1 Semantics

1.1 Grammar and definitions

Value expressions

$$\begin{aligned} V \rightarrow & \ C^{\gamma} \text{ Typed contants} \\ | & \ () \text{ Unit constructor} \\ | & \ \lambda x : \tau . M \text{ term lambda expressions} \\ | & \ x \text{ Variables} \\ | & \ \Lambda \phi . V \text{ Effect lambda expressions} \\ | & \ V \epsilon \text{ Effect specialisation} \end{aligned} \tag{1}$$

Computation expressions todo: is the if rule necessary?

$$\begin{aligned} M \rightarrow & \ V \ V \text{Application} \\ | & \ do \ x \ \leftarrow \ M \ in \ M \text{ Sequencing} \\ | & \ return \ V \text{ Unit computation} \\ | & \ if \ V \ then \ C \ else \ C \text{ Conditional} \\ | & \ Cons^{\tau_1 \rightarrow T_{\xi} \tau_2}(V) \text{ Computation Constructors} \\ | & \ Op^{T_{\xi} \tau_1 \rightarrow T_{\xi} \tau_2}(M) \text{ Computation Combinators} \end{aligned} \tag{2}$$

Value types

$$\begin{aligned} \tau \rightarrow & \ \gamma \text{ Ground Types} \\ | & \ \tau \rightarrow \Theta \text{ Functions are kleisli arrows} \\ | & \ \forall \phi . \tau \text{ Effect quantification} \end{aligned} \tag{3}$$

Computation (Monad) types

$$\Theta \rightarrow T_{\epsilon} \tau \text{ Monad type constructor} \tag{4}$$

Effects

$$\epsilon \rightarrow \phi \mid \xi .. \text{ Effect variables or ground effects}$$

1.2 typing

Value typing rules

$$\text{(Const)}\frac{\Gamma\ OK}{\Gamma \vdash C^{\gamma}\colon \gamma}$$

$$\text{(Unit)}\frac{\Gamma\ OK}{\Gamma \vdash ()\colon unit}$$

$$\text{(Weaken)}\frac{\Gamma \vdash x\colon A}{\Gamma, x' : A' \vdash x\colon A}(x \neq x')$$

$$\text{(Var)}\frac{\Gamma\ OK}{\Gamma, x : A \vdash x\colon A}$$

$$\text{(Eff-Gen)}\frac{\Gamma \vdash V\colon A}{\Gamma \vdash \Lambda\phi.V\colon \forall\phi.A}\phi \notin fev(\Gamma)$$

$$\text{(Eff-Spec)}\frac{\Gamma \vdash V\colon \forall\phi.A}{\Gamma \vdash V\epsilon\colon A[\epsilon\phi]}$$

$$\text{(Lambda)}\frac{\Gamma, x : A \vdash M\colon \Theta}{\Gamma \vdash \lambda x : A.M\colon A \to \Theta}x \notin fv(\Gamma)$$

$$\text{(SubType)}\frac{\Gamma \vdash V\colon A}{\Gamma \vdash V\colon B}A \leq B$$

Computation Typing Rules

$$\text{(Application)}\frac{\Gamma \vdash V_1\colon A \to \Theta\ \Gamma \vdash V_2\colon A}{\Gamma \vdash V_1 V_2\colon \Theta}$$

$$\text{(Sequencing)}\frac{\Gamma \vdash M_1\colon T_{\epsilon_1} A\ \Gamma, x : A \vdash M_2\colon T_{\epsilon_2} B}{\Gamma \vdash do\ x\ \leftarrow\ M_1\ in\ M_2\colon T_{\epsilon_1 \circ \epsilon_2} B}$$

$$\text{(Unit Computation)}\frac{\Gamma \vdash V\colon A}{\Gamma \vdash return(V)\colon T_1 A}$$

$$\text{(Constructor)}\frac{\Gamma \vdash V\colon \tau_1}{\Gamma \vdash Cons^{\tau_1 \to T_\xi \tau_2}(V)\colon T_\xi \tau_2}$$

$$\text{(Combinator)}\frac{\Gamma \vdash M\colon T_{\xi_1}\tau_1}{\Gamma \vdash Op^{T_\xi \tau_1 \to T_\xi \tau_2}(M)\colon T_{\xi_2}\tau_2}$$

$$\text{(if)}\frac{\Gamma \vdash V\colon Bool\ \Gamma \vdash C_1\colon \Theta\ \Gamma \vdash C_2\colon \Theta}{\Gamma \vdash if\ V\ then\ C_1\ else\ C_2\colon \Theta}$$

$$\text{(SubTypeC)}\frac{\Gamma \vdash M\colon \Theta_1}{\Gamma \vdash M\colon \Theta_2}\Theta_2 \leq \Theta_1$$

Subtyping Rules A ranges over value types

$$\text{(Reflexive)}\frac{}{A \leq A}$$

$$\text{(Transitive)}\frac{A \leq B \;\; B \leq C}{A \leq C}$$

$$\text{(Function)}\frac{A' \leq A \quad \Theta \leq \Theta'}{A \to \Theta \leq A' \to \Theta'}$$

$$\text{(Quantification)}\frac{\tau_1 \leq \tau_2}{\Lambda\phi.\tau_1 \leq \Lambda\phi.\tau_2}$$

$$\text{(Computation)}\frac{\epsilon_1 \sqsubseteq \epsilon_2 \quad A \leq B}{T_{\epsilon_1} A \leq T_{\epsilon_2} B}$$