0.1 Terms

0.1.1 Value Terms

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

0.1.2 Computation Terms

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

0.2 Type System

0.2.1 Types

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

Value Types

$$A, B, C ::= \gamma \mid A \to \mathtt{M}_\epsilon B$$

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

0.2.2 Sub-typing

There exists a sub-typing pre-order relation $\leq:_\gamma$ over ground types that is:

- (Reflexive)$\frac{}{A \leq:_\gamma A}$
- (Transitive)$\frac{A \leq:_\gamma B \sigma \sigma B \leq:_\gamma C}{A \leq:_\gamma C}$

  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)$\frac{A \leq: A' \sigma \sigma B' \leq: B \sigma \sigma \epsilon \leq \epsilon'}{A' \to \mathtt{M}_{\epsilon'} B' \leq: A \to \mathtt{M}_\epsilon B}$

0.2.3 Type Environments

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

Domain Function

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

$\mathtt{Ok}$ Predicate

- (Atom) $\frac{}{\diamond\mathtt{Ok}}$
- (Var) $\frac{\Gamma\mathtt{Ok}\sigma\sigma x\notin\mathtt{dom}(\Gamma)}{\Gamma,x{:}A\mathtt{Ok}}$

0.2.4 Type Rules

Value Typing Rules

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

Computation typing rules

- (Return) $\frac{\Gamma\vdash v{:}A}{\Gamma\vdash\mathtt{return}v{:}\mathtt{M}_{\mathbf{1}}A}$
- (Apply) $\frac{\Gamma\vdash v_1{:}A\to\mathtt{M}_{\epsilon}B\sigma\sigma\Gamma\vdash v_2{:}A}{\Gamma\vdash v_1\sigma v_2{:}\mathtt{M}_{\epsilon}B}$
- (if) $\frac{\Gamma\vdash v{:}\mathtt{Bool}\sigma\sigma\Gamma\vdash C_1{:}\mathtt{M}_{\epsilon}A\sigma\sigma\Gamma\vdash C_2{:}\mathtt{M}_{\epsilon}A}{\Gamma\vdash\mathtt{if}_{\epsilon,A}\sigma V\sigma\mathtt{then}\sigma C_1\sigma\mathtt{else}\sigma C_2{:}\mathtt{M}_{\epsilon}A}$
- (Do) $\frac{\Gamma\vdash C_1{:}\mathtt{M}_{\epsilon_1}A\sigma\sigma\Gamma,x{:}A\vdash C_2{:}\mathtt{M}_{\epsilon_2}B}{\Gamma\vdash\mathtt{do}\sigma x\leftarrow C_1\sigma\mathtt{in}\sigma C_2{:}\mathtt{M}_{\epsilon_1\cdot\epsilon_2}B}$
- (Subeffect) $\frac{\Gamma\vdash C{:}\mathtt{M}_{\epsilon_1}A\sigma\sigma A\leq{:}B\sigma\sigma\epsilon_1\leq\epsilon_2}{\Gamma\vdash C{:}\mathtt{M}_{\epsilon_2}B}$

0.2.5 Ok Lemma

If $\Gamma\vdash t{:}\tau$ then $\Gamma\mathtt{Ok}$.

Proof If $\Gamma,x:A\mathtt{Ok}$ then by inversion $\Gamma\mathtt{Ok}$ 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 $\Gamma\mathtt{Ok}$. And all non-axiom derivations preserve the `Ok` property.