1.1 FV(e) definition extensions

$$FV(\text{not } e) = FV(e)$$

$$FV(e_1 \,\&\&\, e_2) = FV(e_1) \cup FV(e_2)$$

$$FV(e_1 \,||\, e_2) = FV(e_1) \cup FV(e_2)$$

$$FV(\text{to\_float } e) = FV(e)$$

$$FV(\text{trunc } e) = FV(e)$$

$$FV\left(\sum_{x=e_1}^{e_2} e\right) = \cancel{FV(e_1 + e_2) = FV(e}$$

$$\cancel{FV(e_1 + \dots + e_2) = FV(e_1) \cup \dots\dots \cup FV(e_2)}$$

$$\cancel{FV(e}$$

$$\cancel{FV([x/e_1]e) \cup FV([x/e_2]e)}$$

$$\cancel{FV(e_1) \cup FV(e_2) \cup FV(e) \setminus x}$$

$$FV(e_1) \cup FV(e_2) \cup (FV(e) / \{x\})$$

1.3 $[e/x]e'$ extension

$$[e/x](\text{Lt } e_1, e_1) = \text{Lt}(([e/x]e_1)([e/x]e_2))$$

$$[e/x](\text{Not } e_1) = \text{not}([e/x]e_1)$$

$$[e/x](e_1 \&\& e_2) = ([e/x]e_1 \;\&\&\; [e/x]e_2)$$

$$[e/x](e_1 \,||\, e_2) = ([e/x]e_1 \;||\; [e/x]e_2).$$

$$[e/x](\text{to\_float } e_1) = (\text{to\_float}([e/x]e_1))$$

$$[e/x](\text{trunc } e_1) = \text{trunc}([e/x]e_1).$$

$$[e/x]\left(\sum_{y=e_1}^{e_2} e_3\right) = \left(\sum_{y=[e/x]e_1}^{[e/x]e_2} [e/x]e_3\right)$$

$$\text{let } \cancel{x = 5 \text{ in}} \left(\cancel{\sum_{x=1}^{3} x}\right)$$

$$\cancel{\text{let } x = 5 \text{ in } (1+2+3)}$$

$$\Rightarrow \cancel{\text{let } x = 5 \text{ in } 6}.$$

$$\Rightarrow [e/x](\text{Eq}(e_1, e_2))$$

$$= \text{Eq}(([e/x]e_1)([e/x]e_2))$$

1.5 Define on paper typing rules for each of the expressions

$$\frac{\Gamma \vdash e_1 : T \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash Lt(e_1, e_2) : bool}$$

$$\frac{\Gamma \vdash e : bool \qquad \Gamma \vdash e_2 : bool}{\Gamma \vdash And(e_1, e_2) : bool}$$

$$\frac{\Gamma \vdash e : bool \qquad \Gamma \vdash e_2 : bool}{\Gamma \vdash Or(e_1, e_2) : bool}$$

$$\frac{\Gamma \vdash e : bool}{\Gamma \vdash Not(e) : bool}$$

$$\frac{\Gamma \vdash e : float}{\Gamma \vdash Trunc(e) : int}$$

$$\frac{\Gamma \vdash e : int}{\Gamma \vdash ToFloat(e) : float}$$

$$\frac{}{\Gamma \vdash x : var}$$

$$\frac{\Gamma \vdash e_1 : bool \qquad \Gamma \vdash e_2 : T \qquad \Gamma \vdash e_3 : T}{\Gamma \vdash If\_Then\_Else(e_1, e_2, e_3) : T}$$

1.6.0 For the typing rule for $\sum_{e_1}^{e_2} e$

$$\frac{\cancel{\Gamma \vdash e_1 : int} \quad \cancel{\Gamma \vdash e_2 : int} \quad \cancel{\Gamma \vdash e : T}}{\cancel{\Gamma \vdash \sum_{x=e_1}^{e_2} e : T}}$$

$$\frac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int \quad \Gamma, x : \underline{int} \vdash \underline{e : float}.}{\Gamma \vdash \sum_{x=e_1}^{e_2} e : float.}$$

1.7 Operational Semantics

$$\frac{e \Downarrow \text{true}}{\text{Not}(e) \Downarrow \text{false}}\ \text{B-NOTTRUE} \qquad \frac{e \Downarrow \text{false}}{\text{Not}(e) \Downarrow \text{true}}\ \text{B-NOTFALSE}$$

$$\frac{e \Downarrow fl \quad \text{int\_of\_float}(fl) \Downarrow v}{\text{Trunc } e \Downarrow v}\ \text{B-TRUNCFLOAT}$$

$$\frac{e \Downarrow n \quad \text{float\_of\_int}(n) \Downarrow v}{\text{Tofloat}(e) \Downarrow v}\ \text{B-TOFLOATINT}$$

$$\frac{e_2 \Downarrow \text{false}}{\text{And } e_1 e_2 \Downarrow \text{false}}\ \text{B-ANDEFALSE} \qquad \frac{e_1 \Downarrow \text{false}}{\text{And } e_1 e_2 \Downarrow \text{false}}\ \text{B-ANDESFALSE}$$

$$\frac{e_1 \Downarrow \text{true} \quad e_2 \Downarrow \text{true}}{\text{And } e_1 e_2 \Downarrow \text{true}}\ \text{B-ANDTrue.}$$

$$\frac{}{n \Downarrow n}\ \text{B-INT} \qquad \frac{}{fl \Downarrow fl}\ \text{B-FLOAT.}$$

$$\boxed{\frac{e_1 \Downarrow n_1 \quad e_2 \Downarrow n_2 \quad \left(\sum_{y=n_1}^{n_2} \text{subst}(y,x)e_3\right) \Downarrow v}{\text{Sum}(e_1, e_2, (x, e_3)) \Downarrow v}}$$

$$\frac{e_1 \Downarrow \text{false} \quad e_2 \Downarrow \text{false}}{\text{or } e_1 e_2 \Downarrow \text{false}}$$

$$\frac{e_1 \Downarrow \text{true} \quad e_2 \Downarrow (\text{true} \,||\, \text{false})}{\text{or } e_1, e_2 \Downarrow \text{true}} \qquad \frac{e_1 \Downarrow (\text{true} \,||\, \text{false}) \quad e_2 \Downarrow \text{true}}{\text{or } e_1 e_2 \Downarrow \text{true}}$$

$$\frac{e_1 \Downarrow N_1 \quad e_2 \Downarrow N_2, \quad (N_1 < N_2) \Downarrow true}{Lt\ e_1, e_2 \Downarrow true}$$

$$\frac{e_1 \Downarrow N_1 \quad e_2 \Downarrow N_2, \quad (N_1 < N_2) \Downarrow false}{Lt\ e_1, e_2 \Downarrow false}$$

(where N_i are numbers (either both ints or both floats)).

$$\frac{e_1 \Downarrow v_1 \quad e_2 \Downarrow v_2 \quad (v_1 = v_2) \Downarrow true}{eq\ e_1, e_2 \Downarrow true}$$

$$\frac{e_1 \Downarrow v_1 \quad e_2 \Downarrow v_2 \quad (v_1 = v_2) \Downarrow false}{eq\ e_1, e_2 \Downarrow false}$$

(where v_1 and v_2 are values of the same type, either ints, floats, or Bools.)

Q2 Type Inference

- Does there exist an instantiation for the type variables α and β s.t.

  fun x y -> (x y) + 2.0 has type $\alpha \Rightarrow \beta$ in Ocaml

  Answer: clearly not since '+' expects two ints, however '2.0' is a float so we have a type error.

- Does there exist an instantiation for the type variables α, β, γ, and δ s.t.

  fun u w x -> u (w x) || (u 2.0) has type

  $$(\alpha \Rightarrow \beta \Rightarrow \gamma) \Rightarrow \delta$$

  Answer: Yes

  α = (float -> bool)
  β = ('a -> float)
  γ = 'a
  δ = bool