

0.1 Terms

0.1.1 Value Terms

$$\begin{aligned}
 v ::= & x \\
 & | \lambda x : A. C \\
 & | \mathbf{c}^A \\
 & | () \\
 & | \mathbf{true} \mid \mathbf{false}
 \end{aligned} \tag{1}$$

0.1.2 Computation Terms

$$\begin{aligned}
 C ::= & \mathbf{if}_{\epsilon, A} \quad v \quad \mathbf{then} \quad C_1 \quad \mathbf{else} \quad C_2 \\
 & | v_1 \quad v_2 \\
 & | \mathbf{do} \quad x \leftarrow C_1 \quad \mathbf{in} \quad C_2 \\
 & | \mathbf{return} v
 \end{aligned} \tag{2}$$

0.2 Type System

0.2.1 Types

Ground Types There exists a set γ of ground types, including `Unit`, `Bool`

Value Types

$$A, B, C ::= \gamma \mid A \rightarrow \mathbf{M}_\epsilon B$$

Computation Types Computation types are of the form $\mathbf{M}_\epsilon A$

0.2.2 Sub-typing

There exists a sub-typing pre-order relation \leq_γ over ground types that is:

- (Reflexive) $\frac{}{A \leq_\gamma A}$
- (Transitive) $\frac{A \leq_\gamma B \quad B \leq_\gamma C}{A \leq_\gamma C}$

We extend this relation with the function sub-typing rule to yield the full sub-typing relation \leq :

- (ground) $\frac{A \leq_\gamma B}{A \leq B}$
- (Fn) $\frac{A \leq A' \quad B' \leq B \quad \epsilon \leq \epsilon'}{A' \rightarrow \mathbf{M}_{\epsilon'} B' \leq A \rightarrow \mathbf{M}_\epsilon B}$

0.2.3 Type Environments

An environment, $G ::= \diamond \mid \Gamma, x : A$

Domain Function

- $\text{dom}(\diamond) = \emptyset$
- $\text{dom}(\Gamma, x : A) = \text{dom}(\Gamma) \cup \{x\}$

Ok Predicate

- (Atom) $\frac{}{\text{Ok}}$
- (Var) $\frac{\text{Ok} \not\in \text{dom}(\Gamma)}{\Gamma, x:A \text{Ok}}$

0.2.4 Type Rules

Value Typing Rules

- (Const) $\frac{\text{Ok}}{\Gamma \vdash C^A : A}$
- (Unit) $\frac{\text{Ok}}{\Gamma \vdash () : \text{Unit}}$
- (True) $\frac{\text{Ok}}{\Gamma \vdash \text{true} : \text{Bool}}$
- (False) $\frac{\text{Ok}}{\Gamma \vdash \text{false} : \text{Bool}}$
- (Var) $\frac{\Gamma, x:A \text{Ok}}{\Gamma, x:A \vdash X : A}$
- (Weaken) $\frac{\Gamma \vdash x:A}{\Gamma, y:B \vdash X:A}$ (if $x \neq y$)
- (Fn) $\frac{\Gamma, x:A \vdash C : \mathbb{M}_\epsilon B}{\Gamma \vdash \lambda x:A. C : A \rightarrow \mathbb{M}_\epsilon B}$
- (Sub) $\frac{\Gamma \vdash v : A \leq B}{\Gamma \vdash v : B}$

Computation typing rules

- (Return) $\frac{\Gamma \vdash v : A}{\Gamma \vdash \text{return } v : \mathbb{M}_1 A}$
- (Apply) $\frac{\Gamma \vdash v_1 : A \rightarrow \mathbb{M}_\epsilon B \quad \Gamma \vdash v_2 : A}{\Gamma \vdash v_1 v_2 : \mathbb{M}_\epsilon B}$
- (if) $\frac{\Gamma \vdash v : \text{Bool} \quad \Gamma \vdash C_1 : \mathbb{M}_{\epsilon_1} A \quad \Gamma \vdash C_2 : \mathbb{M}_{\epsilon_2} A}{\Gamma \vdash \text{if}_{\epsilon, A} v \text{ then } C_1 \text{ else } C_2 : \mathbb{M}_\epsilon A}$
- (Do) $\frac{\Gamma \vdash C_1 : \mathbb{M}_{\epsilon_1} A \quad \Gamma, x:A \vdash C_2 : \mathbb{M}_{\epsilon_2} B}{\Gamma \vdash \text{do } x \leftarrow C_1 \text{ in } C_2 : \mathbb{M}_{\epsilon_1 \cdot \epsilon_2} B}$
- (Subeffect) $\frac{\Gamma \vdash C : \mathbb{M}_{\epsilon_1} A \leq B \quad \epsilon_1 \leq \epsilon_2}{\Gamma \vdash C : \mathbb{M}_{\epsilon_2} B}$

0.2.5 Ok Lemma

If $\Gamma \vdash t : \tau$ then ΓOk .

Proof If $\Gamma, x : A \text{Ok}$ then by inversion ΓOk . Only the type rule **Weaken** adds terms to the environment from its preconditions to its post-condition and it does so in an **Ok** preserving way. Any type derivation tree has at least one leaf. All leaves are axioms which require ΓOk . And all non-axiom derivations preserve the **Ok** property.