

module source where

open import lib

-- Operator precedence and associativity

infix 1 \leq : _

infix 2 \longrightarrow _

infix 4 \vdash _

infix 4 \in _

infixl 5 \rightarrow _

infixr 7 \Rightarrow _

-- Types

data Type : Set where

comm intexp intacc intvar : Type

\Rightarrow _ : Type \rightarrow Type \rightarrow Type

-- Subtype relation

data \leq : _ : Type \rightarrow Type \rightarrow Set where

\leq :-refl : $\forall \{A\} \rightarrow A \leq A$

\leq :-trans : $\forall \{A A' A''\} \rightarrow A \leq A' \rightarrow A' \leq A'' \rightarrow A \leq A''$

\leq :-fn : $\forall \{A A' B B'\} \rightarrow A' \leq A \rightarrow B \leq B' \rightarrow A \Rightarrow B \leq A' \Rightarrow B'$

var- \leq :-exp : intvar \leq intexp

var- \leq :-acc : intvar \leq intacc

-- Contexts

data Context : Set where

\cdot : Context

\rightarrow _ : Context \rightarrow Type \rightarrow Context

-- Variables and the lookup judgement

data \in : Type \rightarrow Context \rightarrow Set where

Zero : $\forall \{\Gamma A\} \rightarrow A \in \Gamma, A$

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

-- Terms and the typing judgement

data \vdash : Context \rightarrow Type \rightarrow Set where

Var : $\forall \{\Gamma A\} \rightarrow A \in \Gamma \rightarrow \Gamma \vdash A$

-- subtyping

Sub : $\forall \{\Gamma A B\} \rightarrow \Gamma \vdash A \rightarrow A \leq B \rightarrow \Gamma \vdash B$

-- lambda function and application

Lambda : $\forall \{\Gamma A B\} \rightarrow \Gamma, A \vdash B \rightarrow \Gamma \vdash A \Rightarrow B$

App : $\forall \{\Gamma A B\} \rightarrow \Gamma \vdash A \Rightarrow B \rightarrow \Gamma \vdash A \rightarrow \Gamma \vdash B$

-- command

Skip : $\forall \{\Gamma\} \rightarrow \Gamma \vdash \text{comm}$

Seq : $\forall \{\Gamma\} \rightarrow \Gamma \vdash \text{comm} \rightarrow \Gamma \vdash \text{comm} \rightarrow \Gamma \vdash \text{comm}$

NewVar : $\forall \{\Gamma\} \rightarrow \Gamma, \text{intvar} \vdash \text{comm} \rightarrow \Gamma \vdash \text{comm}$

Assign : $\forall \{\Gamma\} \rightarrow \Gamma \vdash \text{intacc} \rightarrow \Gamma \vdash \text{intexp} \rightarrow \Gamma \vdash \text{comm}$

-- intexp

Lit : $\forall \{\Gamma\} \rightarrow \mathbb{Z} \rightarrow \Gamma \vdash \text{intexp}$

Neg : $\forall \{\Gamma\} \rightarrow \Gamma \vdash \text{intexp} \rightarrow \Gamma \vdash \text{intexp}$

Plus : $\forall \{\Gamma\} \rightarrow \Gamma \vdash \text{intexp} \rightarrow \Gamma \vdash \text{intexp} \rightarrow \Gamma \vdash \text{intexp}$

-- Operational semantics

data Value : $\forall \{\Gamma A\} \rightarrow \Gamma \vdash A \rightarrow$ Set where

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

V-Lit : $\forall \{\Gamma\} \{i : \mathbb{Z}\} \rightarrow \text{Value} (\text{Lit} \{\Gamma\} i)$

V-Skip : $\forall \{\Gamma\} \rightarrow \text{Value} (\text{Skip} \{\Gamma\})$

-- Renaming

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

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

ext ρ Zero = Zero

ext ρ (Suc x) = Suc (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 (Var a) = Var (ρ a)

rename ρ (Lambda f) = Lambda (rename (ext ρ) f)

rename ρ (Sub a A \leq :B) = Sub (rename ρ a) A \leq :B

rename ρ (App f e) = App (rename ρ f) (rename ρ e)

rename ρ Skip = Skip

rename ρ (Seq c₁ c₂) = Seq (rename ρ c₁) (rename ρ c₂)

rename ρ (NewVar c) = NewVar (rename (ext ρ) c)

rename ρ (Assign a e) = Assign (rename ρ a) (rename ρ e)

rename ρ (Lit i) = Lit i

rename ρ (Neg e) = Neg (rename ρ e)

rename ρ (Plus e₁ e₂) = Plus (rename ρ e₁) (rename ρ e₂)

-- 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 Zero = Var Zero

exts σ (Suc x) = rename Suc (σ 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 (Var a) = σ a

subst σ (Sub a A \leq :B) = Sub (subst σ a) A \leq :B

subst σ (Lambda f) = Lambda (subst (exts σ) f)

subst σ (App f e) = App (subst σ f) (subst σ e)

subst σ Skip = Skip

subst σ (Seq c₁ c₂) = Seq (subst σ c₁) (subst σ c₂)

subst σ (NewVar c) = NewVar (subst (exts σ) c)

subst σ (Assign a e) = Assign (subst σ a) (subst σ e)

subst σ (Lit i) = Lit i

subst σ (Neg e) = Neg (subst σ e)

subst σ (Plus e₁ e₂) = Plus (subst σ e₁) (subst σ e₂)

-- 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

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

σ Zero = M

σ (Suc x) = Var x

-- Reduction

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

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

$\rightarrow F \longrightarrow F' \rightarrow \text{App} F E \longrightarrow \text{App} F' E$

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

$\rightarrow \text{Value} V \rightarrow E \longrightarrow E' \rightarrow \text{App} V E \longrightarrow \text{App} V E'$

Lambda- β : $\forall \{\Gamma A B\} \{F : \Gamma, A \vdash B\} \{V : \Gamma \vdash A\}$

$\rightarrow \text{Value} V \rightarrow \text{App} (\text{Lambda} F) V \longrightarrow F [V]$