$termvar,\ x$ term variable

$index,\ i,\ j,\ k,\ n,\ m,\ p$

$$
\begin{array}{lll}
T & ::= & \\
 & | & \mathbf{bool} \\
 & | & \mathbf{int} \\
 & | & \langle T_1,\ ..\ , T_n \rangle \\
 & | & T@r \\
 & | & T@(r_1,\ ..\ , r_n) \\
 & | & T[r_1,\ ..\ , r_n] \\
 & | & T[r_1/r_1',\ ..\ , r_n/r_n'] \\
 & | & \mathbf{coloring}\ (r) \\
 & | & \mathbf{exists}\ r_1,\ ..\ , r_n.(T_1,\ ..\ ,\ T_m), \Phi, Q \rightarrow \mathbf{Tr}
\end{array}
$$

$$
\begin{array}{lll}
\mathit{fresh} & ::= & \\
 & | & \mathit{iv}
\end{array}
$$

$$
\begin{array}{lll}
\mathit{fns} & ::= & \\
 & | & \mathbf{apply}\ (S, E) \\
 & | & \mathit{valid\_interleave}(S, C, E_1,\ ..\ , E_n) \\
 & | & \mathbf{taskid}\ \mathit{fresh} \\
 & | & \mathit{mark\_coherence}(E, M,\ \mathbf{taskid}\ )
\end{array}
$$

$$
\begin{array}{lll}
r & ::= & \\
 & | & \mathbf{nullr} \\
 & | & \mathbf{notnullr}
\end{array}
$$

$$
\begin{array}{lll}
\mathit{rr} & ::= & \\
 & | & [r_1,\ ..\ , r_n]
\end{array}
$$

$$
\begin{array}{lll}
l & ::= & \\
 & | & \mathbf{nil}
\end{array}
$$

$$
\begin{array}{lll}
\Gamma & ::= & \\
 & | & \{(e_1 : T_1),\ ..\ , (e_n : T_n)\} \\
 & | & \varnothing
\end{array}
$$

$$
\begin{array}{lll}
\Omega & ::= & \\
 & | & \{\omega_1,\ ..\ , \omega_n\} \\
 & | & \varnothing
\end{array}
$$

$$
\begin{array}{lll}
\Omega^* & ::= & \\
 & | & \{\omega_1,\ ..\ , \omega_n\} \\
 & | & \mathbf{emptyOst}
\end{array}
$$

$$
\begin{array}{lll}
\mathit{rs} & ::= & \\
 & | & \{r_1,\ ..\ , r_n\} \\
 & | & \varnothing
\end{array}
$$

$$
\begin{array}{lll}
\omega & ::= &
\end{array}
$$

$$
\begin{array}{lll}
 & | & r_1 \leq r_2 \\
 & | & r_1 * r_2 \\
 & & \\
\Phi & ::= & \\
 & | & \{\phi_1, .., \phi_n\} \\
 & | & \varnothing \\
 & & \\
\Phi^* & ::= & \\
 & | & \{\phi_1, .., \phi_n\} \\
 & | & \textbf{emptyPst} \\
 & & \\
\phi & ::= & \\
 & | & \textbf{reads}\,(r) \\
 & | & \textbf{writes}\,(r) \\
 & | & \textbf{reducesid}\,(r) \\
 & & \\
Q & ::= & \\
 & | & \{q_1, .., q_n\} \\
 & & \\
q & ::= & \\
 & | & \textbf{atomic}\,(r) \\
 & | & \textbf{simult}\,(r) \\
 & & \\
M & ::= & \\
 & | & M[[Q]] \\
 & & \\
L & ::= & \\
 & | & \textbf{nil} \\
 & | & L[(e_1, v_1), .., (e_n, v_n)] \\
 & | & L[e/id] \\
 & & \\
K & ::= & \\
 & & \\
H & ::= & \\
 & | & H(l) \\
 & & \\
S & ::= & \\
 & & \\
C & ::= & \\
 & & \\
E & ::= & \\
 & | & [] \\
 & | & [e] \\
 & | & E + +[e] \\
 & & \\
\rho & ::= &
\end{array}
$$

v ::=
- bv
- iv
- $\langle v_1, v_2 \rangle$
- **null**
- l
- $\langle\langle \rho_1, .., \rho_n, v \rangle\rangle$
- H
- K
- **place**

bv ::=
- **true** — constant true
- **false** — constant false

iv ::=
- 0
- $S\ iv$

ee ::=
- $(e_1, .., e_n)$

id ::=

$excl$ ::=

e ::=
- x
- **new** $T@r$
- **null** $T@r$
- **isnull** (e)
- **upregion** $(e, r_1, .., r_n)$
- **downregion** $(e, r_1, .., r_n)$
- **read** (e)
- **read** $(e_1, excl, e_2, e_3)$
- **write** (e_1, e_2)
- **reduce** (id, e_1, e_2)
- **reduceid** (l, e_1, e_2, e_3)
- **newcolor** r
- **color** (e_1, e_2, e_3)
- $e_1 + e_2$
- $e_1 < e_2$
- **let** $id : T = e_1 \in e_2$
- **if** e_1 **then** e_2 **else** e_3 — conditional
- $id[r_1, .., r_n](e_1, .., e_m)$
- **partition** r_p **using** e_1 **as** $r_1, .., r_n \in e_2$
- **pack** e_1 **as** T

	$\mathbf{unpack}\ e_1\ \mathbf{as}\ id : T \in e_2$		
	$\mathbf{function}\ id[r_1, .., r_n](e_1, .., e_m)$		
	$\{\ \mathbf{function}\ id_1\ rr_1\ ee_1, .., \mathbf{function}\ id_n\ rr_n\ ee_n\}$		
	$e : T$		
	$\mathbf{place}$		
	bv		
	iv		
	$\langle v_1, v_2 \rangle$		
	$\mathbf{null}$		
	l		
	$\langle\langle \rho_1, .., \rho_n, v \rangle\rangle$		
	H		
	K		
	$\mathbf{true}$		constant true
	$\mathbf{false}$		constant false
	0		
	$S\ iv$		

formula	::=		
	judgement		judgement
	$\neg formula$	M	negated formula
	$(formula)$	M	bracketed
	$\forall_i.\phi \in \Phi$	M	for all variables in domain of Φ
	$\forall_i.\phi \in \Phi^*$	M	for all variables in domain of Φ^*
	$\exists_i.\phi \in \Phi$	M	for all variables in domain of Φ
	$\forall_i.\omega \in \Omega$	M	for all variables in domain of Ω
	$\forall_i.\omega \in \Omega^*$	M	for all variables in domain of Ω^*
	$\exists_i.\omega \in \Omega$	M	for all variables in domain of Ω
	$\exists_i.\omega \in \Omega^*$	M	for all variables in domain of Ω^*
	$\forall_i.formula$	M	for all variables in i and *formula*
	$\exists_{formula_1}.formula_2$	M	for all variables in $formula_1$ and *for*
	$\exists_{formula_1}.formula_2$ where $formula_3$	M	exists $formula_1$ and $formula_2$ where
	$\Gamma(id)$		lookup
	$formula_1 = formula_2$		equality
	$formula_1 \wedge formula_2$		equality
	$\bigwedge_i .formula$	M	and fold on i and *formula*
	$formula_1 \cap formula_2$	M	
	$formula_1 \cup formula_2$	M	
	$formula_1 \subseteq formula_2$	M	
	$formula_1 \in formula_2$	M	
	$\Gamma, \Phi, \Omega \rightarrow T$		impl
	$\Gamma, \Phi, Q \rightarrow T$		impl
	$r_1, .., r_n$		region list
	ϕ		phi
	ω		om
	Ω		

| | Φ
| | Φ^*
| | $\Phi[r_1/r'_1, .., r_n/r'_n]$
| | $T[r_1/r'_1, .., r_n/r'_n]$
| | $M[\rho_1/r'_1, .., \rho_n/r'_n]$
| | $M[[T]]$
| | $M[[Q]]$
| | $\mathbf{domain}\,(S)$
| | $M(r)$
| | $\Gamma[r_1/r'_1, .., r_n/r'_n]$
| | $\Gamma[e_1/T_1, .., e_n/T_n]$
| | Γ
| | $\Gamma[T[r_1/r'_1, .., r_n/r'_n]/id]$
| | $\Omega[r_1/r'_1, .., r_n/r'_n]$
| | $regions\_of(\Gamma, T)$
| | $regions\_of(\Gamma, T1, T_2)$
| | rs
| | T
| | fns
| | S
| | C
| | M
| | e
| | E
| | L
| | $\overline{e_i = l_i}^{\,i<n}$

terminals ::=

| | $\exists$
| | $\forall$
| | $\in$
| | ω
| | ϕ
| | ρ
| | $\vee$
| | $\wedge$
| | $\neg$
| | $*$
| | $\leq$
| | $\longrightarrow$
| | $\rightarrow$
| | $\Rightarrow$
| | λ
| | $\mapsto$
| | $\vdash$
| | $\varnothing$

	$\varnothing$
	$\varnothing$
	$\varnothing$
	$\times$
	$<:$
	$\langle$
	$\rangle$
	$<$
	$\Downarrow$
	σ
	Γ
	ε

Jtype	::=		
	|	$\Gamma, \Phi, \Omega \vdash e : T$	Typing

Jop	::=		
	|	$M, L, H, S, C \vdash e \mapsto v, E$	Evaluation

judgement	::=	
	|	*Jtype*
	|	*Jop*

user_syntax	::=	
	|	*termvar*
	|	*index*
	|	T
	|	*fresh*
	|	*fns*
	|	r
	|	rr
	|	l
	|	Γ
	|	Ω
	|	Ω^*
	|	rs
	|	ω
	|	Φ
	|	Φ^*
	|	ϕ
	|	Q
	|	q
	|	M
	|	L
	|	K
	|	H
	|	S

$$
\begin{array}{l}
\mid\ C \\
\mid\ E \\
\mid\ \rho \\
\mid\ v \\
\mid\ bv \\
\mid\ iv \\
\mid\ ee \\
\mid\ id \\
\mid\ excl \\
\mid\ e \\
\mid\ formula \\
\mid\ terminals
\end{array}
$$

$\boxed{\Gamma, \Phi, \Omega \vdash e : T}$ Typing

$$\frac{\Gamma, \Phi, \Omega \vdash e_1 : T@(r_1, .., r_n)}{\Gamma, \Phi, \Omega \vdash \mathbf{read}\,(e_1) : T}\ \text{T\_Read}$$

$$\frac{\begin{array}{c}\Gamma, \Phi, \Omega \vdash e_1 : T@(r_1, .., r_n) \\ \Gamma, \Phi, \Omega \vdash e_2 : T\end{array}}{\Gamma, \Phi, \Omega \vdash \mathbf{write}\,(e_1, e_2) : T@(r_1, .., r_n)}\ \text{T\_Write}$$

$$\frac{\begin{array}{c}\Gamma, \Phi, \Omega \vdash e_1 : T_1@(r_1, .., r_n) \\ \Gamma, \Phi, \Omega \vdash e_2 : T_2\end{array}}{\Gamma, \Phi, \Omega \vdash \mathbf{reduce}\,(id, e_1, e_2) : T_1@(r_1, .., r_n)}\ \text{T\_Reduce}$$

$$\frac{}{\Gamma, \Phi, \Omega \vdash \mathbf{new}\ T@r : T@r}\ \text{T\_New}$$

$$\frac{\Gamma, \Phi, \Omega \vdash e : T@(r'_1, .., r'_k)}{\Gamma, \Phi, \Omega \vdash \mathbf{upregion}\,(e_1, r_1, .., r_n) : T@(r_1, .., r_n)}\ \text{T\_UpRgn}$$

$$\frac{\Gamma, \Phi, \Omega \vdash e : T@(r'_1, .., r'_k)}{\Gamma, \Phi, \Omega \vdash \mathbf{downregion}\,(e, r_1, .., r_n) : T@(r_1, .., r_n)}\ \text{T\_DnRgn}$$

$$\frac{}{\Gamma, \Phi, \Omega \vdash \mathbf{newcolor}\ r : \mathbf{coloring}\,(r)}\ \text{T\_NewColor}$$

$$\frac{\begin{array}{c}\Gamma, \Phi, \Omega \vdash e_1 : \mathbf{coloring}\,(r) \\ \Gamma, \Phi, \Omega \vdash e_2 : T@r \\ \Gamma, \Phi, \Omega \vdash e_3 : \mathbf{int}\end{array}}{\Gamma, \Phi, \Omega \vdash \mathbf{color}\,(e_1, e_2, e_3) : \mathbf{coloring}\,(r)}\ \text{T\_Color}$$

$$\frac{\begin{array}{c}\Gamma, \Phi, \Omega \vdash e_1 : \mathbf{coloring}\,(r_p) \\ \Gamma, \Phi, \Omega' \vdash e_2 : T\end{array}}{\Gamma, \Phi, \Omega \vdash \mathbf{partition}\ r_p\ \mathbf{using}\ e_1\ \mathbf{as}\ r_1, .., r_k\ \in\ e_2 : T}\ \text{T\_Partition}$$

$$\frac{\Gamma, \Phi, \Omega \vdash e_1 : T_2[r_1/r'_1, .., r_k/r'_k]}{\Gamma, \Phi, \Omega \vdash \mathbf{pack}\ e_1\ \mathbf{as}\ T_1[r_1, .., r_k] : T_1}\ \text{T\_Pack}$$

$$\frac{\begin{array}{c}\Gamma, \Phi, \Omega \vdash e_1 : T_1 \\ \Gamma', \Phi, \Omega' \vdash e_2 : T_3\end{array}}{\Gamma, \Phi, \Omega \vdash \mathbf{unpack}\ e_1\ \mathbf{as}\ id : T_1[r_1, .., r_k]\ \in\ e_2 : T_3}\ \text{T\_UnPack}$$

$$\frac{}{\Gamma, \Phi, \Omega \vdash id[r_1, .., r_k](e_1, .., e_n) : T}\ \text{T\_Call}$$

$$\frac{}{\Gamma, \Phi, \Omega \vdash \{\,\mathbf{function}\ id_1\,[r_1, .., r_k]\,(e_1, .., e_m), .., \mathbf{function}\ id_n\,[r_1, .., r_k]\,(e_1, .., e_m)\} : T}\ \text{T\_Program}$$

$\boxed{M, L, H, S, C \vdash e \mapsto v, E}$ Evaluation

$$\frac{}{M, L, H, S, C \vdash x \mapsto v, E} \text{ EVA}$$

$$\frac{M, L, H, S, C \vdash e \mapsto l, E}{M, L, H, S, C \vdash \mathbf{read}\,(e) \mapsto v, E} \text{ EREAD1}$$

$$\frac{M, L, H, S, C \vdash e \mapsto l, E}{M, L, H, S, C \vdash \mathbf{read}\,(e) \mapsto H(l), E} \text{ EREAD2}$$

$$\frac{\begin{array}{c}M, L, H, S, C \vdash e_1 \mapsto l, E_1 \\ M, L, H, S', C \vdash e_2 \mapsto v, E_2\end{array}}{M, L, H, S, C \vdash \mathbf{write}\,(e_1, e_2) \mapsto l, E} \text{ EWRITE}$$

$$\frac{\begin{array}{c}M, L, H, S, C \vdash e_1 \mapsto l, E_1 \\ M, L, H, S', C \vdash e_2 \mapsto v, E_2\end{array}}{M, L, H, S, C \vdash \mathbf{reduce}\,(id, e_1, e_2) \mapsto l, E} \text{ EREDUCE}$$

$$\frac{}{M, L, H, S, C \vdash \mathbf{null} \mapsto \mathbf{null}, []} \text{ ENULL}$$

$$\frac{}{M, L, H, S, C \vdash \mathbf{new}\ T@r \mapsto l, []} \text{ ENEW}$$

$$\frac{}{M, L, H, S, C \vdash \mathbf{null}\ T@\mathbf{nullr} \mapsto \mathbf{true}, []} \text{ ENULLTRUE}$$

$$\frac{\neg\,(T@r = T@\mathbf{nullr})}{M, L, H, S, C \vdash \mathbf{null}\ T@r \mapsto \mathbf{false}, []} \text{ ENULLFALSE}$$

$$\frac{\begin{array}{c}M, L, H, S, C \vdash e_1 \mapsto l, E_1 \\ l = \mathbf{null}\end{array}}{M, L, H, S, C \vdash \mathbf{isnull}\,(l) \mapsto \mathbf{true}, []} \text{ EISNULLTRUE}$$

$$\frac{\begin{array}{c}M, L, H, S, C \vdash e_1 \mapsto l, E_1 \\ \neg\,(l = \mathbf{null})\end{array}}{M, L, H, S, C \vdash \mathbf{isnull}\,(l) \mapsto \mathbf{false}, []} \text{ EISNULLFALSE}$$

$$\frac{M, L, H, S, C \vdash e \mapsto v, E}{M, L, H, S, C \vdash \mathbf{upregion}\,(e, r_1, \ldots, r_n) \mapsto v, E} \text{ EUPRGN}$$

$$\frac{M, L, H, S, C \vdash e \mapsto v, E}{M, L, H, S, C \vdash \mathbf{downregion}\,(e, r_1, \ldots, r_n) \mapsto l, E} \text{ EDNRGN1}$$

$$\frac{M, L, H, S, C \vdash e \mapsto v, E}{M, L, H, S, C \vdash \mathbf{downregion}\,(e, r_1, \ldots, r_n) \mapsto \mathbf{null}, E} \text{ EDNRGN2}$$

$$\frac{\begin{array}{c}M, L, H, S, C \vdash e_1 \mapsto l, E_1 \\ M, L, H, S', C \vdash e_2 \mapsto v, E_2 \\ M, L, H, S'', C \vdash e_3 \mapsto v, E_3\end{array}}{M, L, H, S, C \vdash \mathbf{write}\,(e_1, e_2) \mapsto l, E} \text{ ECOLOR}$$

$$\frac{\begin{array}{c}M, L, H, S, C \vdash e_1 \mapsto K, E_1 \\ M' = M[\rho_1/r_1, \ldots, \rho_k/r_k] \\ M, L, H, S', C \vdash e_2 \mapsto v, E_2\end{array}}{M, L, H, S, C \vdash \mathbf{partition}\ r_p\ \mathbf{using}\ e_1\ \mathbf{as}\ r_1, \ldots, r_k\ \in\ e_2 \mapsto l, E'} \text{ EPARTITION}$$

$$\frac{M, L, H, S, C \vdash e_1 \mapsto K, E_1}{M, L, H, S, C \vdash \mathbf{pack}\ e_1\ \mathbf{as}\ T_1[r_1, \ldots, r_k] \mapsto v', E} \text{ EPACK}$$

$$\frac{\begin{array}{l} M, L, H, S, C \vdash e_1 \mapsto \langle\langle \rho_1, .., \rho_k, v\rangle\rangle, E_1 \\ M' = M[\rho_1/r_1, .., \rho_k/r_k] \\ L' = L[v_1/id] \\ M', L', H, S', C \vdash e_2 \mapsto v, E_2 \end{array}}{M, L, H, S, C \vdash \textbf{unpack}\, e_1\, \textbf{as}\, id : T_1[r_1, .., r_k] \in e_2 \mapsto v_2, E'} \text{EUnpack}$$

$$\frac{\begin{array}{l} L' = L[(e_1, v_1), .., (e_n, v_n)] \\ x = \textbf{taskid}\, fresh \\ valid\_interleave(S, C, E', E_1, E_2) \end{array}}{M, L, H, S, C \vdash id[r_1, .., r_k](e_1, .., e_n) \mapsto v_n, E''} \text{ECall}$$

$$\frac{\begin{array}{l} M, L, H, S, C \vdash e_1 \mapsto v_1, E_1 \\ M, L, H, S, C \vdash e_2 \mapsto v_2, E_2 \\ v = v_1 + v_2 \end{array}}{M, L, H, S, C \vdash e_1 + e_2 \mapsto v, E} \text{EIntOp}$$

$$\frac{\begin{array}{l} M, L, H, S, C \vdash e_1 \mapsto v_1, E_1 \\ M, L, H, S, C \vdash e_2 \mapsto v_2, E_2 \\ v = v_1 < v_2 \end{array}}{M, L, H, S, C \vdash e_1 < e_2 \mapsto v, E} \text{EComp}$$

$$\frac{\begin{array}{l} M, L, H, S, C \vdash e_1 \mapsto v_1, E_1 \\ M, L, H, S, C \vdash e_2 \mapsto v_2, E_2 \\ L' = L[v_1/id] \end{array}}{M, L, H, S, C \vdash \textbf{let}\, id : T = e_1 \in e_2 \mapsto v, E} \text{ELet}$$

$$\frac{\begin{array}{l} M, L, H, S, C \vdash e_1 \mapsto \textbf{true}, E_1 \\ M, L, H, S, C \vdash e_2 \mapsto v, E_1 \end{array}}{M, L, H, S, C \vdash \textbf{if}\, e_1\, \textbf{then}\, e_2\, \textbf{else}\, e_3 \mapsto v, E} \text{EIfTrue}$$

$$\frac{\begin{array}{l} M, L, H, S, C \vdash e_1 \mapsto \textbf{false}, E_1 \\ M, L, H, S, C \vdash e_3 \mapsto v, E_2 \end{array}}{M, L, H, S, C \vdash \textbf{if}\, e_1\, \textbf{then}\, e_2\, \textbf{else}\, e_3 \mapsto v, E} \text{EIfFalse}$$

$$\frac{L' = L[id[r_1, .., r_n](e_1, .., e_m)/id]}{M, L, H, S, C \vdash \textbf{function}\, id[r_1, .., r_n](e_1, .., e_m) \mapsto v, E} \text{EFuncDef}$$

$$\frac{}{M, L, H, S, C \vdash \{\, \textbf{function}\, id_1\, rr_1\, ee_1, .., \textbf{function}\, id_n\, rr_n\, ee_n\} \mapsto v, E} \text{EFuncDefList}$$

$$\frac{M, L, H, S, C \vdash e \mapsto v, E}{M, L, H, S, C \vdash e : T \mapsto v, E} \text{ETypedExpr}$$

$$\frac{}{M, L, H, S, C \vdash \textbf{place} \mapsto \textbf{place}, E} \text{EPlace}$$

$$\frac{}{M, L, H, S, C \vdash bv \mapsto bv, E} \text{Ebv}$$

$$\frac{}{M, L, H, S, C \vdash iv \mapsto iv, E} \text{Eiv}$$

$$\frac{}{M, L, H, S, C \vdash \langle v_1, v_2\rangle \mapsto \langle v_1, v_2\rangle, E} \text{ETuple}$$

$$\frac{}{M, L, H, S, C \vdash \textbf{null} \mapsto \textbf{null}, E} \text{ENullLoc}$$

$$\frac{}{M, L, H, S, C \vdash l \mapsto l, E} \text{EMemoryLoc}$$

$$\frac{}{M, L, H, S, C \vdash \langle\langle \rho_1, .., \rho_n, v\rangle\rangle \mapsto \langle\langle \rho_1, .., \rho_n, v\rangle\rangle, E} \text{ERegRelInst}$$

$$\frac{}{M, L, H, S, C \vdash H \mapsto H, E}\ \textsc{EHeapVal}$$

$$\frac{}{M, L, H, S, C \vdash K \mapsto K, E}\ \textsc{EKthing}$$

```
Definition rules:          49 good    0 bad
Definition rule clauses:  107 good    0 bad
```