3.0

CS-350 Quiz-2 190549

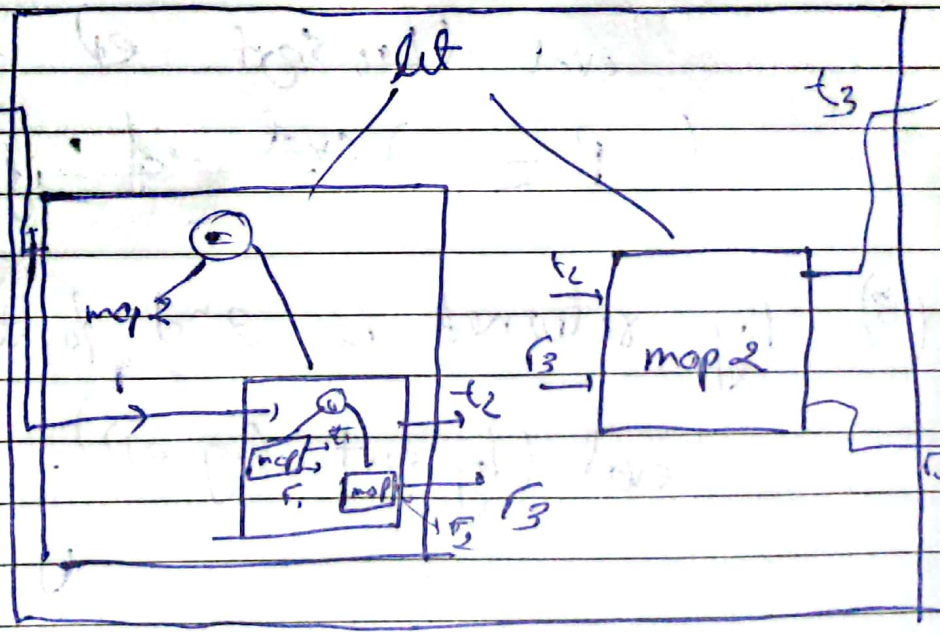
Q-1

$$T = \left\langle \begin{array}{l} (o) : \forall\beta\,\forall\alpha.\,(\beta\to\gamma)\to(\alpha\to\beta)\to\alpha\to\gamma \\ map : \forall\alpha\,\beta.\,(\alpha\to\beta)\to[\alpha]\to[\beta] \end{array} \right\rangle$$

let
map2 = map ∘ map
in
map2

T. f : type of map2.

Box-Diagram :



(i) App Rule :

$\Gamma \vdash_S o : \forall\beta\,\forall\alpha.\,(\beta\to\gamma)\to(\alpha\to\beta)\to\alpha\to\gamma$

$\Gamma \vdash_S map : \forall\alpha\,\beta.\,(\alpha\to\beta)\to[\alpha]\to[\beta]$

$\Gamma \vdash_S (o)\ map : \tau_1$

Scanned with CamScanner

19054

$t_1 \Rightarrow \forall \alpha, a, b.\ (\alpha \to (a \to b)) \to \alpha \to ([a] \to [b])$

This leads to conclude. [Inst Γ_1]

$\Gamma_1 \Rightarrow$ { Γ_5 ; (0) map :: $\forall \alpha, a, b.\ (\alpha \to (a \to b) \to \alpha \to ([a] \to [b])$

- Applying APP :-

$$\frac{\Gamma \vdash (0)\ map :: \forall \alpha, a, b\ (\alpha \to (a \to b)) \to \alpha \to ([a] \to [b]) \quad \Gamma_1 \vdash_5 map :: \forall \alpha \beta.\ (\alpha \to \beta) \to [\alpha] \to [\beta]}{\Gamma_2\ \Gamma_5 \vdash map\ map :: \forall \alpha, \beta.\ (\alpha \to \beta) \to [[\alpha]] \to [[\beta]]}$$

- $t_2 \Rightarrow \forall \alpha, \beta.\ (\alpha \to \beta) \to [[\alpha]] \to [[\beta]]$

Let map2 = map map
in map2

$$\frac{\Gamma \vdash_5 map\ map : t_2 \qquad \Gamma_3\ map2 : \Gamma_3 \vdash map^2 : t_3}{\Gamma \vdash_5 \text{let } map2 = map\ map \text{ in } map2 : t_3}$$

t_2 is of type t_3

$b : t_3 = \forall \alpha, \beta.\ (\alpha \to \beta) \to [[\alpha]] \to [[\beta]]$

The given is of t_3 which is

Q-2 190549

data LogSqrt = Val Float | Log LogSqrt | Sqrt LogSqrt

• eval used to check positive

eval check > 0
P z check ≥ 0

Checks for the interpreter:

(1) To check that the variable is float as follows

type state = Var → float val :: statecheck → state → (float, state)

(2) After checking that the variable is float, we need to check if the float is -ve.

Check positive

eval (Log sqrt e) =
(P z) eval ~~tof~~ ; if yes, return (log e)

eval (log e) =>

(3) log returned in 2nd point should be (+).

eval (log sqrt (log e)) = eval (log e) =>
if yes, return (sqrt (log e))

P.T.O

(continued)

Camin

19054

⟹

eval :: logsqrt → maybe float

eval (Val i) = return i

eval (log e_1) =

eval e_1 >>= $\lambda i \rightarrow$ if $i > 0$ return log i

else ⊥

eval (sqrt e_1) =

eval e_1 >>= $\lambda i \rightarrow$ if ($i \geq 0$) return sqrt i

else nothing.

types

return i = Just i

:: a → Maybe a