POPL
Assignment - 3

Aryan Agarwal 19CS30005

1.(a) Assumption

$true : bool \in EC$

$\mathcal{E}_0 \cup \{y : Ref\ bool\}$

By Identifier Rule,

$\mathcal{E}_0 \cup \{y : Ref\ bool\} \vdash y : Ref\ bool$ — ①

$\mathcal{E}_0 \cup \{y : Ref\ bool\} \vdash true : bool$ — ② [Constant Rule]

Using Assignment Rule,

$\mathcal{E}_0 \cup \{y : Ref\ bool\} \vdash y := true : Command$

∴ Type of expression : Command

$$\frac{\dfrac{}{\mathcal{E}_0 \cup \{y : Ref\ bool\} \vdash y : Ref\ bool}①\qquad \dfrac{}{\mathcal{E}_0 \cup \{y : Ref\ bool\} \vdash true : bool}②}{\mathcal{E}_0 \cup \{y : Ref\ bool\} \vdash y := true : Command}$$

(b) Given :- $func1 : monad \to \psi$, $func2 : \varphi \to \psi$

$\mathcal{E}_0 \cup \{x : monad\} \vdash func1 : monad \to \psi$ — ① [Constant Rule]

$\mathcal{E}_0 \cup \{x : monad\} \vdash x : monad.$ — ② [Identifier Rule]

By Application Rule on 1 and 2,

$\mathcal{E}_0 \cup \{x : monad\} \vdash func1\ x : \psi$ — ③

$\Sigma_0 \cup \{x : monad\} \vdash (func1\ x) : \Psi$ — (4) [Paren Rule]

$\Sigma_0 \vdash \lambda\,(x : monad).\,(func1\ x) : monad \rightarrow \Psi$ — (5) [Function Rule]

Similarly,

$\Sigma_0 \cup \{q : \varphi\} \vdash func2 : \varphi \rightarrow \Psi$ — (6) [Constant Rule]

$\Sigma_0 \cup \{q : \varphi\} \vdash q : \varphi$ — (7) [Identifier Rule]

By Application Rule on 6 and 7,

$\Sigma_0 \cup \{q : \varphi\} \vdash func2\ q : \Psi$ — (8)

$\Sigma_0 \cup \{q : \varphi\} \vdash (func2\ q) : \Psi$ — (9) [Paren Rule]

$\Sigma_0 \vdash \lambda\,(q : \varphi).\,(func2\ q) : \varphi \rightarrow \Psi$ — (10) [Function Rule]

Using Sequencing rule on 5 and 10,

$\Sigma_0 \vdash \lambda(x : monad).(func1\ x);\ \lambda(q : \varphi).(func2\ q) : \varphi \rightarrow \Psi$ — (11)

$\cancel{\dfrac{}{\Sigma_0 \cup \{x : monad\} \vdash func1 : monad \rightarrow \Psi}\ (1) \quad \dfrac{}{\Sigma_0 \cup \{x : monad\} \vdash x : monad}\ (2)}$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Sigma_0 \cup \{x : monad\} \vdash func1 : monad \rightarrow \Psi}\ (1) \quad \dfrac{}{\Sigma_0 \cup \{x : monad\} \vdash x : monad}\ (2)}{\Sigma_0 \cup \{x : monad\} \vdash (func1\ x) : \Psi}\ (4)}{\Sigma_0 \vdash \lambda(x : monad).(func1\ x) : monad \rightarrow \Psi}\ (5) \quad \dfrac{\dfrac{\dfrac{}{\Sigma_0 \cup \{q : \varphi\} \vdash func2 : \varphi \rightarrow \Psi}\ (6) \quad \dfrac{}{\Sigma_0 \cup \{q : \varphi\} \vdash q : \varphi}\ (7)}{\Sigma_0 \cup \{q : \varphi\} \vdash (func2\ q) : \Psi}\ (9)}{\Sigma_0 \vdash \lambda(q : \varphi).(func2\ q) : \varphi \rightarrow \Psi}\ (10)}{\Sigma_0 \vdash \lambda(x : monad).(func1\ x);\ \lambda(q : \varphi).(func2\ q) : \varphi \rightarrow \Psi}\ (11)$$

Type of expression : $\varphi \rightarrow \Psi$

(c) $\lambda(\omega : \Psi \to \Pi) . \lambda(x : \Psi) . (\omega((x | t) x))$

We have,

By Identifier Rule

$$\Sigma \cup \{x : \Psi\} \vdash x : \Psi$$

By Constant Rule

$$\Sigma_1 \vdash t : \Psi$$
$$\Sigma_1 \vdash | : \Psi \to \Psi \to \Psi \to \Psi$$

∴ Using Application Rule, twice [And Paren Rule]

$$\Sigma_1 \vdash (x | t) : \Psi \to \Psi$$

Again using Application Rule, [And Paren Rule]

$$\Sigma_1 \vdash ((x | t) x) : \Psi$$

By Identifier Rule,

$$\Sigma \cup \{\omega : \Psi \to \Pi\} \vdash \omega : \Psi \to \Pi$$

∴ Using Application Rule and Paren Rule,

$$\Sigma_1 \vdash (\omega((\cancel{x} | t) x)) : \Pi$$

Using function Rule

$$\Sigma_1 \vdash \lambda(x : \Psi) . (\omega((x | t) x)) : \Psi \to \Pi$$

Using Function Rule again

$$\Sigma_0 \vdash \lambda(\omega : \Psi \to \Pi) . \lambda(x : \Psi) . (\omega((x | t) x)) : \Psi \to \Pi \to \Psi \to \Pi$$

∴ Type of the expression : $\Psi \to \Pi \to \Psi \to \Pi$

(d) $\lambda(f: X \to Y) \cdot \lambda(x : X).\ f(+x)$

By Identifier Rule,

$$\Sigma \cup \{x : X\} \vdash x : X$$

By Constant Rule

$$\Sigma_1 \vdash + : X \to X$$

$\therefore$ Using Application Rule and Paren Rule

$$\Sigma_1 \vdash (+x) : X$$

Again by Identifier Rule,

$$\Sigma \cup \{f : X \to Y\} \vdash f : X \to Y$$

$\therefore$ Using Application Rule,

$$\Sigma_1 \vdash f(+x) : Y$$

Now, using Function Rule,

$$\Sigma_1 \vdash \lambda(x : X) \cdot f(+x) : X \to Y$$

Again using Function Rule,

$$\Sigma_0 \vdash \lambda(f : X \to Y).\ \lambda(x : X).f\ (+x) : X \to Y \to X \to Y$$

$\therefore$ Type of the expression : $X \to Y \to X \to Y$

$$\dfrac{\Sigma_1 \cup \{f : X \to Y\} \vdash f : X \to Y \qquad \dfrac{\Sigma_1 \cup \{x : X\} \vdash x : X \qquad \Sigma_1 \cup \{x : X\} \vdash + : X \to X}{\Sigma_1 \cup \{x : X\} \vdash (+x) : X}}{\dfrac{\Sigma_1 \cup \{f : X \to Y\} \cup \{x : X\} \vdash f(+x) : Y}{\dfrac{\Sigma_1 \cup \{f : X \to Y\} \vdash \lambda(x : X) . f(+x) : X \to Y}{\Sigma_1 \vdash \lambda(f : X \to Y) . \lambda(x : X) . f(0+x) : X \to Y \to X \to Y}}}$$

(e) Given,

$\Sigma_0 = \{x : \text{Ref Bool}, y : \text{Bool}\}$

succ = Int $\to$ Int

true = Bool

4 : Int

$\Sigma_0 \vdash$ succ : Int $\to$ Int ——— ① [Constant Rule]

$\Sigma_0 \vdash 4$: Int ——— ② [Constant Rule]

$\Sigma_0 \vdash$ succ 4 : Int ——— ③ [Application Rule]

$\Sigma_0 \vdash x$: Ref Bool ——— ④ [Identifier Rule]

$\Sigma_0 \vdash$ true : Bool ——— ⑤ [Constant Rule]

$\Sigma_0 \vdash x :=$ true : Command. ——— ⑥ [Assignment Rule]

Using sequencing rule on 3 and 6,

$\Sigma_0 \vdash$ succ 4; $x :=$ true : Command ——— ⑦

$\therefore$ Type of expression : Command

$$\dfrac{\dfrac{\overline{\mathcal{E}_0 \vdash succ : Int \to Int}\ (1) \qquad \overline{\mathcal{E}_0 \vdash 4 : Int}\ (2)}{\mathcal{E}_0 \vdash succ\ 4 : Int}\ (3) \qquad \dfrac{\overline{\mathcal{E}_0 \vdash x : Ref\ Bool}\ (4) \qquad \overline{\mathcal{E}_0 \vdash true : Bool}\ (5)}{\mathcal{E}_0 \vdash x := true : Command}\ (6)}{\mathcal{E}_0 \vdash succ\ 4\ ;\ x := true : Command}\ (7)$$

2.7 (a) Given,

$$\phi : \zeta_g \to \zeta_e$$
$$\Phi : \zeta_s \to \zeta_e$$

Let,

$$\mathcal{E}_1 = \mathcal{E}_0 \cup \{z : \zeta_e \to \zeta_e\}$$
$$\mathcal{E}_2 = \mathcal{E}_1 \cup \{+ : \zeta_e \to \zeta_e\}$$
$$\mathcal{E}_3 = \mathcal{E}_2 \cup \{n : \zeta_e \to \zeta_e\}$$
$$\mathcal{E}_4 = \mathcal{E}_3 \cup \{\mu : \zeta_e\}$$

$\mathcal{E}_4 \vdash n : \zeta_e \to \zeta_e$ —— (1) [Identifier Rule]

$\mathcal{E}_4 \vdash \mu : \zeta_e$ —— (2) [Identifier Rule]

$\mathcal{E}_4 \vdash n\mu : \zeta_e$ —— (3) [Application Rule]

$\mathcal{E}_4 \vdash (n\mu) : \zeta_e$ —— (4) [Paren Rule]

$\mathcal{E}_4 \vdash n(n\mu) : \zeta_e$ —— (5) [Application Rule (1 and 4)]

$\mathcal{E}_4 \vdash (n(n\mu)) : \zeta_e$ —— (6) [Paren Rule]

$\mathcal{E}_4 \vdash + : \zeta_e \to \zeta_e$ —— (7) [Identifier Rule]

$\mathcal{E}_4 \vdash +(n(n\mu)) : \zeta_e$ —— (8) [Application Rule (6 and 7)]

$\mathcal{E}_4 \vdash (+(n(n\mu))) : \zeta_e$ —— (9) [Paren Rule]

$\mathcal{E}_4 \vdash z : \zeta_e \to \zeta_e$ —— (10) [Identifier Rule]

$\mathcal{E}_4 \vdash z(+(n(n\mu))) : \zeta_e$ —— (11) [Application Rule (9 and 10)]

$\Sigma_3 \vdash \lambda(\mu:\xi_q).z(+(n(n\mu))): \xi_q \to \xi_q$ — (12) [Function Rule]

$\Sigma_2 \vdash \lambda(n:\xi_q \to \xi_q).\lambda(\mu:\xi_q).z(+(n(n\mu))): \xi_q \to \xi_q \to \xi_q \to \xi_q$ — (13)

[Function Rule]

$\Sigma_1 \vdash \lambda(+:\xi_q \to \xi_q).\lambda(n\ \xi_q \to \xi_q).\lambda(\mu:\xi_q).z(+(n(n\mu))):$
$\xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q$ — (14)

[Function Rule]

$\Sigma_0 \vdash \lambda(z:\xi_q \to \xi_q).\lambda(+\ \xi_q \to \xi_q).\lambda(n:\xi_q \to \xi_q).\lambda(\mu:\xi_q).$
$z(+(n(n\mu))).\ \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q$ — (15)

[Function Rule]

$\Sigma_0 \vdash (\lambda(z:\xi_q \to \xi_q).\lambda(+:\xi_q \to \xi_q).\lambda(n:\xi_q \to \xi_q).\lambda(\mu:\xi_q).$
$z(+(n(n\mu)))): \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q$ — (16)

[Paren Rule]

$\Sigma_0 \vdash \phi : \xi_q \to \xi_q$ — (17) [Constant Rule]

$\Sigma_0 \vdash (\lambda(z:\xi_q \to \xi_q).\lambda(+:\xi_q \to \xi_q).\lambda(n:\xi_q \to \xi_q).\lambda(\mu:\xi_q).$
$z(+(n(n\mu))))\phi : \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q$ — (18)

[Application Rule]

$\Sigma_0 \vdash ((\lambda(z:\xi_q \to \xi_q).\lambda(+:\xi_q \to \xi_q).\lambda(n:\xi_q \to \xi_q).\lambda(\mu:\xi_q).$
$z(+(n(n\mu))))\phi): \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q \to \xi_q$ — (19)

[Paren Rule]

$\Sigma_0 \vdash \Phi : \xi_q \to \xi_q$ — (20) [Constant Rule]

$\Sigma_0 \vdash ((\lambda(2 : \mathcal{C}_q \to \mathcal{C}_q).\lambda(+ : \mathcal{C}_q \to \mathcal{C}_q).\lambda(n : \mathcal{C}_q \to \mathcal{C}_q).\lambda(n : \mathcal{C}_q).$
$2(+(n(n\,A)))\,\phi)\,\Phi : \mathcal{C}_q \to \mathcal{C}_q \to \mathcal{C}_q \to \mathcal{C}_q$ —— (21)

[Application Rule]

$\therefore$ $\boxed{\text{Type of the expression: } \mathcal{C}_q \to \mathcal{C}_q \to \mathcal{C}_q \to \mathcal{C}_q}$

(b) Given,

$\phi : \theta \to \theta \to \theta$
$true : \theta$

Let,
$\Sigma_1 = \Sigma_0 \cup \{funcl : \theta \to char\}$
and $\Sigma_2 = \Sigma_1 \cup \{T : \theta\}$.

Now, $\Sigma_2 \vdash T : \theta$ —— (1) [Identifier Rule]
$\Sigma_2 \vdash \phi : \theta \to \theta \to \theta$ —— (2) [Constant Rule]

Using Application Rule,
$\Sigma_2 \vdash \phi\,T : \theta \to \theta$ —— (3) [We are using prefix notation, assuming $\phi\,T$ true instead of $T\,\phi$ true]

$\Sigma_2 \vdash true : \theta$ —— (4) [Constant Rule]

By Application Rule,
$\Sigma_2 \vdash \phi\,T\;true : \theta$ —— (5)

$\Sigma_2 \vdash (\phi\,T\;true) : \theta$ —— (6) [Paren Rule]

$\Sigma_2 \vdash funcl : \theta \to char$ —— (7) [Identifier Rule]

$\Sigma_2 \vdash funcl\,(\phi\,T\;true) : char$ —— (8) [Application Rule]

$\Sigma_1 \vdash \lambda(\gamma:\theta).\,\text{func1}\,(\phi\ \gamma\ \text{true}) : \theta \to \text{char}$ —— (9) [Function Rule]

$\Sigma_0 \vdash \lambda(\text{func1} : \theta \to \text{char}).\,\lambda(\gamma:\theta).$
$\text{func1}\,(\phi\ \gamma\ \text{true}) : \theta \to \text{char} \to \theta \to \text{char}$ —— (10) [Function Rule]

$\therefore$ Type of expression : $\theta \to \text{char} \to \theta \to \text{char}$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Sigma_2 \vdash \text{func1} : \theta \to \text{char}}(7) \quad \dfrac{\dfrac{\dfrac{}{\Sigma_2 \vdash \phi : \theta \to \theta \to \theta}(2) \quad \dfrac{}{\Sigma_2 \vdash \gamma : \theta}(1)}{\Sigma_2 \vdash \phi\ \gamma : \theta \to \theta}(3) \quad \dfrac{}{\Sigma_2 \vdash \text{true} : \theta}(4)}{\Sigma_2 \vdash (\phi\ \gamma\ \text{true}) : \theta}(6)}{\Sigma_2 \vdash \text{func1}\,(\phi\ \gamma\ \text{true}) : \text{char}}(8)}{\Sigma_1 \vdash \lambda(\gamma:\theta).\,\text{func1}\,(\phi\ \gamma\ \text{true}) : \theta \to \text{char}}(9)}{\Sigma_0 \vdash \lambda(\text{func1}:\theta \to \text{char}).\,\lambda(\gamma:\theta).\,\text{func1}\,(\phi\ \gamma\ \text{true}) : \theta \to \text{char} \to \theta \to \text{char}}(10)$$

* Here, we ~~shoo~~ have prefix ~~for ϕ, insto~~ notation for ϕ, instead of the infix notation.