

## Fixed Points

### Instructions:

**Solutions of the exercises are to be delivered before Thursday, the 19th of April at 10:15AM.**  
Solutions should be placed in a separate folder with the name “**Assignment06**”.  
Please submit answers to all the exercises in **one** text file.

### Exercise 1 (3 points)

We represent non-negative integers with the following Lambda expressions:

$$\begin{aligned}0 &\equiv \lambda f . \lambda x . x \\1 &\equiv \lambda f . \lambda x . f x \\2 &\equiv \lambda f . \lambda x . f(f x) \\&\vdots \\n &\equiv \lambda f . \lambda x . f^n x\end{aligned}$$

Suppose you have defined the function **if** and the operations **add**, **pred** and **isZero**. Consider the following recursive (and hence not valid) definition for the multiplication:

$$\mathbf{times} = \lambda n_1 . \lambda n_2 . \mathbf{if} (\mathbf{isZero} \ n_1) \ \mathbf{0} \ (\mathbf{add} \ n_2 \ (\mathbf{times} \ (\mathbf{pred} \ n_1) \ n_2))$$

If we abstract the name **times**, we get the new expression:

$$\mathbf{t} = \lambda f . \lambda n_1 . \lambda n_2 . \mathbf{if} (\mathbf{isZero} \ n_1) \ \mathbf{0} \ (\mathbf{add} \ n_2 \ (f \ (\mathbf{pred} \ n_1) \ n_2))$$

By the FP theorem we know that  $(\mathbf{Y} \ \mathbf{t})$  is a non-recursive equivalent of the above **times** definition.

The exercise: write down the reduction sequence to demonstrate that

$$(((\mathbf{Y} \ \mathbf{t}) \ \mathbf{1}) \ \mathbf{k}) \rightarrow \mathbf{k}.$$

### Exercise 2 (3 points)

We can represent lists and list operators with the following Lambda expressions:

$$\begin{aligned}\mathbf{nil} &= \lambda f . \mathit{true} \\ \mathbf{null} &= \lambda l . l \ (\lambda h . \lambda t . \mathit{false}) \\ \mathbf{cons} &= \lambda h . \lambda t . \lambda f . \mathit{fht} \\ \mathbf{head} &= \lambda l . l \ (\lambda h . \lambda t . h) \\ \mathbf{tail} &= \lambda l . l \ (\lambda h . \lambda t . t)\end{aligned}$$

Example: the list [1, 2, 3] is represented by the  $\lambda$ -expression **cons 1 (cons 2 (cons 3 nil))**.

To do:

1. Translate the following definition into a non-recursive form:

$$\mathbf{append} = \lambda l_1 . \lambda l_2 . \mathbf{if} (\mathbf{null} l_1) l_2 (\mathbf{cons} (\mathbf{head} l_1) (\mathbf{append} (\mathbf{tail} l_1) l_2))$$

2. Test your result by appending list  $L_2$  to list  $L_1$ , which are defined below:

$$L_1 = \mathbf{cons} \mathbf{1} (\mathbf{cons} \mathbf{2} \mathbf{nil}) \text{ and } L_2 = \mathbf{cons} \mathbf{3} \mathbf{nil}$$