3 Computation Theory (tgg22)

There are many ways to represents lists in the λ-calculus. You will explore one of them here. Let $\underline{n}$ be a λ-term representing the integer n. For this question, integer lists are represented as

$$\begin{aligned} \underline{[]} &= \lambda x\ f.\ x \\ \underline{[n_1,\ n_2,\ \ldots,\ n_m]} &= \lambda x\ f.\ f\ \underline{n_1}(f\ \underline{n_2}\ \ldots\ (f\ \underline{n_m}\ x)\ \ldots) \end{aligned}$$

(*a*) Present, with justification, a λ-term $\mathbf{H}$ such that we have

$$\mathbf{H}\ \underline{[n_1,\ n_2,\ \ldots, n_m]} =_\beta \underline{n_1}.$$

[2 marks]

(*b*) Present, with justification, a λ-term $\mathbf{L}$ such that for any list l we have

$$\underline{\text{length}\ l} =_\beta \mathbf{L}\ \underline{l}.$$

[4 marks]

(*c*) Suppose a function g is represented with the λ-term $\mathbf{G}$. Present, with justification, a λ-term $\mathbf{M}$ such that for any list l we have

$$\underline{\text{map}\ g\ l} =_\beta \mathbf{M}\ \mathbf{G}\ \underline{l}.$$

[4 marks]

(*d*) Consider this Ocaml code for reversing a list:

```
let rec aux r l =
    if l = []
    then r
    else aux ((hd l) :: r) (tl l)

let rev = aux []
```

In answering the following questions you may use these λ-terms and facts.

$$\begin{array}{rcl|rcl} \mathbf{Y}\ F & =_\beta & F\ (\mathbf{Y}\ F) & \mathbf{IF}\ \mathbf{TRUE}\ F\ S & =_\beta & F \\ \mathbf{N}\ \underline{[]} & =_\beta & \mathbf{TRUE} & \mathbf{IF}\ \mathbf{FALSE}\ F\ S & =_\beta & S \\ \mathbf{N}\ \underline{[n_1,\ \ldots,\ n_m]} & =_\beta & \mathbf{FALSE} & \mathbf{C}\ \underline{n}\ \underline{l} & =_\beta & \underline{n :: l} \end{array}$$

(*i*) Define λ-terms $\mathbf{A}$ representing `aux` and $\mathbf{R}$ representing `rev`. [4 marks]

(*ii*) Prove that $\mathbf{R}\ \underline{l}$ correctly implements list reversal for every list l.

[6 marks]