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

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

$$
\begin{array}{lll}
 & | & \in \\
 & | & \notin \\
 & | & \vdash
\end{array}
$$

$$
\begin{array}{lll}
\textit{formula} & ::= & \\
 & | & \textit{judgement} \\
 & | & \mathsf{value}\ v \\
 & | & \mathsf{uniq}\ \Gamma \\
 & | & \mathsf{closed}\ \sigma \\
 & | & a \notin \textit{ftvany} \\
 & | & x : \sigma\ \in\ \Gamma
\end{array}
$$

$$
\begin{array}{lll}
\textit{JTyping} & ::= & \\
 & | & \Gamma \vdash t : \sigma
\end{array}
$$

$$
\begin{array}{lll}
\textit{JStep} & ::= & \\
 & | & t \longrightarrow u
\end{array}
$$

$$
\begin{array}{lll}
\textit{judgement} & ::= & \\
 & | & \textit{JTyping} \\
 & | & \textit{JStep}
\end{array}
$$

$$
\begin{array}{lll}
\textit{user\_syntax} & ::= & \\
 & | & \text{Term variables} \\
 & | & \text{Type variables} \\
 & | & \text{Integer} \\
 & | & \text{Monotypes} \\
 & | & \text{Rho-types} \\
 & | & \text{Polytypes} \\
 & | & \text{Terms} \\
 & | & \textit{value} \\
 & | & \text{Typing contexts} \\
 & | & \textit{fv} \\
 & | & \textit{ftvany} \\
 & | & \textit{terminals} \\
 & | & \textit{formula}
\end{array}
$$

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

$$\frac{}{\Gamma \vdash i : \mathtt{Int}}\ \text{TYP\_INT}$$

$$\frac{\mathsf{uniq}\ \Gamma \qquad x : \sigma\ \in\ \Gamma}{\Gamma \vdash i : \sigma}\ \text{TYP\_VAR}$$

$$\frac{\Gamma, x : \tau_1 \vdash t : \tau_2}{\Gamma \vdash (\lambda x.t) : (\tau_1 \to \tau_2)}\ \text{TYP\_ABS}$$

$$\frac{\Gamma \vdash t : \tau_1 \to \tau_2 \qquad \Gamma \vdash u : \tau_1}{\Gamma \vdash t\ u : \tau_2}\ \text{TYP\_APP}$$

$$\frac{\Gamma \vdash u : \sigma \quad \Gamma, x : \sigma \vdash t : \rho}{\Gamma \vdash \texttt{let}\ x = u\ \texttt{in}\ t : \rho}\ \text{TYP\_LET}$$

$$\frac{\mathsf{closed}\,\sigma \quad \Gamma \vdash t : \sigma}{\Gamma \vdash (t :: \sigma) : \sigma}\ \text{TYP\_ANNOT}$$

$$\frac{\Gamma \vdash t : \rho}{\Gamma \vdash t : \forall a.\rho}\ \text{TYP\_GEN}$$

$$\frac{\Gamma \vdash t : \forall a.\rho}{\Gamma \vdash t : [a \mapsto \tau]\,\rho}\ \text{TYP\_INST}$$

$\boxed{t \longrightarrow u}$

$$\frac{u \longrightarrow u'}{\texttt{let}\ x = u\ \texttt{in}\ t \longrightarrow \texttt{let}\ x = u'\ \texttt{in}\ t}\ \text{STEP\_LET1}$$

$$\frac{}{\texttt{let}\ x = v\ \texttt{in}\ t \longrightarrow t\,[x \rightsquigarrow v]}\ \text{STEP\_LET}$$

$$\frac{t \longrightarrow t'}{t\ u \longrightarrow t'\ u}\ \text{STEP\_APP1}$$

$$\frac{u \longrightarrow u'}{(\lambda x.t)\ u \longrightarrow (\lambda x.t)\ u'}\ \text{STEP\_APP2}$$

$$\frac{}{(\lambda x.t)\ v \longrightarrow t\,[x \rightsquigarrow v]}\ \text{STEP\_APP}$$

$$\frac{u \longrightarrow u'}{(\lambda(x :: \sigma).t)\ u \longrightarrow (\lambda(x :: \sigma).t)\ u'}\ \text{STEP\_ANNOT\_APP2}$$

$$\frac{}{(\lambda(x :: \sigma).t)\ v \longrightarrow t\,[x \rightsquigarrow v]}\ \text{STEP\_ANNOT\_APP}$$

$$\frac{}{t :: \sigma \longrightarrow t}\ \text{STEP\_ERASE}$$

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