During static type checking a process P mentions only variables x but not addresses a. At runtime, processes P can also refer to addresses. Since they are treated identically to variables, we do not explicitly write out the dynamic version of the typing judgment for processes.

Abstract Syntax

Types	τ	$::=$	$\tau_1 \to \tau_2 \mid \&_{i\in I}(i : \tau_i) \mid \tau_1 \times \tau_2 \mid 1 \mid \sum_{i\in I}(i : \tau_i) \mid \mu\alpha.\,\tau$	
Contexts	Γ	$::=$	$\cdot \mid \Gamma, x : \tau$	(all variables distinct)
Processes	P	$::=$	$x \leftarrow P \,;\, Q$	allocate/spawn
		$\mid$	$x^w \leftarrow y^R$	copy
		$\mid$	$x^W.V \mid \mathsf{case}\ x^R\ K$	$(1, \times, +, \mu)$
		$\mid$	$x^R.V \mid \mathsf{case}\ x^W\ K$	$(\to, \&)$
		$\mid$	$\mathsf{fix}\ f.\,P$	recursion
Small values	V	$::=$	$\langle\,\rangle \mid \langle a_1, a_2\rangle \mid i \cdot a \mid \mathsf{fold}\ a$	
Continuations	K	$::=$	$(\langle\,\rangle \Rightarrow P) \mid (\langle x_1, x_2\rangle \Rightarrow P) \mid (i \cdot x_i \Rightarrow P_i)_{i\in I} \mid (\mathsf{fold}\ x \Rightarrow P)$	
Cell contents	W	$::=$	$V \mid K$	
Configurations	$\mathcal{C}$	$::=$	$\cdot \mid \mathcal{C}_1, \mathcal{C}_2 \mid \mathsf{proc}\ d\ P \mid \mathsf{cell}\ c\ W$	
Address typings	Ψ, Δ	$::=$	$a : \tau \mid \cdot \mid \Psi_1, \Psi_2$	

Judgments

$\Gamma \vdash P :: (z : \sigma)$	process P reads from Γ and writes to $z : \sigma$
$\Psi \vdash \mathcal{C} :: \Delta$	configuration $\mathcal{C}$ reads from Ψ and writes to Δ
$\Psi \vdash \mathcal{C}$ *final*	configuration $\mathcal{C}$ is final (consists only of cells)

Theorems

Preservation. If $\Psi \vdash \mathcal{C} :: \Delta$ and $\mathcal{C} \mapsto \mathcal{C}'$ then $\Psi \vdash \mathcal{C}' :: \Delta'$ for some $\Delta' \supseteq \Delta$.

Progress. If $\cdot \vdash \mathcal{C} :: \Delta$ then either $\mathcal{C} \mapsto \mathcal{C}'$ for some $\mathcal{C}'$ or $\mathcal{C}$ *final*.

Statics, Allocate and Copy

$$\frac{\Gamma \vdash P :: (x : \tau) \quad \Gamma, x : \tau \vdash Q :: (z : \sigma)}{\Gamma \vdash (x \leftarrow P \,;\, Q) :: (z : \sigma)}\ \mathsf{tp/alloc} \qquad \frac{y : \tau \in \Gamma}{\Gamma \vdash x \leftarrow y :: (x : \tau)}\ \mathsf{tp/copy}$$

Statics, Positive Types

$$\frac{}{\Gamma \vdash x^W.\langle\,\rangle :: (x : 1)}\ \textsf{w/unit} \qquad \frac{x : 1 \in \Gamma \quad \Gamma \vdash P :: (z : \sigma)}{\Gamma \vdash \textsf{case}\ x^R\ (\langle\,\rangle \Rightarrow P) :: (z : \sigma)}\ \textsf{r/unit}$$

$$\frac{y : \tau \in \Gamma \qquad z : \sigma \in \Gamma}{\Gamma \vdash x^W.\langle y, z\rangle :: (x : \tau \times \sigma)}\ \textsf{w/pair} \qquad \frac{x : \tau_1 \times \tau_2 \in \Gamma \quad \Gamma, x_1 : \tau_1, x_2 : \tau_2 \vdash P :: (z : \sigma)}{\Gamma \vdash \textsf{case}\ x^R\ (\langle x_1, x_2\rangle \Rightarrow P) : (z : \sigma)}\ \textsf{r/pair}$$

$$\frac{(j \in I) \quad y : \tau_j \in \Gamma}{\Gamma \vdash x^W.(j \cdot y) :: (x : \sum_{i \in I}(i : \tau_i))}\ \textsf{w/tag} \qquad \frac{x : \sum_{i \in I}(i : \tau_i) \in \Gamma \quad \Gamma, y : \tau_i \vdash P_i :: (z : \sigma)\ (\forall i \in I)}{\Gamma \vdash \textsf{case}\ x^R\ (i \cdot y \Rightarrow P_i)_{i \in I} :: (z : \sigma)}\ \textsf{r/tag}$$

$$\frac{y : [\mu\alpha.\,\tau/\alpha]\tau \in \Gamma}{\Gamma \vdash x^W.(\textsf{fold}\ y) :: (x : \mu\alpha.\,\tau)}\ \textsf{w/fold} \qquad \frac{x : \mu\alpha.\,\tau \in \Gamma \quad \Gamma, y : [\mu\alpha.\,\tau/\alpha]\tau \vdash P :: (z : \sigma)}{\Gamma \vdash \textsf{case}\ x^R\ (\textsf{fold}\ y \Rightarrow P) :: (z : \sigma)}\ \textsf{r/fold}$$

Statics, Negative Types

$$\frac{\Gamma, y : \tau \vdash P :: (z : \sigma)}{\Gamma \vdash \textsf{case}\ x^W\ (\langle y, z\rangle \Rightarrow P) :: (x : \tau \to \sigma)}\ \textsf{w/fun} \qquad \frac{x : \tau \to \sigma \in \Gamma \quad y : \tau \in \Gamma}{\Gamma \vdash x^R.\langle y, z\rangle :: (z : \sigma)}\ \textsf{r/fun}$$

$$\frac{\Gamma \vdash P_i :: (z_i : \tau_i) \quad (\text{for all } i \in I)}{\Gamma \vdash \textsf{case}\ x^W\ (i \cdot z_i \Rightarrow P_i)_{i \in I} :: (x : \&_{i \in I}(i : \tau_i))}\ \textsf{w/record} \qquad \frac{x : \&_{i \in I}(i : \tau_i) \in \Gamma \quad j \in I}{\Gamma \vdash x^R.(j \cdot z) :: (z : \tau_j)}\ \textsf{r/record}$$

Statics, Configurations

$$\frac{\Psi \vdash P :: (c : \tau)}{\Psi \vdash \textsf{proc}\ c\ P :: (\Psi, c : \tau)}\ \textsf{tp/proc}$$

$$\frac{\Psi \vdash c^W.V :: (c : \tau)}{\Psi \vdash \textsf{cell}\ c\ V :: (\Psi, c : \tau)}\ \textsf{tp/cell/val} \qquad \frac{\Psi \vdash \textsf{case}\ c^W\ K :: (c : \tau)}{\Psi \vdash \textsf{cell}\ c\ K :: (\Psi, c : \tau)}\ \textsf{tp/cell/cont}$$

$$\frac{}{\Psi \vdash (\cdot) :: \Psi}\ \textsf{tp/empty} \qquad \frac{\Psi \vdash \mathcal{C}_1 :: \Psi_1 \quad \Psi_1 \vdash \mathcal{C}_2 :: \Psi_2}{\Psi \vdash (\mathcal{C}_1, \mathcal{C}_2) :: \Psi_2}\ \textsf{tp/join}$$

$$\frac{}{\Psi \vdash (\cdot)\ \mathit{final}}\ \textsf{fin/empty} \qquad \frac{\Psi \vdash \mathcal{C}\ \mathit{final}}{\Psi \vdash (\mathcal{C}, \textsf{cell}\ c\ W)\ \mathit{final}}\ \textsf{fin/cell}$$

Dynamics

$$
\begin{array}{llcll}
 & \mathsf{proc}\ d\ (x \leftarrow P\ ;\ Q) & \mapsto & \mathsf{proc}\ c\ ([c/x]P), \mathsf{proc}\ d\ ([c/x]Q) & (\text{alloc/spawn, } c \text{ fresh}) \\
!\mathsf{cell}\ c\ W, & \mathsf{proc}\ d\ (d^W \leftarrow c^R) & \mapsto & !\mathsf{cell}\ d\ W & (\text{copy}) \\
\\
 & \mathsf{proc}\ d\ (d^W.V) & \mapsto & !\mathsf{cell}\ d\ V & (\text{write: } \times, 1, +, \mu) \\
!\mathsf{cell}\ c\ V, & \mathsf{proc}\ d\ (\mathsf{case}\ c^R\ K) & \mapsto & \mathsf{proc}\ d\ (V \triangleright K) & (\text{read: } \times, 1, +, \mu) \\
\\
 & \mathsf{proc}\ d\ (\mathsf{case}\ d^W\ K) & \mapsto & !\mathsf{cell}\ d\ K & (\text{write: } \rightarrow, \&) \\
!\mathsf{cell}\ c\ K, & \mathsf{proc}\ d\ (c^R.V) & \mapsto & \mathsf{proc}\ d\ (V \triangleright K) & (\text{read: } \rightarrow, \&)
\end{array}
$$

$$
\begin{array}{rcccl}
\langle\,\rangle & \triangleright & (\langle\,\rangle \Rightarrow P) & = & P \\
\langle c_1, c_2\rangle & \triangleright & (\langle x_1, x_2\rangle \Rightarrow P) & = & [c_1/x_1, c_2/x_2]P \\
j \cdot c & \triangleright & (i \cdot x_i \Rightarrow P_i)_{i \in I} & = & [c/x_j]P_j \\
\mathsf{fold}\ c & \triangleright & (\mathsf{fold}\ x \Rightarrow P) & = & [c/x]P
\end{array}
$$