

# Recent advances in software engineering 32039

## Laboratory exercises: Week 5

**Question 1** Write these out without any  $\lambda$ .

1. `elim` =  $\lambda x.x \lambda y \rightarrow y$
2. `equal` =  $\lambda x.$   
     $x \rightarrow \text{true}$   
    |  $\lambda y.\text{false}$
3. `elim*` =  $\lambda x.$   
     $x \lambda y \rightarrow \text{elim* } y$   
    |  $\lambda y \rightarrow y$
4. `update-c` =  $\lambda x \rightarrow \lambda f \rightarrow$   
     $x \lambda y \rightarrow x (f y)$   
    |  $\lambda y \lambda z \rightarrow (\text{update-c } x f y) (\text{update-cx } f z)$   
    |  $\lambda y \rightarrow y$

**Question 2** Evaluate

1. `elim Leaf (Leaf 3)`.
2. `elim Leaf (Node 3)`.
3. `elim singleton (singleton 3)` (as defined in Lecture 5)
4. `equal Nil Nil`
5. `equal (Cons 3 Nil) (Cons 3 Nil)` (assume that 3 is a constructor)
6. `equal Nil (Cons 3 Nil)`
7. `equal ( $\lambda x.x$ ) ( $\lambda x.x$ )`
8. `elim* Leaf (Leaf 4)`
9. `elim* Leaf (Pair (Leaf 4) (Leaf 5))`
10. `elim* Leaf (Leaf (Leaf 4))`

11. `update-c Leaf ( $\lambda x.x + 1$ ) (Pair (Leaf 4) (Leaf 5))`
12. `update-c Dept (update-c Leaf ( $\lambda x.x+1$ )) (Pair (Dept (Leaf 4)) (Leaf 5))`

**Question 2** 3

Define the *free matchables* and *free variables* of a term  $t$ .

**Question 4** Use `update-c` to define a function that will update the salaries of employees in departments in divisions, where

- a salary is of the form `Salary  $x$`  where  $x$  is a floating point number.
- an employee is of the form `Employee  $n$   $s$`  where  $n$  is a string (the name) and  $s$  is the salary.
- a department is of the form `Department  $s$   $es$`  where  $s$  is its name and  $es$  is a list of employees.
- a division is of the form `Division  $ds$`  where  $ds$  is a list of departments.

Provide an example.

**Question 5** Solve Question 4 using `apply2all` instead of `update-c`. Encode your answer in `bondi`. Does this mean that `apply2all` can do anything that `update-c` can?