

```

module source where

open import lib

-- Operator precedence and associativity
infix 1 _≤:_
infix 2 _→_
infix 4 _⊢_
infix 4 _∈_
infixl 5 _',_
infixr 7 _⇒_

-- Types
data Type : Set where
  comm : Type
  intexp : Type
  intacc : Type
  intvar : Type
  _⇒_ : Type → Type → Type

-- Contexts
data Context : Set where
  · : Context
  _',_ : Context → Type → Context

-- Variables and the lookup judgement
data _∈_ : Type → Context → Set where
  Z : ∀{Γ A} → A ∈ Γ , A
  S : ∀{Γ A B} → B ∈ Γ → B ∈ Γ , A

-- Terms and the typing judgement
data _⊢_ : Context → Type → Set where
  Var : ∀{Γ A} → A ∈ Γ → Γ ⊢ A

  -- lambda function and application
  Lambda : ∀{Γ A B} → Γ , A ⊢ B → Γ ⊢ A ⇒ B
  App : ∀{Γ A B} → Γ ⊢ A ⇒ B → Γ ⊢ A → Γ ⊢ B

  -- command
  Skip : ∀{Γ} → Γ ⊢ comm
  Seq : ∀{Γ} → Γ ⊢ comm → Γ ⊢ comm → Γ ⊢ comm

  -- intexp
  Lit : ∀{Γ} → ℤ → Γ ⊢ intexp
  Neg : ∀{Γ} → Γ ⊢ intexp → Γ ⊢ intexp
  Plus : ∀{Γ} → Γ ⊢ intexp → Γ ⊢ intexp → Γ ⊢ intexp

data Value : ∀{Γ A} → Γ ⊢ A → Set where
  V-Lit : ∀{Γ} {i : ℤ} → Value (Lit {Γ} i)

  V-Lambda : ∀{Γ A B} {F : Γ , A ⊢ B} → Value (Lambda {Γ} F)

-- Renaming
ext : ∀{Γ Δ} → (∀{A} → A ∈ Γ → A ∈ Δ) → (∀{A B} → B ∈ Γ , A → B ∈ Δ , A)
ext ρ Z = Z

```

ext ρ (S x) = S (ρ x)

rename : $\forall \{\Gamma \Delta\} \rightarrow (\forall \{A\} \rightarrow A \in \Gamma \rightarrow A \in \Delta) \rightarrow (\forall \{A\} \rightarrow \Gamma \vdash A \rightarrow \Delta \vdash A)$

rename ρ ($\text{Var } x$) = $\text{Var } (\rho \ x)$

rename ρ ($\text{Lambda } F$) = $\text{Lambda } (\text{rename } (\text{ext } \rho) \ F)$

rename ρ ($\text{App } F \ E$) = $\text{App } (\text{rename } \rho \ F) (\text{rename } \rho \ E)$

rename ρ Skip = Skip

rename ρ ($\text{Seq } c_1 \ c_2$) = $\text{Seq } (\text{rename } \rho \ c_1) (\text{rename } \rho \ c_2)$

rename ρ ($\text{Lit } i$) = $\text{Lit } i$

rename ρ ($\text{Neg } I$) = $\text{Neg } (\text{rename } \rho \ I)$

rename ρ ($\text{Plus } I_1 \ I_2$) = $\text{Plus } (\text{rename } \rho \ I_1) (\text{rename } \rho \ I_2)$

-- Simultaneous substitution

exts : $\forall \{\Gamma \Delta\} \rightarrow (\forall \{A\} \rightarrow A \in \Gamma \rightarrow \Delta \vdash A) \rightarrow (\forall \{A B\} \rightarrow B \in \Gamma, A \rightarrow \Delta, A \vdash B)$

exts σ $Z = \text{Var } Z$

exts σ (S x) = $\text{rename } S \ (\sigma \ x)$

subst : $\forall \{\Gamma \Delta\} \rightarrow (\forall \{A\} \rightarrow A \in \Gamma \rightarrow \Delta \vdash A) \rightarrow (\forall \{A\} \rightarrow \Gamma \vdash A \rightarrow \Delta \vdash A)$

subst σ ($\text{Var } x$) = $\sigma \ x$

subst σ ($\text{Lambda } F$) = $\text{Lambda } (\text{subst } (\text{exts } \sigma) \ F)$

subst σ ($\text{App } F \ E$) = $\text{App } (\text{subst } \sigma \ F) (\text{subst } \sigma \ E)$

subst σ Skip = Skip

subst σ ($\text{Seq } c_1 \ c_2$) = $\text{Seq } (\text{subst } \sigma \ c_1) (\text{subst } \sigma \ c_2)$

subst σ ($\text{Lit } i$) = $\text{Lit } i$

subst σ ($\text{Neg } I$) = $\text{Neg } (\text{subst } \sigma \ I)$

subst σ ($\text{Plus } I_1 \ I_2$) = $\text{Plus } (\text{subst } \sigma \ I_1) (\text{subst } \sigma \ I_2)$

-- Single substitution

$_[_]$: $\forall \{\Gamma A B\} \rightarrow \Gamma, B \vdash A \rightarrow \Gamma \vdash B \rightarrow \Gamma \vdash A$

$_[_]$ $\{\Gamma\} \{A\} \{B\} \ N \ M = \text{subst } \{\Gamma, B\} \ \{\Gamma\} \ \sigma \ \{A\} \ N$

where

$\sigma : \forall \{A\} \rightarrow A \in \Gamma, B \rightarrow \Gamma \vdash A$

$\sigma \ Z = M$

$\sigma \ (S \ x) = \text{Var } x$

-- Reduction

data $_ \rightarrow _$: $\forall \{\Gamma A\} \rightarrow (\Gamma \vdash A) \rightarrow (\Gamma \vdash A) \rightarrow \text{Set where}$

App-cong₁ : $\forall \{\Gamma A B\} \{F F' : \Gamma \vdash A \Rightarrow B\} \{E : \Gamma \vdash A\} \rightarrow F \rightarrow F' \rightarrow \text{App } F \ E \rightarrow \text{App } F' \ E$

App-cong₂ : $\forall \{\Gamma A B\} \{V : \Gamma \vdash A \Rightarrow B\} \{E E' : \Gamma \vdash A\} \rightarrow \text{Value } V \rightarrow E \rightarrow E' \rightarrow \text{App } V \ E \rightarrow \text{App } V \ E'$

Lambda- β : $\forall \{\Gamma A B\} \{F : \Gamma, A \vdash B\} \{V : \Gamma \vdash A\} \rightarrow \text{Value } V \rightarrow \text{App } (\text{Lambda } F) \ V \rightarrow F \ [\ V \]$

-- Subtype relation

data $_ \leq _$: $\text{Type} \rightarrow \text{Type} \rightarrow \text{Set where}$

$\leq \text{-refl} : \forall \{T\} \rightarrow T \leq T$

$\leq \text{-trans} : \forall \{T T' T''\} \rightarrow T \leq T' \rightarrow T' \leq T'' \rightarrow T \leq T''$

$\leq \text{-fn} : \forall \{T_1 T_1' T_2 T_2'\} \rightarrow T_1' \leq T_1 \rightarrow T_2 \leq T_2' \rightarrow T_1 \Rightarrow T_2 \leq T_1' \Rightarrow T_2'$

var- $\leq \text{-exp} : \text{intvar} \leq \text{intexp}$

var- $\leq \text{-acc} : \text{intvar} \leq \text{intacc}$