

Term variables, x, y, z variables
Type variables, a, b type variables
Integer, i

Monotypes, τ	$::=$ $ $ Int $ $ a $ $ $\tau_1 \rightarrow \tau_2$	Monotypes
Rho-types, ρ	$::=$ $ $ τ $ $ (ρ)	Rho-types S
Polytypes, σ	$::=$ $ $ ρ $ $ $\forall a. \sigma$ $ $ $[a \mapsto \tau] \sigma$	bind a in σ M
Terms, t, u	$::=$ $ $ i $ $ x $ $ $\lambda x. t$ $ $ $t u$ $ $ $\lambda(x :: \sigma). t$ $ $ let $x = u$ in t $ $ $t :: \sigma$ $ $ (t) $ $ $t[x \rightsquigarrow t']$	Literal Variable Abstraction Application Typed abstraction Local binding Type annotation S M
<i>value</i> , v	$::=$ $ $ i $ $ $\lambda x. t$ $ $ $\lambda(x :: \sigma). t$	bind x in t bind x in t
Typing contexts, Γ	$::=$ $ $ ϵ $ $ $\Gamma, x : \sigma$	empty context assumption
fv	$::=$ $ $ $fv(t)$	M
$ftvany$	$::=$ $ $ $ftv(\tau)$ $ $ $ftv(\rho)$ $ $ $ftv(\sigma)$ $ $ $ftv(\Gamma)$	M M M M
<i>terminals</i>	$::=$ $ $ \longrightarrow $ $ \rightarrow $ $ $::$ $ $ fv $ $ ftv	

		\in
		\notin
		\vdash
<i>formula</i>	$::=$	
		<i>judgement</i>
		value v
		uniq Γ
		closed σ
		$a \notin ftvany$
		$x : \sigma \in \Gamma$
<i>JTyping</i>	$::=$	
		$\Gamma \vdash t : \sigma$
<i>JStep</i>	$::=$	
		$t \longrightarrow u$
<i>judgement</i>	$::=$	
		<i>JTyping</i>
		<i>JStep</i>
<i>user_syntax</i>	$::=$	
		Term variables
		Type variables
		Integer
		Monotypes
		Rho-types
		Polytypes
		Terms
		<i>value</i>
		Typing contexts
		<i>fv</i>
		<i>ftvany</i>
		<i>terminals</i>
		<i>formula</i>

$\boxed{\Gamma \vdash t : \sigma}$

$$\begin{array}{c}
\frac{}{\Gamma \vdash i : \mathbf{Int}} \quad \text{TYP_INT} \\
\\
\frac{\text{uniq } \Gamma \quad x : \sigma \in \Gamma}{\Gamma \vdash i : \sigma} \quad \text{TYP_VAR} \\
\\
\frac{\Gamma, x : \tau_1 \vdash t : \tau_2}{\Gamma \vdash (\lambda x. t) : (\tau_1 \rightarrow \tau_2)} \quad \text{TYP_ABS} \\
\\
\frac{\Gamma \vdash t : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash u : \tau_1}{\Gamma \vdash t u : \tau_2} \quad \text{TYP_APP}
\end{array}$$

$$\begin{array}{c}
\frac{\Gamma \vdash u : \sigma \quad \Gamma, x : \sigma \vdash t : \rho}{\Gamma \vdash \text{let } x = u \text{ in } t : \rho} \text{ TYP_LET} \\
\\
\frac{\text{closed } \sigma \quad \Gamma \vdash t : \sigma}{\Gamma \vdash (t :: \sigma) : \sigma} \text{ TYP_ANNOT} \\
\\
\frac{\Gamma \vdash t : \rho}{\Gamma \vdash t : \forall a. \rho} \text{ TYP_GEN} \\
\\
\frac{\Gamma \vdash t : \forall a. \rho}{\Gamma \vdash t : [a \mapsto \tau] \rho} \text{ TYP_INST}
\end{array}$$

$$\boxed{t \longrightarrow u}$$

$$\begin{array}{c}
\frac{u \longrightarrow u'}{\text{let } x = u \text{ in } t \longrightarrow \text{let } x = u' \text{ in } t} \text{ STEP_LET1} \\
\\
\frac{}{\text{let } x = v \text{ in } t \longrightarrow t[x \rightsquigarrow v]} \text{ STEP_LET} \\
\\
\frac{t \longrightarrow t'}{t \ u \longrightarrow t' \ u} \text{ STEP_APP1} \\
\\
\frac{u \longrightarrow u'}{(\lambda x. t) \ u \longrightarrow (\lambda x. t) \ u'} \text{ STEP_APP2} \\
\\
\frac{}{(\lambda x. t) \ v \longrightarrow t[x \rightsquigarrow v]} \text{ STEP_APP} \\
\\
\frac{u \longrightarrow u'}{(\lambda(x :: \sigma). t) \ u \longrightarrow (\lambda(x :: \sigma). t) \ u'} \text{ STEP_ANNOT_APP2} \\
\\
\frac{}{(\lambda(x :: \sigma). t) \ v \longrightarrow t[x \rightsquigarrow v]} \text{ STEP_ANNOT_APP} \\
\\
\frac{}{t :: \sigma \longrightarrow t} \text{ STEP_ERASE}
\end{array}$$

Definition rules: 16 good 0 bad
 Definition rule clauses: 31 good 0 bad