$x,\ y$	variables		
$i,\ n$	indicies		
e	::=		expressions
	$\mid$	$\mathbf{proc}\ pat \to cmd$	
pat	::=		patterns
$binds$	::=		bindings
$\alpha,\ \beta,\ \sigma,\ \tau$	::=		types
	$\mid$	x	variables
	$\mid$	$\tau_1\ \tau_2$	application
	$\mid$	$\tau_1 \to \ ...\ \to \tau_n$	functions
	$\mid$	$\forall x.\ \tau$	quantification
	$\mid$	$'[]$	promoted nil
	$\mid$	$\tau_1\ ':\ ...\ ':\ \tau_n$	promoted cons
$\Gamma,\ \Delta$	::=		contexts
	$\mid$	$\emptyset$	
	$\mid$	$\Gamma_1,\ ...\ , \Gamma_n$	concatenation
cmd	::=		commands
	$\mid$	$e_1 \multimap e_2$	arrow application (first-order)
	$\mid$	$e_1 \mathrel{-\!\!\prec} e_2$	arrow application (higher-order)
	$\mid$	$\mathbf{if}\ e\ \mathbf{then}\ cmd_1\ \mathbf{else}\ cmd_2$	branching
	$\mid$	$\mathbf{case}\ e\ \mathbf{of}\ \{\ alts\ \}$	case analysis
	$\mid$	$\mathbf{let}\ binds\ \mathbf{in}\ cmd$	local binding
	$\mid$	$\lambda\, pat \to cmd$	command abstraction
	$\mid$	$cmd\ e$	command application
	$\mid$	$(\!\mid e\ cmd_1\ ...\ cmd_n \mid\!)$	control operator
	$\mid$	$\mathbf{do}\ \{\ stmt;\ cmd\ \}$	sequencing
$alts$	::=		
	$\mid$	$\overline{pat_i \to cmd_i}^{\,i}$	
$stmt$	::=		statements
	$\mid$	$\mathbf{let}\ binds$	
	$\mid$	$pat \leftarrow cmd$	

$\boxed{pat :: \tau \Rightarrow \Delta}$ pattern typing

$\boxed{binds \Rightarrow \Delta}$ binding typing

$\boxed{\Gamma \vdash e :: \tau}$ expression typing

$$\frac{\begin{array}{l} pat :: \tau_1 \Rightarrow \Delta \\ \Gamma \mid \Delta \vdash_\alpha cmd :: '[] \rightharpoonup \tau_2 \end{array}}{\Gamma \vdash \mathbf{proc}\ pat \to cmd :: \alpha\ \tau_1\ \tau_2}\ \textsc{Expr\_Proc}$$

$\boxed{\Gamma \mid \Delta \vdash_\alpha cmd :: \sigma \rightharpoonup \tau}$ command typing

$$\frac{\Gamma \vdash e_1 :: \alpha\ (\textsc{Stk}[\![\beta\ ' : \sigma]\!])\ \tau \qquad \Gamma, \Delta \vdash e_2 :: \beta}{\Gamma \,|\, \Delta \vdash_\alpha e_1 \prec e_2 :: \sigma \rightharpoonup \tau}\ \textsc{Cmd\_AppF}$$

$$\frac{\Gamma, \Delta \vdash e_1 :: \alpha\ (\textsc{Stk}[\![\beta\ ' : \sigma]\!])\ \tau \qquad \Gamma, \Delta \vdash e_2 :: \beta}{\Gamma \,|\, \Delta \vdash_\alpha e_1 \prec\!\!\prec e_2 :: \sigma \rightharpoonup \tau}\ \textsc{Cmd\_AppH}$$

$$\frac{\Gamma \,|\, \Delta \vdash_\alpha cmd :: (\beta\ ' : \sigma) \rightharpoonup \tau \qquad \Gamma, \Delta \vdash e :: \beta}{\Gamma \,|\, \Delta \vdash_\alpha cmd\ e :: \sigma \rightharpoonup \tau}\ \textsc{Cmd\_AppC}$$

$$\frac{pat :: \beta \Rightarrow \Delta_2 \qquad \Gamma \,|\, \Delta_1, \Delta_2 \vdash_\alpha cmd :: \sigma \rightharpoonup \tau}{\Gamma \,|\, \Delta_1 \vdash_\alpha \lambda\, pat \to cmd :: (\beta\ ' : \sigma) \rightharpoonup \tau}\ \textsc{Cmd\_Abs}$$

$$\frac{\Gamma \vdash e :: \forall x.\ \overline{\alpha_i\ (\textsc{Env}[\![x, \sigma_i]\!])\ \tau_i}^{\,i} \to \alpha\ (\textsc{Env}[\![x, \sigma_1]\!])\ \tau_1 \qquad \overline{\Gamma \,|\, \Delta \vdash_{\alpha_i} cmd_i :: \sigma_i \rightharpoonup \tau_i}^{\,i}}{\Gamma \,|\, \Delta \vdash_{\alpha_1} (\!|\, e\ \overline{cmd_i}^{\,i}\ |\!) :: \sigma_1 \rightharpoonup \tau_1}\ \textsc{Cmd\_Op}$$

$$\frac{binds \Rightarrow \Delta_2 \qquad \Gamma \,|\, \Delta_1, \Delta_2 \vdash_\alpha cmd :: '[] \rightharpoonup \tau}{\Gamma \,|\, \Delta_1 \vdash_\alpha \textbf{let}\ binds\ \textbf{in}\ cmd :: '[] \rightharpoonup \tau}\ \textsc{Cmd\_Let}$$

$$\frac{\Gamma, \Delta \vdash e :: \tau_1 \qquad \overline{pat_i :: \tau_1 \Rightarrow \Delta_i}^{\,i} \qquad \overline{\Gamma \,|\, \Delta_1, \Delta_i \vdash_\alpha cmd_i :: '[] \rightharpoonup \tau_2}^{\,i}}{\Gamma \,|\, \Delta_1 \vdash_\alpha \textbf{case}\ e\ \textbf{of}\ \{\ \overline{pat_i \to cmd_i}^{\,i}\ \} :: '[] \rightharpoonup \tau_2}\ \textsc{Cmd\_Case}$$

$$\frac{\Gamma, \Delta \vdash e :: \mathrm{Bool} \qquad \Gamma \,|\, \Delta \vdash_\alpha cmd_1 :: '[] \rightharpoonup \tau \qquad \Gamma \,|\, \Delta \vdash_\alpha cmd_2 :: '[] \rightharpoonup \tau}{\Gamma \,|\, \Delta \vdash_\alpha \textbf{if}\ e\ \textbf{then}\ cmd_1\ \textbf{else}\ cmd_2 :: '[] \rightharpoonup \tau}\ \textsc{Cmd\_If}$$

$$\frac{\Gamma \,|\, \Delta \vdash_\alpha \textbf{let}\ binds\ \textbf{in}\ cmd :: '[] \rightharpoonup \tau}{\Gamma \,|\, \Delta \vdash_\alpha \textbf{do}\ \{\ \textbf{let}\ binds;\ cmd\ \} :: '[] \rightharpoonup \tau}\ \textsc{Cmd\_Do\_Let}$$

$$\frac{pat :: \tau_1 \Rightarrow \Delta_2 \qquad \Gamma \,|\, \Delta_1 \vdash_\alpha cmd_1 :: '[] \rightharpoonup \tau_1 \qquad \Gamma \,|\, \Delta_1, \Delta_2 \vdash_\alpha cmd_2 :: '[] \rightharpoonup \tau_2}{\Gamma \,|\, \Delta_1 \vdash_\alpha \textbf{do}\ \{\ pat \leftarrow cmd_1;\ cmd_2\ \} :: '[] \rightharpoonup \tau_2}\ \textsc{Cmd\_Do\_Bind}$$

```
Definition rules:         11 good    0 bad
Definition rule clauses: 35 good    0 bad
```