$x,\ y$	variables		
$i,\ n$	indicies		
e	$::=$		expressions
	$\mid$	$\mathbf{proc}\, pat \rightarrow cmd$	
pat	$::=$		patterns
$binds$	$::=$		bindings
$\alpha,\ \sigma,\ \tau$	$::=$		types
	$\mid$	x	variables
	$\mid$	$\tau_1\ \tau_2$	application
	$\mid$	$\tau_1 \rightarrow\ ...\ \rightarrow \tau_n$	functions
	$\mid$	$\forall x.\ \tau$	quantification
	$\mid$	'[]	promoted nil
	$\mid$	τ_1 ': ... ': τ_n	promoted cons
$\Gamma,\ \Delta$	$::=$		contexts
	$\mid$	$\varnothing$	
	$\mid$	$\Gamma_1,\ ...\ , \Gamma_n$	concatenation
cmd	$::=$		commands
	$\mid$	$e_1 \mathrel{-\!\!\prec} e_2$	arrow application (first-order)
	$\mid$	$e_1 \mathrel{-\!\!\prec\!\!\!\prec} e_2$	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
	$\mid$	$\lambda\, pat \rightarrow cmd$	command abstraction
	$\mid$	$cmd\ e$	command application
	$\mid$	$(\!\mid e\ cmd_1\ ...\ cmd_n \mid\!)$	control operator
	$\mid$	$\mathbf{do}\ \{\ stmt;\ cmd\ \}$	sequencing
$alts$	$::=$		
	$\mid$	$\overline{pat_i \rightarrow cmd_i}^{\,i}$	
$stmt$	$::=$		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

$$\frac{\begin{array}{c} pat :: \tau_1 \Rightarrow \Delta \\ \Gamma \,|\, \Delta \vdash_\alpha cmd :: \text{'[]} \rightharpoonup \tau_2 \end{array}}{\Gamma \vdash \mathbf{proc}\, pat \rightarrow cmd :: \alpha\ \tau_1\ \tau_2} \quad \textsc{Expr\_Proc}$$

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

$$\frac{\begin{array}{l}\Gamma \vdash e_1 :: \alpha\ (\textsc{Stk}[\![\tau_1\ \mathrel{'{:}}\ \sigma]\!])\ \tau_2\\ \Gamma, \Delta \vdash e_2 :: \tau_1\end{array}}{\Gamma \,|\, \Delta \vdash_{\alpha} e_1 \mathrel{-\!\!\!\prec} e_2 :: \sigma \rightharpoonup \tau_2}\quad \textsc{Cmd\_AppF}$$

$$\frac{\begin{array}{l}\Gamma, \Delta \vdash e_1 :: \alpha\ (\textsc{Stk}[\![\tau_1\ \mathrel{'{:}}\ \sigma]\!])\ \tau_2\\ \Gamma, \Delta \vdash e_2 :: \tau_1\end{array}}{\Gamma \,|\, \Delta \vdash_{\alpha} e_1 \mathrel{-\!\!\!\prec\!\!\!\prec} e_2 :: \sigma \rightharpoonup \tau_2}\quad \textsc{Cmd\_AppH}$$

$$\frac{\begin{array}{l}\Gamma \,|\, \Delta \vdash_{\alpha} cmd :: (\tau_1\ \mathrel{'{:}}\ \sigma) \rightharpoonup \tau_2\\ \Gamma, \Delta \vdash e :: \tau_1\end{array}}{\Gamma \,|\, \Delta \vdash_{\alpha} cmd\ e :: \sigma \rightharpoonup \tau_2}\quad \textsc{Cmd\_AppC}$$

$$\frac{\begin{array}{l}pat :: \tau_1 \Rightarrow \Delta_2\\ \Gamma \,|\, \Delta_1, \Delta_2 \vdash_{\alpha} cmd :: \sigma \rightharpoonup \tau_2\end{array}}{\Gamma \,|\, \Delta_1 \vdash_{\alpha} \lambda\, pat \rightarrow cmd :: (\tau_1\ \mathrel{'{:}}\ \sigma) \rightharpoonup \tau_2}\quad \textsc{Cmd\_Abs}$$

$$\frac{\begin{array}{l}\Gamma \vdash e :: \forall x.\ \overline{\alpha_i\ (\textsc{Env}[\![x, \sigma_i]\!])\ \tau_i}^{\,i} \rightarrow \alpha_1\ (\textsc{Env}[\![x, \sigma_1]\!])\ \tau_1\\ \overline{\Gamma \,|\, \Delta \vdash_{\alpha_i} cmd_i :: \sigma_i \rightharpoonup \tau_i}^{\,i}\end{array}}{\Gamma \,|\, \Delta \vdash_{\alpha_1} \llparenthesis\, e\ \overline{cmd_i}^{\,i}\ \rrparenthesis :: \sigma_1 \rightharpoonup \tau_1}\quad \textsc{Cmd\_Op}$$

$$\frac{\begin{array}{l}binds \Rightarrow \Delta_2\\ \Gamma \,|\, \Delta_1, \Delta_2 \vdash_{\alpha} cmd :: \sigma \rightharpoonup \tau\end{array}}{\Gamma \,|\, \Delta_1 \vdash_{\alpha} \mathbf{let}\ binds\ \mathbf{in}\ cmd :: \sigma \rightharpoonup \tau}\quad \textsc{Cmd\_Let}$$

$$\frac{\begin{array}{l}\Gamma, \Delta \vdash e :: \tau_1\\ \overline{pat_i :: \tau_1 \Rightarrow \Delta_i}^{\,i}\\ \overline{\Gamma \,|\, \Delta_1, \Delta_i \vdash_{\alpha} cmd_i :: \sigma \rightharpoonup \tau_2}^{\,i}\end{array}}{\Gamma \,|\, \Delta_1 \vdash_{\alpha} \mathbf{case}\ e\ \mathbf{of}\ \{\ \overline{pat_i \rightarrow cmd_i}^{\,i}\ \} :: \sigma \rightharpoonup \tau_2}\quad \textsc{Cmd\_Case}$$

$$\frac{\begin{array}{l}\Gamma, \Delta \vdash e :: \mathsf{Bool}\\ \Gamma \,|\, \Delta \vdash_{\alpha} cmd_1 :: \sigma \rightharpoonup \tau\\ \Gamma \,|\, \Delta \vdash_{\alpha} cmd_2 :: \sigma \rightharpoonup \tau\end{array}}{\Gamma \,|\, \Delta \vdash_{\alpha} \mathbf{if}\ e\ \mathbf{then}\ cmd_1\ \mathbf{else}\ cmd_2 :: \sigma \rightharpoonup \tau}\quad \textsc{Cmd\_If}$$

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

$$\frac{\begin{array}{l}pat :: \tau_1 \Rightarrow \Delta_2\\ \Gamma \,|\, \Delta_1 \vdash_{\alpha} cmd_1 :: {'}[] \rightharpoonup \tau_1\\ \Gamma \,|\, \Delta_1, \Delta_2 \vdash_{\alpha} cmd_2 :: {'}[] \rightharpoonup \tau_2\end{array}}{\Gamma \,|\, \Delta_1 \vdash_{\alpha} \mathbf{do}\ \{\ pat \leftarrow cmd_1;\ cmd_2\ \} :: {'}[] \rightharpoonup \tau_2}\quad \textsc{Cmd\_Do\_Bind}$$

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