

Supervision 1

1.3

a

$$\frac{\frac{f : \text{Bool} \rightarrow \text{Bool} \vdash \text{false} : \text{Bool}}{\text{false}} \quad \frac{f : \text{Bool} \rightarrow \text{Bool} \vdash \text{true} : \text{Bool}}{\text{false}} \quad \frac{f : \text{Bool} \rightarrow \text{Bool} \vdash \text{false} : \text{Bool}}{\text{true}}}{f : \text{Bool} \rightarrow \text{Bool} \vdash \text{if false then true else false} : \text{Bool}} \text{if}$$

$$\frac{f : \text{Bool} \rightarrow \text{Bool} \vdash f(\text{if false then true else false}) : \text{Bool}}{f : \text{Bool} \rightarrow \text{Bool} \vdash f(\text{if false then true else false}) : \text{Bool}} \text{app}$$

Figure 1: 1.3(a)

b

$$\frac{\frac{f : \text{Bool} \rightarrow \text{Bool}, x : \text{Bool} \vdash x : \text{Bool}}{x : \text{Bool}} \text{var} \quad \frac{f : \text{Bool} \rightarrow \text{Bool}, x : \text{Bool} \vdash \text{false} : \text{Bool}}{\text{false}} \quad \frac{f : \text{Bool} \rightarrow \text{Bool}, x : \text{Bool} \vdash x : \text{Bool}}{x : \text{Bool}} \text{var}}{f : \text{Bool} \rightarrow \text{Bool}, x : \text{Bool} \vdash \text{if } x \text{ then false else } x : \text{Bool}} \text{if}$$

$$\frac{f : \text{Bool} \rightarrow \text{Bool}, x : \text{Bool} \vdash f(\text{if } x \text{ then false else } x) : \text{Bool} \rightarrow \text{Bool}}{f : \text{Bool} \rightarrow \text{Bool} \vdash \lambda x : \text{Bool} . f(\text{if } x \text{ then false else } x) : \text{Bool} \rightarrow \text{Bool}} \text{app}$$

$$\frac{f : \text{Bool} \rightarrow \text{Bool} \vdash \lambda x : \text{Bool} . f(\text{if } x \text{ then false else } x) : \text{Bool} \rightarrow \text{Bool}}{f : \text{Bool} \rightarrow \text{Bool} \vdash \lambda x : \text{Bool} . f(\text{if } x \text{ then false else } x) : \text{Bool} \rightarrow \text{Bool}} \text{lambda}$$

Figure 2: 1.3(b)

1.4

Suppose we want to construct a proof that

$$\cdot \vdash x : t$$

This does not match rule **VAR**, or rule **LAM**. Therefore, it must be of the form of rule **APP**.

Therefore, x must be one of the two terms whose types are derived from **APP**.

Case 0: $x = E_1$, $t = \tau' \rightarrow \tau$

This would require us to prove that $\cdot \vdash xE_2 : \tau$

Case 1: $x = E_2$, $t = \tau'$

This would require us to prove that $\cdot \vdash E_1x : \tau$

In either case, we are left with a problem in the same shape as the original. Therefore, any derivation tree for the type of x would have to apply **APP** infinitely many times without ever reaching an axiom.

1.5

a

Lemma 1: There exists no term n such that $\cdot \vdash n : 0$.

Lemma 1 is analogous to the fact that one cannot prove **false**.

Case 0

Assumption 1: there exists a term p such that $\cdot \vdash p : \pi$

Assumption 2: there exists a term p' such that $\cdot \vdash p' : \pi \rightarrow 0$

By **APP**, $\cdot \vdash p'p : 0$

This violates lemma 1, and so is a contradiction.

This is analogous to the fact that P and $\neg P$ cannot both be true.

Case 1

Assumption 1: there does not exist a term p such that $\cdot \vdash p : \pi$

Assumption 2: there does not exist a term p' such that $\cdot \vdash p' : \pi \rightarrow 0$

Claim: By assumption 1, π is therefore equivalent to 0. (I don't know how to prove this)

By **LAM**, $\cdot \vdash \lambda x : 0.x : \pi \rightarrow 0$.

This violates assumption 2, and so is a contradiction

This is analogous to the fact that P and $\neg P$ cannot both be false.

b

i

Type:

$$((\pi \rightarrow 0) \times (\pi' \rightarrow 0)) \rightarrow ((\pi + \pi') \rightarrow 0)$$

Term:

$$\lambda x : ((\pi \rightarrow 0) \times (\pi' \rightarrow 0)). (\lambda y : (\pi + \pi'). \text{case}(y, \text{Ly}_1 \rightarrow (\text{fst } x)y_1, \text{Ry}_2 \rightarrow (\text{snd } x)y_2))$$

ii

Type:

$$((\pi + \pi') \rightarrow 0) \rightarrow ((\pi \rightarrow 0) \times (\pi' \rightarrow 0))$$

Term:

$$\lambda y : ((\pi + \pi') \rightarrow 0). < \lambda x_1 : \pi. y\text{L}x_1, \lambda x_2 : \pi'. y\text{R}x_2 >$$

iii

Type:

$$((\pi \rightarrow 0) + (\pi' \rightarrow 0)) \rightarrow ((\pi \times \pi') \rightarrow 0)$$

There is not a term which has this type

iv

Type:

$$((\pi \times \pi') \rightarrow 0) \rightarrow ((\pi \rightarrow 0) + (\pi' \rightarrow 0))$$

There is not a term which has this type

c

This means that the correspondence between STLC and propositional logic is not perfect. Specifically, there are some propositions which are provable in propositional logic, such that the STLC has no terms of the corresponding type.

1.6

Proof that if $\Gamma \vdash x : \tau$, then $fv(x) \subseteq dom(\Gamma)$.

Case 0: $1I$

$$x = <>.$$

$$\text{Therefore } fv(x) = \{\} \subseteq dom(\Gamma)$$

Case 1: $\times I$

$$x = \langle e, e' \rangle$$

By induction, $fv(e) \subseteq dom(\Gamma)$

By induction, $fv(e') \subseteq dom(\Gamma)$

$$fv(x) = fv(e) \cup fv(e')$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 2: E_1

$$x = \text{fst } e$$

By induction, $fv(e) \subseteq dom(\Gamma)$

$$fv(x) = fv(e)$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 3: E_2

$$x = \text{snd } e$$

By induction, $fv(e) \subseteq dom(\Gamma)$

$$fv(x) = fv(e)$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 4: Hyp

$$x : \tau \in \Gamma$$

Therefore $x \in dom(\Gamma)$

Claim: $fv(x) = \{x\}$ (I don't know how to prove this)

Therefore $fv(x) \subseteq dom(\Gamma)$

Case 5: $\rightarrow I$

$$x = \lambda y : T. e$$

By induction, $fv(e) \subseteq (dom(\Gamma) \cup \{y\})$

$$fv(x) = fv(e) \setminus \{y\}$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 6: $\rightarrow E$

$$x = ee'$$

By induction, $fv(e) \subseteq dom(\Gamma)$

By induction, $fv(e') \subseteq dom(\Gamma)$

$$fv(x) = fv(e) \cup fv(e')$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 7: $+I_1$

$$x = \text{Le}$$

By induction, $fv(e) \subseteq dom(\Gamma)$

$$fv(x) = fv(e)$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 8: $+I_2$

$$x = \text{Re}$$

By induction, $fv(e) \subseteq dom(\Gamma)$

$$fv(x) = fv(e)$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 9: $+E$

$$x = \text{case}(e, \text{Ly} \rightarrow e', \text{Rz} \rightarrow e'')$$

By induction, $fv(e) \subseteq dom(\Gamma)$

By induction, $fv(e') \subseteq (dom(\Gamma) \cup \{y\})$

By induction, $fv(e'') \subseteq (dom(\Gamma) \cup \{z\})$

$$fv(x) = fv(e) \cup (fv(e') \setminus \{y\}) \cup (fv(e'') \setminus \{z\})$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Case 10: $+E$

$$x = \text{abort } e$$

By induction $fv(e) \subseteq dom(\Gamma)$

$$fv(x) = fv(e)$$

Therefore, $fv(x) \subseteq dom(\Gamma)$

Conclusion

Suppose $\Gamma = \cdot$

Therefore, $fv(x) \subseteq dom(\cdot)$

Therefore, $fv(x) \subseteq \emptyset$

Therefore, $fv(x) = \emptyset$

Therefore x has no free variables.