Prof. O. Nierstrasz
Boris Spasojević, Jan Kurš

Serie 6 - Fixed Points

Exercise 1

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

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

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:

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

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

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

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

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

Exercise 2

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:

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