Written Assignment 2

Vinay Patil

vpatil3@wisc.edu

1 Types, Terms, and Contexts (5)

1.1 Fill in the Types

(a) $G \vdash x : Bool, y : Bool$ and $t = Bool \to Bool$

(b) $G \vdash x : Bool \to Bool, y : Bool$ and $t = (Bool \to Bool) \to Bool \to Bool$

(c) $G \vdash x : Bool$ and $t = Bool \to Bool$

(d) $G \vdash x : Bool \to Bool, y : Bool$ and $t = (Bool \to Bool) \to Bool$

(e) $G \vdash x : Bool \to Bool, y : Bool$ and $t = (Bool \to Bool) \to Bool$

(f) $G \vdash x : Bool, y : Bool$ and $t = Bool \to Bool$

(g) $G \vdash x : Bool, y : Bool, z : Bool$ and $t = Bool \to Bool \to Bool \to Bool$

1.2 Fill in the Terms

(a) $\lambda x : Bool.\ y : Bool$

(b) $\lambda x : Bool.\ \lambda y : Bool.\ if\ x : Bool\ then\ x : Bool\ else\ y : Bool$

(c) $\lambda x : Bool \to Bool.\ y : Bool$

(d) $(\lambda x : (Bool \to Bool \to Bool).\ z : Bool)\ (\lambda x : Bool.\ \lambda y : Bool.\ if\ x : Bool\ then\ x : Bool\ else\ y : Bool)$
(Also provided the function input which is fed to first lambda to get the required input typing)

(e) $if\ x\ then\ x\ else\ ((y\ x)\ x)$

2 Inhabitants of a Type (10)

2.1 How Many Programs?

(a) 2 - (true) and (false)

(b) 2 - $(\lambda x : true.\ x1 : false)$ and $(\lambda x : false.\ x1 : true)$.

(c) Y (Assumption for (b) and (c) is that two functions f, g to be equal if f x and g x reduce to equal values). So the total unique map values will be equal to Y.

2.2 Programs as Proofs

(a) $\lambda x : P.\ y : P$

(b) $\lambda x : P \times Q.\ fst(x)$

(c) $\lambda x : P \times Q.\ snd(x)\ and\ fst(x)$

(d) $\lambda x : P.\ left(e1 : P \times Q)\ or\ right(e1 : P \times Q)$

(e) $\lambda x : P \times (P \to Q).\ (snd(x)\ fst(x))$

(f) $\lambda x : P.\ (\lambda y : Q.\ z : P)$

(g) $\lambda x : (P \to Q) \times (Q \to R).\ (\lambda y : P.\ z : R)$

Are there programs of the following types?

(a) If say y:Q was in the context then we can always create a lambda program which takes in x:P and returns y:Q. Not possible with an empty context.

(b) No because the return is always P which may not be true all the time since the input is P or Q.

(c) No because a function from P to Q cannot step to P using the grammar, but can step to Q if applied on x:P.

3 Adding Triples (15)

The first part is easiest: we add a new type for triples:

$$t ::= \cdots \mid (t_1, t_2, t_3)$$

3.1 Projections

(a)

$$e ::= \cdots \mid (e_1, e_2, e_3) \mid \mathit{fst}(e) \mid \mathit{snd}(e) \mid \mathit{thd}(e)$$

(b)

$$\frac{G \vdash e_1 : t_1 \quad G \vdash e_2 : t_2 \quad G \vdash e_3 : t_3}{G \vdash (e_1,\ e_2,\ e_3) : (t1, t2, t3)}$$

$$\frac{G \vdash e : (t1, t2, t3)}{G \vdash \mathit{fst}(e) : t1}$$

$$\frac{G \vdash e : (t1, t2, t3)}{G \vdash \mathit{snd}(e) : t2}$$

$$\frac{G \vdash e : (t1, t2, t3)}{G \vdash \mathit{thd}(e) : t3}$$

(c)

$$\frac{e_1 \to e_1^1}{(e_1,\ e_2,\ e_3) \to (e_1^1,\ e_2,\ e_3)}$$

$$\frac{e_2 \to e_2^1}{(e_1,\ e_2,\ e_3) \to (e_1,\ e_2^1,\ e_3)}$$

$$\frac{e_3 \to e_3^1}{(e_1,\ e_2,\ e_3) \to (e_1,\ e_2,\ e_3^1)}$$

$$\frac{e_1 \to v_1 \quad e_2 \to v_2 \quad e_3 \to v_3}{fst(e_1,\ e_2,\ e_3) \to fst(v_1, v_2, v_3)}$$

$$\frac{v_1}{fst(v_1, v_2, v_3) \to v_1}$$

$$\frac{e_1 \to v_1 \quad e_2 \to v_2 \quad e_3 \to v_3}{snd(e_1,\ e_2,\ e_3) \to snd(v_1, v_2, v_3)}$$

$$\frac{v_2}{snd(v_1, v_2, v_3) \to v_2}$$

$$\frac{e_1 \to v_1 \quad e_2 \to v_2 \quad e_3 \to v_3}{thd(e_1,\ e_2,\ e_3) \to thd(v_1, v_2, v_3)}$$

$$\frac{v_3}{fst(v_1, v_2, v_3) \to v_3}$$

3.2 Pattern matching

(a)

$$e ::= \cdots \mid case\ e\ of\ \{left(x) \to e_l; mid(x) \to e_m; right(y) \to e_r\}$$

(b)

$$\frac{G \vdash e : (t1, t2, t3)\ \ G \vdash e_l : t1\ \ G \vdash e_m : t2\ \ G \vdash e_r : t3}{G \vdash case\ e\ of\ \{left(x) \to e_l; mid(y) \to e_m; right(z) \to e_r\} : t}$$

(c)

$$\frac{e \to e^1}{case\ e\ of\ \{left(x) \to e_l; mid(y) \to e_m; right(z) \to e_r\} \to case\ e^1\ of\ \{left(x) \to e_l; mid(y) \to e_m; right(z) \to e_r\}}$$

$$\frac{v_1 \quad v_2 \quad v_3}{case\ e^1\ of\ \{left(x) \to e_l; mid(y) \to e_m; right(z) \to e_r\} \to e_l[x \to v1] \quad e_m[y \to v2] \quad e_r[z \to v3]}$$

(Note: As per the suggestion ignoring the rules which are similar to the three rule for the triples in 3.1(c))