\Rightarrow (\lambda x. x) b$
 $\Rightarrow b$

c.) (2 pts) $(\lambda f. \lambda x. f (f x)) (\lambda u. z) a b$
 $\Rightarrow (\lambda x. (\lambda u. z) ((\lambda u. z) x)) a b$
 $\Rightarrow (\lambda x. (\lambda u. z) z) a b$
 $\Rightarrow (\lambda x. z) a b$
 $\Rightarrow z b$

d.) (2 pts) $(\lambda f. \lambda y. \lambda x. x (y f)) y x f$
 $\Rightarrow (\lambda a \lambda b. \lambda c. c (b a)) y x f$ (* alpha conversion *)
 $\Rightarrow (\lambda b. \lambda c. c (b y)) x f$
 $\Rightarrow (\lambda c. c (x y)) f$
 $\Rightarrow f (x y)$

2. (9 pts) Consider the OCaml type definition *myTree*:

```
type myTree =  
  Nil  
  | Leaf of int  
  | Node of myTree * myTree
```

A value of the type *myTree* is made up of Nil elements; Leaf elements, which have an associated integer value; and Node elements, which have two *myTree* children. Here are some example *myTree* values

```
Node(Leaf 1, Leaf 2)  
Node(Node(Leaf 1,Leaf 2), Leaf 3)  
Node(Leaf 1, Node(Leaf 2, Node(Leaf 3, Nil)))
```

Write a function called *switch* that swaps the children of each Node in a *myTree*. E.g.:

```
switch Node(Leaf 1, Leaf 2)           => Node(Leaf 2, Leaf 1)  
switch Node(Leaf 1, Node(Leaf 2, Nil)) => Node(Node(Nil, Leaf 2), Leaf 1)  
switch (Leaf 1)                       => Leaf 1
```

Your code must work in linear time (i.e. don't cycle through a *myTree* multiple times). You are not allowed to use any OCaml library functions. You may use helper functions.

```
let rec switch n = match n with  
  Nil -> Nil  
  | Leaf v -> Leaf v  
  | Node (left,right) -> Node (switch right,switch left)  
;;
```