1 Semantics

$$\frac{}{(\textbf{if true then } c_1 \textbf{ else } c_2) \rightsquigarrow c_1}\ (\textsc{If-True}) \qquad \frac{}{(\textbf{if false then } c_1 \textbf{ else } c_2) \rightsquigarrow c_2}\ (\textsc{If-False})$$

$$\frac{}{(\textbf{fun } x \mapsto c)\ e \rightsquigarrow c[e/x]}\ (\textsc{Fun-App})$$

$$\frac{}{(\textbf{match } (\textbf{Left } e) \textbf{ with Left } x \mapsto c_1 \ \parallel\ \textbf{Right } x \mapsto c_2) \rightsquigarrow c_1[e/x]}\ (\textsc{Match-Left})$$

$$\frac{}{(\textbf{match } (\textbf{Right } e) \textbf{ with Left } x \mapsto c_1 \ \parallel\ \textbf{Right } x \mapsto c_2) \rightsquigarrow c_2[e/x]}\ (\textsc{Match-Right})$$

$$\frac{}{(\textbf{match } (e_1, e_2) \textbf{ with } (f, s) \mapsto c) \rightsquigarrow c[e_1/f, e_2/s]}\ (\textsc{Match-Prod})$$

$$\frac{c_1 \rightsquigarrow c_1'}{\textbf{let } x = c_1 \textbf{ in } c_2 \rightsquigarrow \textbf{let } x = c_1' \textbf{ in } c_2}\ (\textsc{Let-Step}) \qquad \frac{}{\textbf{let } x = (\textbf{val } e) \textbf{ in } c \rightsquigarrow c[e/x]}\ (\textsc{Let-Val})$$

$$\frac{}{\textbf{let } x = E(e, y.c_1) \textbf{ in } c_2 \rightsquigarrow E(e, y.\textbf{let } x = c_1 \textbf{ in } c_2)}\ (\textsc{Let-Effect})$$

$$\frac{}{(\textbf{let rec } f\ x = c_1 \textbf{ in } c_2) \rightsquigarrow c_2[(\textbf{fun } x \mapsto \textbf{let rec } f\ x = c_1 \textbf{ in } c_1)/f]}\ (\textsc{Let-Rec})$$

$$\frac{\kappa \text{ is current delimited continuation}}{\textbf{perform } (E\ e) \rightsquigarrow E(e, \kappa)}\ (\textsc{Perform})$$

$$\frac{c \rightsquigarrow c'}{\textbf{with } e \textbf{ handle } c \rightsquigarrow \textbf{with } e \textbf{ handle } c'}\ (\textsc{Handle-Step})$$

$$\frac{h \stackrel{\text{def}}{=} (\textbf{handler val } x \mapsto c_v \ \parallel\ \dots)}{\textbf{with } h \textbf{ handle } (\textbf{val } e) \rightsquigarrow c_v[e/x]}\ (\textsc{Handle-Val})$$

$$\frac{h \stackrel{\text{def}}{=} (\textbf{handler } \dots \overline{\parallel\ \textbf{effect } E_i\ x\ k \mapsto c_i} \ \parallel\ \dots) \qquad \exists i. E_i = E}{\textbf{with } h \textbf{ handle } E(e, y.c) \rightsquigarrow c_i[e/x, (y.c)/k]}\ (\textsc{Handle-Eff-Match})$$

$$\frac{h \stackrel{\text{def}}{=} (\textbf{handler } \dots \overline{\parallel\ \textbf{effect } E_i\ x\ k \mapsto c_i} \ \parallel\ \dots) \qquad \forall i. E_i \neq E}{\textbf{with } h \textbf{ handle } E(e, y.c) \rightsquigarrow E(e, y.\textbf{with } h \textbf{ handle } c)}\ (\textsc{Handle-Eff-Rise})$$

Figure 1: Small step operational semantics of Eff

2 Typing

$$\frac{x : A \in \Gamma}{\Gamma \vdash_e x : A}\ (\text{T-}\textsc{Var}) \qquad \frac{}{\Gamma \vdash_e \textbf{true} : \texttt{bool}}\ (\text{T-}\textsc{True}) \qquad \frac{}{\Gamma \vdash_e \textbf{false} : \texttt{bool}}\ (\text{T-}\textsc{False})$$

$$\frac{}{\Gamma \vdash_e () : \texttt{unit}}\ (\text{T-}\textsc{Unit}) \qquad \frac{\Gamma \vdash_e e_1 : A \qquad \Gamma \vdash_e e_2 : B}{\Gamma \vdash_e (e_1, e_2) : A * B}\ (\text{T-}\textsc{Pair})$$

$$\frac{\Gamma \vdash_e e : A}{\Gamma \vdash_e \textbf{Left } e : A + B}\ (\text{T-}\textsc{SumLeft}) \qquad \frac{\Gamma \vdash_e e : B}{\Gamma \vdash_e \textbf{Right } e : A + B}\ (\text{T-}\textsc{SumRight})$$

$$\frac{x : A, \Gamma \vdash_c c : C}{\Gamma \vdash_e (\textbf{fun } x \mapsto c) : A \to C}\ (\text{T-}\textsc{Fun})$$

$$\frac{x : A, \Gamma \vdash_c c_v : C \quad \forall i.\ e_i : A_i, k_i : B_i, \Gamma \vdash_c c_i : C \quad x : C, \Gamma \vdash_c c_f : D}{\Gamma \vdash_e h : A \Rightarrow D}\ (\textsc{T-Handler})$$

where $h \stackrel{\text{def}}{=} (\textbf{handler val}\ x \mapsto c_v\ \overline{\|\ \textbf{effect}\ E_i\ e_i\ k_i \mapsto c_i}\ \|\ \textbf{finally}\ x \mapsto c_f)$ and $\forall i.\ E_i : A_i \twoheadrightarrow B_i \in \Sigma_E$

$$\frac{\Gamma \vdash_e e : A}{\Gamma \vdash_c \textbf{val}\ e : A}\ (\textsc{T-Val}) \qquad \frac{\Gamma \vdash_e e : \texttt{empty}}{\Gamma \vdash_c \textbf{absurd}\ e : A}\ (\textsc{T-Absurd})$$

$$\frac{\Gamma \vdash_c c_1 : A \quad x : A, \Gamma \vdash_c c_2 : B}{\Gamma \vdash_c (\textbf{let}\ x = c_1\ \textbf{in}\ c_2) : B}\ (\textsc{T-Let})$$

$$\frac{x : A, f : A \to B \Gamma \vdash_c c_1 : B \quad f : A \to B, \Gamma \vdash_c c_2 : C}{\Gamma \vdash_c (\textbf{let rec}\ f\ x = c_1\ \textbf{in}\ c_2) : C}\ (\textsc{T-LetRec})$$

$$\frac{\Gamma \vdash_e e_1 : A \to B \quad \Gamma \vdash_e e_2 : A}{\Gamma \vdash_c e_1\ e_2 : B}\ (\textsc{T-FunApp})$$

$$\frac{\Gamma \vdash_e e : \texttt{bool} \quad \Gamma \vdash_c c_1 : A \quad \Gamma \vdash_c c_2 : A}{\Gamma \vdash_c \textbf{if}\ e\ \textbf{then}\ c_1\ \textbf{else}\ c_2 : A}\ (\textsc{T-If})$$

$$\frac{\Gamma \vdash_e e : A + B \quad x : A, \Gamma \vdash_c c_l : C \quad x : B, \Gamma \vdash_c c_r : C}{\textbf{match}\ e\ \textbf{with Left}\ x \mapsto c_l\ \|\ \textbf{Right}\ x \mapsto c_r : C}\ (\textsc{T-MatchSum})$$

$$\frac{\Gamma \vdash_e e : A * B \quad f : A, s : B, \Gamma \vdash_c c_l : C}{\textbf{match}\ e\ \textbf{with}\ (f, s) \mapsto c : C}\ (\textsc{T-MatchProd})$$

$$\frac{E : A \twoheadrightarrow B \in \Sigma_E \quad \Gamma \vdash_e e : A}{\textbf{perform}\ (E\ e) : B}\ (\textsc{T-Perform}) \quad \frac{\Gamma \vdash_e e : A \Rightarrow B \quad \Gamma \vdash_c c : A}{\textbf{with}\ e\ \textbf{handle}\ c : B}\ (\textsc{T-WithHandle})$$

Figure 2: All typing rules for Eff