

0.1 Denotations of Types

0.1.1 Denotation of Type Environments

Given a function $\llbracket _ \rrbracket_M$ mapping types to objects in the category \mathbb{C} , we can define the denotation of an Ok type environment Γ .

$$\begin{aligned}\llbracket \diamond \rrbracket_M &= 1 \\ \llbracket \Gamma, x : A \rrbracket_M &= (\llbracket \Gamma \rrbracket_M \times \llbracket A \rrbracket_M)\end{aligned}$$

For ease of notation, and since we normally only talk about one denotation function at a time, I shall typically drop the denotation notation when talking about the denotation of value types and type environments. Hence,

$$\llbracket \Gamma, x : A \rrbracket_M = \Gamma \times A$$

0.1.2 Denotation of Computation Type

Given a function $\llbracket _ \rrbracket_M$ mapping value types to objects in the category \mathbb{C} , we write the denotation of Computation types $\mathbb{M}_\epsilon A$ as so:

$$\llbracket \mathbb{M}_\epsilon A \rrbracket_M = T_\epsilon \llbracket A \rrbracket_M$$

Since we can infer the denotation function, we can include it implicitly and drop the denotation sign.

$$\llbracket \mathbb{M}_\epsilon A \rrbracket_M = T_\epsilon A$$

0.1.3 Denotation of Function Types

Given a function $\llbracket _ \rrbracket_M$ mapping types to objects in the category \mathbb{C} , we write the denotation of a function type $A \rightarrow \mathbb{M}_\epsilon B$ as so:

$$\llbracket A \rightarrow \mathbb{M}_\epsilon B \rrbracket_M = (T_\epsilon \llbracket B \rrbracket_M)^{\llbracket A \rrbracket_M}$$

Again, since we can infer the denotation function, Let us drop the notation.

$$\llbracket A \rightarrow \mathbb{M}_\epsilon B \rrbracket_M = (T_\epsilon B)^A$$

0.2 Denotation of Terms

Given the denotation of types and typing environments, we can now define denotations of well typed terms.

$$\llbracket \Gamma \vdash t : \tau \rrbracket_M : \Gamma \rightarrow \llbracket \tau \rrbracket_M$$

Denotations are defined recursively over the typing derivation of a term. Hence, they implicitly depend on the exact derivation used. Since, as proven in the chapter on the uniqueness of derivations, the denotations of all type derivations yielding the same type relation $\Gamma \vdash t : \tau$ are equal, we need not refer to the derivation that yielded each denotation.

0.2.1 Denotation of Value Terms

- (Unit) $\frac{\text{rOk}}{\llbracket \Gamma \vdash () : \text{Unit} \rrbracket_M = \llbracket () \rrbracket_M \circ \langle \rangle_\Gamma : \Gamma \rightarrow \llbracket \text{Unit} \rrbracket_M}$
- (Const) $\frac{\text{rOk}}{\llbracket \Gamma \vdash \mathbf{C}^A : A \rrbracket_M = \llbracket \mathbf{C}^A \rrbracket_M \circ \langle \rangle_\Gamma : \Gamma \rightarrow \llbracket A \rrbracket_M}$
- (True) $\frac{\text{rOk}}{\llbracket \Gamma \vdash \text{true} : \text{Bool} \rrbracket_M = \llbracket \text{true} \rrbracket_M \circ \langle \rangle_\Gamma : \Gamma \rightarrow \llbracket \text{Bool} \rrbracket_M}$

- (False) $\frac{\text{rOk}}{\llbracket \Gamma \vdash \text{false} : \text{Bool} \rrbracket_M = \llbracket \text{false} \rrbracket_M \circ \langle \rangle_\Gamma : \Gamma \rightarrow \llbracket \text{Bool} \rrbracket_M}$
- (Var) $\frac{\text{rOk}}{\llbracket \Gamma, x : A \vdash x : A \rrbracket_M = \pi_2 : \Gamma \times A \rightarrow A}$
- (Weaken) $\frac{f = \llbracket \Gamma \vdash x : A \rrbracket_M : \Gamma \rightarrow A}{\llbracket \Gamma, y : B \vdash x : A \rrbracket_M = f \circ \pi_1 : \Gamma \times B \rightarrow A}$
- (Lambda) $\frac{f = \llbracket \Gamma, x : A \rrbracket_M \text{CM}_\epsilon B : \Gamma \times A \rightarrow T_\epsilon B}{\llbracket \Gamma \vdash \lambda x : A. C : A \rightarrow \mathbf{M}_\epsilon B \rrbracket_M = \text{cur}(f) : \Gamma \rightarrow (T_\epsilon B)^A}$
- (Subtype) $\frac{f = \llbracket \Gamma \vdash v : A \rrbracket_M : \Gamma \rightarrow A \text{ and } g = \llbracket A \leq B \rrbracket_M}{\llbracket \Gamma \vdash v : B \rrbracket_M = g \circ f : \Gamma \rightarrow B}$

0.2.2 Denotation of Computation Terms

- (Return) $\frac{f = \llbracket \Gamma \vdash v : A \rrbracket_M}{\llbracket \Gamma \vdash \text{return } v : \mathbf{M}_1 A \rrbracket_M = \eta_A \circ f}$
- (If) $\frac{f = \llbracket \Gamma \vdash v : \text{Bool} \rrbracket_M, g = \llbracket \Gamma \vdash C_1 : \mathbf{M}_\epsilon A \rrbracket_M, h = \llbracket \Gamma \vdash C_2 : \mathbf{M}_\epsilon A \rrbracket_M}{\llbracket \Gamma \vdash \text{if}_{\epsilon, A} v \text{ then } C_1 \text{ else } C_2 : \mathbf{M}_\epsilon A \rrbracket_M = \text{If}_{\mathbf{M}_\epsilon B} \circ \langle f, \langle g, h \rangle \rangle : \Gamma \rightarrow T_\epsilon A}$
- (Bind) $\frac{f = \llbracket \Gamma \vdash C_1 : \mathbf{M}_{\epsilon_1} A : \Gamma \rightarrow T_{\epsilon_1} A \rrbracket_M, g = \llbracket \Gamma, x : A \vdash C_2 : \mathbf{M}_{\epsilon_2} B \rrbracket_M : \Gamma \times A \rightarrow T_{\epsilon_2} B}{\llbracket \Gamma \vdash \text{do } x \leftarrow C_1 \text{ in } C_2 : \mathbf{M}_{\epsilon_1 \cdot \epsilon_2} B \rrbracket_M = \mu_{\epsilon_1, \epsilon_2, B} \circ T_{\epsilon_1} g \circ \text{t}_{\Gamma, A, \epsilon_1} \circ \langle \text{Id}_\Gamma, f \rangle : \Gamma \rightarrow T_{\epsilon_1 \cdot \epsilon_2} B}$
- (Subeffect) $\frac{f = \llbracket \Gamma \vdash c : \mathbf{M}_{\epsilon_1} A \rrbracket_M : \Gamma \rightarrow T_{\epsilon_1} A, g = \llbracket A \leq B \rrbracket_M, h = \llbracket \epsilon_1 \leq \epsilon_2 \rrbracket}{\llbracket \Gamma \vdash C : \mathbf{M}_{\epsilon_2} B \rrbracket_M = h_B \circ T_{\epsilon_1} g \circ f}$
- (Apply) $\frac{f = \llbracket \Gamma \vdash v_1 : A \rightarrow \mathbf{M}_\epsilon B \rrbracket_M : \Gamma \rightarrow (T_\epsilon B)^A, g = \llbracket \Gamma \vdash v_2 : A \rrbracket_M : \Gamma \rightarrow A}{\llbracket \Gamma \vdash v_1 v_2 : \mathbf{M}_\epsilon B \rrbracket_M = \text{app} \circ \langle f, g \rangle : \Gamma \rightarrow T_\epsilon B}$