| | | | |
|---|---|---|---|
| $x$ | term variable | | |
| $\alpha$ | type variable | | |
| $T$ | type constructor | | |

| | | | |
|---|---|---|---|
| $t$ | $::=$ | | term |
| | $\|$ | $x$ | variable |
| | $\|$ | $\lambda(x : \tau) \Rightarrow t$ | abstraction |
| | $\|$ | $\lambda\{\alpha : \kappa\} \Rightarrow t$ | type abstraction |
| | $\|$ | $t_1\, t_2$ | application |
| | $\|$ | $t\ [\tau]$ | type application |
| | $\|$ | $t : \tau$ | type annotation |
| $\kappa$ | $::=$ | | kind |
| | $\|$ | $*$ | star |
| | $\|$ | $\kappa_1 \to \kappa_2$ | kind arrow |
| $\tau$ | $::=$ | | type |
| | $\|$ | $\alpha$ | type variable |
| | $\|$ | $T$ | type constructor |
| | $\|$ | $\tau_1 \to \tau_2$ | $\equiv (\to)\ \tau_1\ \tau_2$ |
| | $\|$ | $\lambda(\alpha : \kappa), \tau$ | operator abstraction |
| | $\|$ | $\forall\,(\alpha : \kappa), \tau$ | universal quantification |
| | $\|$ | $\tau_1\, \tau_2$ | operator application |
| $\Gamma$ | $::=$ | | typing environment |
| | $\|$ | $\emptyset$ | empty |
| | $\|$ | $\Gamma, x : \tau$ | variable |
| | $\|$ | $\Gamma, T : \kappa$ | type constructor |
| | $\|$ | $\Gamma, \alpha : \kappa$ | type variable |
| $tv$ | $::=$ | | typed value |
| | $\|$ | $\lambda(x : \tau) \Rightarrow t$ | abstraction |
| | $\|$ | $\lambda\{\alpha : \kappa\} \Rightarrow tv$ | type abstraction |
| | $\|$ | $tv\ [\tau]$ | type application |
| | $\|$ | $tv : \tau$ | type annotation |
| $v$ | $::=$ | | value |
| | $\|$ | $\lambda x \Rightarrow t$ | abstraction |

---

Initial environment: $\Gamma = \emptyset, (\to) : * \to * \to *$

---

$\boxed{[\alpha \mapsto \tau_1]\ \tau_2 \rhd \tau_3}$ Type substitution

$$\frac{}{[\alpha \mapsto \tau]\ \alpha \rhd \tau}\ \textsc{SubstT\_Var1}$$

$$\frac{\alpha_1 \neq \alpha_2}{[\alpha_1 \mapsto \tau]\ \alpha_2 \rhd \alpha_2}\ \textsc{SubstT\_Var2}$$

$$\frac{}{[\alpha \mapsto \tau]\ T \rhd T}\ \textsc{SubstT\_Type}$$

$$\frac{[\alpha_1 \mapsto \tau_1]\ \tau_2 \triangleright \tau_2'}{[\alpha_1 \mapsto \tau_1]\ (\lambda(\alpha_2 : \kappa), \tau_2) \triangleright (\lambda(\alpha_2 : \kappa), \tau_2')} \quad \textsc{SubstT\_Abs}$$

$$\frac{[\alpha_1 \mapsto \tau_1]\ \tau_2 \triangleright \tau_2'}{[\alpha_1 \mapsto \tau_1]\ (\forall\,(\alpha_2 : \kappa), \tau_2) \triangleright (\forall\,(\alpha_2 : \kappa), \tau_2')} \quad \textsc{SubstT\_Forall}$$

$$\frac{\begin{array}{l}[\alpha \mapsto \tau_1]\ \tau_2 \triangleright \tau_2' \\ [\alpha \mapsto \tau_1]\ \tau_3 \triangleright \tau_3'\end{array}}{[\alpha \mapsto \tau_1]\ \tau_2\, \tau_3 \triangleright \tau_2'\, \tau_3'} \quad \textsc{SubstT\_App}$$

$\boxed{\Gamma \vdash t : \tau}$ Typing rules

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \quad \textsc{T\_Var}$$

$$\frac{\begin{array}{l}\Gamma, x : \tau_1 \vdash t : \tau_2 \\ \Gamma \vdash \tau_1 : *\end{array}}{\Gamma \vdash (\lambda(x : \tau_1) \Rightarrow t) : \tau_1 \to \tau_2} \quad \textsc{T\_Abs}$$

$$\frac{\begin{array}{l}\Gamma, \alpha : \kappa \vdash t : \tau \\ \alpha \notin \Gamma\end{array}}{\Gamma \vdash (\lambda\{\alpha : \kappa\} \Rightarrow t) : \forall\,(\alpha : \kappa), \tau} \quad \textsc{T\_TyAbs}$$

$$\frac{\begin{array}{l}\Gamma \vdash t_1 : \tau_2 \to \tau_1 \\ \tau_2 \equiv \tau_2' \\ \Gamma \vdash t_2 : \tau_2'\end{array}}{\Gamma \vdash t_1\ t_2 : \tau_1} \quad \textsc{T\_App}$$

$$\frac{\begin{array}{l}\Gamma \vdash t : \forall\,(\alpha : \kappa), \tau_2 \\ \Gamma \vdash \tau_1 : \kappa \\ [\alpha \mapsto \tau_1]\ \tau_2 \triangleright \tau_2'\end{array}}{\Gamma \vdash t\ [\tau_1] : \tau_2'} \quad \textsc{T\_TyApp}$$

$$\frac{\begin{array}{l}\Gamma \vdash t : \tau_2 \\ \tau_1 \equiv \tau_2\end{array}}{\Gamma \vdash (t : \tau_1) : \tau_1} \quad \textsc{T\_Annot}$$

$\boxed{\Gamma \vdash \tau : \kappa}$ Kinding rules

$$\frac{\alpha : \kappa \in \Gamma}{\Gamma \vdash \alpha : \kappa} \quad \textsc{K\_Var}$$

$$\frac{T : \kappa \in \Gamma}{\Gamma \vdash T : \kappa} \quad \textsc{K\_TypeConstr}$$

$$\frac{\Gamma, \alpha : \kappa_1 \vdash \tau : \kappa_2}{\Gamma \vdash (\lambda(\alpha : \kappa_1), \tau) : \kappa_1 \to \kappa_2} \quad \textsc{K\_Abs}$$

$$\frac{\begin{array}{l}\Gamma \vdash \tau_1 : \kappa_2 \to \kappa_1 \\ \Gamma \vdash \tau_2 : \kappa_2\end{array}}{\Gamma \vdash \tau_1\, \tau_2 : \kappa_1} \quad \textsc{K\_App}$$

$$\frac{\Gamma, \alpha : \kappa \vdash \tau : *}{\Gamma \vdash (\forall\,(\alpha : \kappa), \tau) : *} \quad \textsc{K\_Forall}$$

$\boxed{\tau_1 \equiv \tau_2}$ Type equivalence

$$\frac{}{\tau \equiv \tau} \quad \textsc{EQ\_Refl}$$

$$\frac{\tau_2 \equiv \tau_1}{\tau_1 \equiv \tau_2} \quad \text{EQ\_SYM}$$

$$\frac{\begin{array}{l}\tau_1 \equiv \tau_2 \\ \tau_2 \equiv \tau_3\end{array}}{\tau_1 \equiv \tau_3} \quad \text{EQ\_TRANS}$$

$$\frac{}{\alpha \equiv \alpha} \quad \text{EQ\_VAR}$$

$$\frac{}{T \equiv T} \quad \text{EQ\_TYPECONSTR}$$

$$\frac{\tau_1 \equiv \tau_2}{\forall\,(\alpha : \kappa), \tau_1 \equiv \forall\,(\alpha : \kappa), \tau_2} \quad \text{EQ\_FORALL}$$

$$\frac{\tau_1 \equiv \tau_2}{\lambda(\alpha : \kappa), \tau_1 \equiv \lambda(\alpha : \kappa), \tau_2} \quad \text{EQ\_ABS}$$

$$\frac{\begin{array}{l}\tau_1 \equiv \tau_1' \\ \tau_2 \equiv \tau_2'\end{array}}{\tau_1\ \tau_2 \equiv \tau_1'\ \tau_2'} \quad \text{EQ\_APP}$$

$$\frac{[\alpha \mapsto \tau_2]\ \tau_1 \rhd \tau_1'}{(\lambda(\alpha : \kappa), \tau_1)\ \tau_2 \equiv \tau_1'} \quad \text{EQ\_APPABS}$$

$\boxed{[x \mapsto tv]\ t_1 \rhd t_2}$ substitution

$$\frac{}{[x \mapsto tv]\ x \rhd tv} \quad \text{SUBST\_VAR1}$$

$$\frac{x_1 \neq x_2}{[x_1 \mapsto tv]\ x_2 \rhd x_2} \quad \text{SUBST\_VAR2}$$

$$\frac{}{[x \mapsto tv]\ (\lambda(x : \tau) \Rightarrow t) \rhd (\lambda(x : \tau) \Rightarrow t)} \quad \text{SUBST\_ABS1}$$

$$\frac{\begin{array}{l}x_1 \neq x_2 \\ [x_1 \mapsto tv]\ t_1 \rhd t_2\end{array}}{[x_1 \mapsto tv]\ (\lambda(x_2 : \tau) \Rightarrow t_1) \rhd (\lambda(x_2 : \tau) \Rightarrow t_2)} \quad \text{SUBST\_ABS2}$$

$$\frac{[x \mapsto tv]\ t_1 \rhd t_2}{[x \mapsto tv]\ (\lambda\{\alpha : \kappa\} \Rightarrow t_1) \rhd (\lambda\{\alpha : \kappa\} \Rightarrow t_2)} \quad \text{SUBST\_TABS}$$

$$\frac{\begin{array}{l}[x \mapsto tv]\ t_1 \rhd t_1' \\ [x \mapsto tv]\ t_2 \rhd t_2'\end{array}}{[x \mapsto tv]\ t_1\ t_2 \rhd t_1'\ t_2'} \quad \text{SUBST\_APP}$$

$$\frac{[x \mapsto tv]\ t_1 \rhd t_2}{[x \mapsto tv]\ (t_1\ [\tau]) \rhd (t_2\ [\tau])} \quad \text{SUBST\_TAPP}$$

$$\frac{[x \mapsto tv]\ t_1 \rhd t_2}{[x \mapsto tv]\ (t_1 : \tau) \rhd (t_2 : \tau)} \quad \text{SUBST\_ANNOT}$$

$\boxed{[\alpha \mapsto \tau]\ t_1 \rhd t_2}$ substitution of type variable in term

$$\frac{}{[\alpha \mapsto \tau]\ x \rhd x} \quad \text{TTSUBST\_VAR}$$

$$\frac{\begin{array}{l}[\alpha \mapsto \tau_1]\ \tau_2 \rhd \tau_2' \\ [\alpha \mapsto \tau_1]\ t_1 \rhd t_2\end{array}}{[\alpha \mapsto \tau_1]\ (\lambda(x : \tau_2) \Rightarrow t_1) \rhd (\lambda(x : \tau_2') \Rightarrow t_2)} \quad \text{TTSUBST\_ABS}$$

$$\frac{[\alpha_1 \mapsto \tau]\ t_1 \rhd t_2}{[\alpha_1 \mapsto \tau]\ (\lambda\{\alpha_2 : \kappa\} \Rightarrow t_1) \rhd (\lambda\{\alpha_2 : \kappa\} \Rightarrow t_2)} \quad \textsc{TtSubst\_TAbs}$$

$$\frac{\begin{array}{l}[\alpha \mapsto \tau]\ t_1 \rhd t_1' \\ [\alpha \mapsto \tau]\ t_2 \rhd t_2'\end{array}}{[\alpha \mapsto \tau]\ t_1\ t_2 \rhd t_1'\ t_2'} \quad \textsc{TtSubst\_App}$$

$$\frac{\begin{array}{l}[\alpha \mapsto \tau_1]\ t_1 \rhd t_2 \\ [\alpha \mapsto \tau_1]\ \tau_2 \rhd \tau_2'\end{array}}{[\alpha \mapsto \tau_1]\ (t_1\ [\tau_2]) \rhd (t_2\ [\tau_2'])} \quad \textsc{TtSubst\_TApp}$$

$$\frac{\begin{array}{l}[\alpha \mapsto \tau_1]\ t_1 \rhd t_2 \\ [\alpha \mapsto \tau_1]\ \tau_2 \rhd \tau_2'\end{array}}{[\alpha \mapsto \tau_1]\ (t_1 : \tau_2) \rhd (t_2 : \tau_2')} \quad \textsc{TtSubst\_Annot}$$

$\boxed{t \longrightarrow tv}$ Operational semantics

$$\frac{\begin{array}{l}t_2 \longrightarrow tv_2 \\ t_1\ tv_2 \longrightarrow tv_3\end{array}}{t_1\ t_2 \longrightarrow tv_3} \quad \textsc{E\_App1}$$

$$\frac{\begin{array}{l}t \longrightarrow tv_1 \\ tv_1\ tv_2 \longrightarrow tv_3\end{array}}{t\ tv_2 \longrightarrow tv_3} \quad \textsc{E\_App2}$$

$$\frac{\begin{array}{l}[x \mapsto tv_1]\ t \rhd t' \\ t' \longrightarrow tv_2\end{array}}{(\lambda(x : \tau) \Rightarrow t)\ tv_1 \longrightarrow tv_2} \quad \textsc{E\_AppAbs}$$

$$\frac{t \longrightarrow tv}{(\lambda\{\alpha : \kappa\} \Rightarrow t) \longrightarrow (\lambda\{\alpha : \kappa\} \Rightarrow tv)} \quad \textsc{E\_TAbs}$$

$$\frac{t \longrightarrow tv}{t\ [\tau] \longrightarrow tv\ [\tau]} \quad \textsc{E\_TApp}$$

$$\frac{\begin{array}{l}[\alpha \mapsto \tau]\ t \rhd t' \\ t' \longrightarrow tv\end{array}}{(\lambda\{\alpha : \kappa\} \Rightarrow t)\ [\tau] \longrightarrow tv} \quad \textsc{E\_TAppAbs}$$

$$\frac{t \longrightarrow tv}{(t : \tau) \longrightarrow (tv : \tau)} \quad \textsc{E\_Annot}$$

$\boxed{tv \longrightarrow v}$ type erasure

$$\frac{}{(\lambda(x : \tau) \Rightarrow t) \longrightarrow (\lambda x \Rightarrow t)} \quad \textsc{Erase\_Abs}$$

$$\frac{tv \longrightarrow v}{(\lambda\{\alpha : \kappa\} \Rightarrow tv) \longrightarrow v} \quad \textsc{Erase\_TAbs}$$

$$\frac{tv \longrightarrow v}{(tv\ [\tau]) \longrightarrow v} \quad \textsc{Erase\_TApp}$$

$$\frac{tv \longrightarrow v}{(tv : \tau) \longrightarrow v} \quad \textsc{Erase\_Annot}$$

```
Definition rules:         51 good    0 bad
Definition rule clauses: 114 good    0 bad
```