Solution 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:

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

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

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

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

The exercise: write down the reduction sequence to demonstrate that

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

Answer:

$t \equiv \lambda f.\ \lambda n_1.\ \lambda n_2.\ if\ (\ iszero\ n_1)\ 0\ (add\ n_2\ (\ f\ (pred\ n_1)\ n_2))$

$((\ Y\ t\ )\ 1\ )\ k \equiv$

/* Fixpoint Theorem tells us that Y t = t (Y t) */

$(t\ (\ Y\ t\ )\ 1\ )\ k \equiv$

$(\lambda f.\ \lambda n_1.\ \lambda n_2.\ if\ (\ isZero\ n_1\ )\ 0\ (\ add\ n_2\ (\ f\ (\ pred\ n_1\ )\ n_2\ ))(\ Y\ t\ )\ 1)\ k \equiv$

/* $f = Y\ t.\ n_1 = 1.\ n_2 = k$ */

$if\ (\ isZero\ 1)\ 0\ (add\ k\ ((\ Y\ t\ )(\ pred\ 1\ )\ k\ )) \equiv$

/* isZero 1 = false */

$if\ false\ 0\ (\ add\ k\ ((\ Y\ t\ )(\ pred\ 1\ )\ k)) \equiv$

$add\ k\ ((\ Y\ t\ )(\ pred\ 1\ )\ k\ ) \equiv$

/* pred 1 = 0 */

$add\ k\ ((\ Y\ t\ )\ 0\ k\ ) \equiv$

/* Fixpoint Theorem tells us that Y t = t (Y t) */

$add\ k\ ((\ t\ (\ Y\ t\ ))\ 0\ k\ ) \equiv$

$add\ k\ ((\lambda f.\ \lambda n_1.\ \lambda n_2.\ if\ (\ isZero\ n_1)\ 0\ (\ add\ n_2\ (\ f\ (\ pred\ n_1\ )\ n_2\ ))\ (\ Y\ t\ ))\ 0\ k\ ) \equiv$

/* $f = Y\ t.\ n_1 = 0.\ n_2 = k$ */

$add\ k\ (\ if\ (\ isZero\ 0)\ 0\ (\ add\ k\ (\ (\ Y\ t\ )\ (\ pred\ 0\ )\ k\ ))) \equiv$

/* isZero 0 = true */

$add\ k\ (\ if\ true\ 0\ (\ add\ k\ (\ (\ Y\ t\ )\ (\ pred\ 0\ )\ k\ ))) \equiv$

$add\ k\ 0 \equiv$

k

Exercise 2 (3 points)

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

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

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

To do:

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

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

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

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

<u>**Answer:**</u>

Non-recursive form of append:
app $\equiv \lambda$ *f.* λ l_1. λ l_2. *if (null* l_1 *)* l_2 *(cons (head* l_1 *) (f (tail* l_1 *)* l_2 *))*

Test:

(Y app) L_1 L_2 $\equiv$
    / Fixpoint theorem applied */*
app (Y app) L_1 L_2 $\equiv$
(λ *f.* λ l_1. λ l_2. *if (null* l_1 *)* l_2 *(cons (head* l_1 *)(f (tail* l_1 *)* l_2 *))) (Y app)* L_1 L_2 $\equiv$
    / f = Y app.* $l_1 = L_1$. $l_2 = L_2$. **/*
if (null L_1 *)* L_2 *(cons (head* L_1 *)((Y app)(tail* L_1 *)* L_2 *))* $\equiv$
    / null* L_1 *= false */*
cons (head L_1 *)((Y app)(tail* L_1 *)* L_2 *)* $\equiv$
    / head* L_1 *= 1. tail* L_1 *= cons 2 nil */*
cons 1 ((Y app)(cons 2 nil) L_2*)* $\equiv$
    / Fixpoint theorem applied */*
cons 1 (app (Y app)(cons 2 nil) L_2*)* $\equiv$
cons 1 [λ *f.* λ l_1. λ l_2. *if (null* l_1 *)* l_2 *(cons (head* l_1 *f (tail* l_1 *)* l_2 *))] (Y app) (cons 2 nil)* L_2*)* $\equiv$
    / f = Y app.* l_1 *= cons 2 nil.* $l_2 = L_2$. **/*
cons 1 (if (null (cons 2 nil)) L_2 *(cons (head (cons 2 nil))((Y app)(tail (cons 2 nil))* L_2*))* $\equiv$
    / null cons 2 nil = false. head cons 2 nil = 2. tail cons 2 nil = nil */*
cons 1 (cons 2 ((Y app) nil L_2*))* $\equiv$
    / Fixpoint theorem applied */*
cons 1 (cons 2 (app (Y app)(nil) L_2*))* $\equiv$
cons 1 (cons 2 (λ *f.* λ l_1. λ l_2. *if (null* l_1 *)* l_2 *(cons (head* l_1 *)(f (tail* l_1 *)* l_2 *)) (Y app) nil* L_2 *))* $\equiv$
    / f = Y app.* l_1 *= nil.* $l_2 = L_2$. **/*
cons 1 (cons 2 (if (null nil) L_2 *(cons (head nil)((Y app)(tail nil)* L_2*))))* $\equiv$
    / null nil = true */*
cons 1 (cons 2 L_2 *))* $\equiv$
    */** L_2 *= cons 3 nil */*
cons 1 (cons 2 (cons 3 nil)) $\equiv$ *[1, 2, 3]*