$ident$, x, y, y_p, y_f, _, $abbrev$, r	subscripts: p for pointers, f for functions
n, i, j	index variables
$impl\_const$	implementation-defined constant
$member$	C struct/union member name
	Ott-hack, ignore (annotations)
nat	OCaml arbitrary-width natural number
mem_ptr	abstract pointer value
mem_val	abstract memory value
	Ott-hack, ignore (locations)
mem_iv_c	OCaml type for memory constraints on integer values
UB_name	undefined behaviour
$string$	OCaml string
	Ott-hack, ignore (OCaml type variable TY)
	Ott-hack, ignore (OCaml Symbol.prefix)
mem_order, _	OCaml type for memory order
$linux\_mem\_order$	OCaml type for Linux memory order
	Ott-hack, ignore (OCaml type variable bt)

$Sctypes\_t,\ \tau$	$::=$			C type
	$\mid$	$\tau *$		pointer to type τ
tag	$::=$			OCaml type for struct/union tag
	$\mid$	$ident$		
$\beta,\ \_$	$::=$			base types
	$\mid$	`unit`		unit
	$\mid$	`bool`		boolean
	$\mid$	`integer`		integer
	$\mid$	`real`		rational numbers?
	$\mid$	`loc`		location
	$\mid$	`array` β		array
	$\mid$	`list` β		list
	$\mid$	$\overline{\beta_i}^{\,i}$		tuple
	$\mid$	`struct` tag		struct
	$\mid$	`set` β		set
	$\mid$	`opt` (β)		option
	$\mid$	$\beta \to \beta'$		parameter types
	$\mid$	β_τ	M	of a C type
$binop$	$::=$			binary operators
	$\mid$	`+`		addition
	$\mid$	`-`		subtraction
	$\mid$	`*`		multiplication
	$\mid$	`/`		division
	$\mid$	`rem_t`		modulus
	$\mid$	`rem_f`		remainder
	$\mid$	`^`		exponentiation
	$\mid$	`=`		equality, defined both for integer and C types

	$\mid$	!=	inequality, similiarly defined
	$\mid$	>	greater than, similarly defined
	$\mid$	<	less than, similarly defined
	$\mid$	>=	greater than or equal to, similarly defined
	$\mid$	<=	less than or equal to, similarly defined
	$\mid$	/\	conjucntion
	$\mid$	\/	disjunction
$binop_{arith}$	::=		arithmentic binary operators
	$\mid$	+	
	$\mid$	-	
	$\mid$	*	
	$\mid$	/	
	$\mid$	rem_t	
	$\mid$	rem_f	
	$\mid$	^	
$binop_{rel}$	::=		relational binary operators
	$\mid$	=	
	$\mid$	!=	
	$\mid$	>	
	$\mid$	<	
	$\mid$	>=	
	$\mid$	<=	
$binop_{bool}$	::=		boolean binary operators
	$\mid$	/\	
	$\mid$	\/	
mem_int	::=		memory integer value

	$\mid$	1	M
	$\mid$	0	M
$object\_value$	::=		C object values (inhabitants of object types), which can be read/stored
	$\mid$	mem_int	integer value
	$\mid$	mem_ptr	pointer value
	$\mid$	$\texttt{array}\,(\,\overline{loaded\_value_i}^{\,i}\,)$	C array value
	$\mid$	$(\,\texttt{struct}\,ident)\{\,\overline{.\,member_i{:}\tau_i = mem\_val_i}^{\,i}\,\}$	C struct value
	$\mid$	$(\,\texttt{union}\,ident)\{\,.\,member = mem\_val\}$	C union value
$loaded\_value$	::=		potentially unspecified C object values
	$\mid$	$\texttt{specified}\,object\_value$	specified loaded value
$value$	::=		Core values
	$\mid$	$object\_value$	C object value
	$\mid$	$loaded\_value$	loaded C object value
	$\mid$	$\texttt{Unit}$	unit
	$\mid$	$\texttt{True}$	boolean true
	$\mid$	$\texttt{False}$	boolean false
	$\mid$	$\beta[\,\overline{value_i}^{\,i}\,]$	list
	$\mid$	$(\,\overline{value_i}^{\,i}\,)$	tuple
$bool\_value$	::=		Core booleans
	$\mid$	$\texttt{True}$	boolean true
	$\mid$	$\texttt{False}$	boolean false
$ctor\_val$	::=		data constructors
	$\mid$	$\texttt{Nil}\,\beta$	empty list
	$\mid$	$\texttt{Cons}$	list cons
	$\mid$	$\texttt{Tuple}$	tuple

	|	`Array`	C array
	|	`Specified`	non-unspecified loaded value
ctor_expr	::=		data constructors
	|	`Ivmax`	max integer value
	|	`Ivmin`	min integer value
	|	`Ivsizeof`	sizeof value
	|	`Ivalignof`	alignof value
	|	`IvCOMPL`	bitwise complement
	|	`IvAND`	bitwise AND
	|	`IvOR`	bitwise OR
	|	`IvXOR`	bitwise XOR
	|	`Fvfromint`	cast integer to floating value
	|	`Ivfromfloat`	cast floating to integer value
name	::=		
	|	*ident*	Core identifier
	|	*impl_const*	implementation-defined constant
pval	::=		pure values
	|	*ident*	Core identifier
	|	*impl_const*	implementation-defined constant
	|	*value*	Core values
	|	$\texttt{constrained}\,(\overline{mem\_iv\_c_i, pval_i}^{\,i})$	constrained value
	|	$\texttt{error}\,(string, pval)$	impl-defined static error
	|	$ctor\_val(\overline{pval_i}^{\,i})$	data constructor application
	|	$(\texttt{struct}\,ident)\{\overline{.member_i = pval_i}^{\,i}\}$	C struct expression
	|	$(\texttt{union}\,ident)\{.member = pval\}$	C union expression
tpval	::=		top-level pure values

	|	**undef** UB_name		undefined behaviour
	|	**done** $pval$		pure done
$ident\_opt\_\beta$	::=			type annotated optional identifier
	|	$\_{:}\beta$	binders = $\{\}$	
	|	$ident{:}\beta$	binders = $ident$	
$pattern$	::=			
	|	$ident\_opt\_\beta$	binders = binders($ident\_opt\_\beta$)	
	|	$ctor\_val(\overline{pattern_i}^{\,i})$	binders = binders($\overline{pattern_i}^{\,i}$)	
z	::=			OCaml arbitrary-width integer
	|	i	M	literal integer
	|	mem_int	M	
	|	size_of(τ)	M	size of a C type
	|	offset_of$_{tag}$($member$)	M	offset of a struct member
	|	**ptr_size**	M	size of a pointer
	|	max_int$_\tau$	M	maximum value of int of type τ
	|	min_int$_\tau$	M	minimum value of int of type τ
$\mathbb{Q}$, q, $\_$	::=			OCaml type for rational numbers
	|	$\frac{int_1}{int_2}$		
lit	::=			
	|	$ident$		
	|	**unit**		
	|	$bool$		
	|	z		
	|	$\mathbb{Q}$		

$ident\_or\_pattern$	::=			
	|	$ident$	binders = $ident$	
	|	$pattern$	binders = binders($pattern$)	
$bool\_op$	::=			
	|	$\neg\, term$		
	|	$term_1 = term_2$		
	|	$\bigwedge(\overline{term_i}^{\,i})$		
	|	$\bigvee(\overline{term_i}^{\,i})$		
	|	$term_1\ binop_{bool}\ term_2$	M	
	|	$\mathtt{if}\ term_1\ \mathtt{then}\ term_2\ \mathtt{else}\ term_3$		
$arith\_op$	::=			
	|	$term_1 + term_2$		
	|	$term_1 - term_2$		
	|	$term_1 \times term_2$		
	|	$term_1 / term_2$		
	|	$term_1\ \mathtt{rem\_t}\ term_2$		
	|	$term_1\ \mathtt{rem\_f}\ term_2$		
	|	$term_1\ \hat{}\ term_2$		
	|	$term_1\ binop_{arith}\ term_2$	M	
cmp_op	::=			
	|	$term_1 < term_2$		less than
	|	$term_1 \leq term_2$		less than or equal
	|	$term_1\ binop_{rel}\ term_2$	M	
$list\_op$	::=			
	|	$\mathtt{nil}$		
	|	$term_1 :: term_2$		

	$\vert$	$\texttt{tl}\, term$
	$\vert$	$term^{(int)}$
$tuple\_op$	::=	
	$\vert$	$(\overline{term_i}^{\,i})$
	$\vert$	$term^{(int)}$
$pointer\_op$	::=	
	$\vert$	mem_ptr
	$\vert$	$term_1 +_{\mathrm{ptr}} term_2$
	$\vert$	$\texttt{cast\_int\_to\_ptr}\, term$
	$\vert$	$\texttt{cast\_ptr\_to\_int}\, term$
$array\_op$	::=	
	$\vert$	$[\vert\ \overline{term_i}^{\,i}\ \vert]$
	$\vert$	$term_1[term_2]$
$param\_op$	::=	
	$\vert$	$ident{:}\beta.\, term$
	$\vert$	$term(term_1, \ldots, term_n)$
$struct\_op$	::=	
	$\vert$	$term.member$
ct_pred	::=	
	$\vert$	$\texttt{representable}\,(\tau, term)$
	$\vert$	$\texttt{aligned}\,(\tau, term)$
	$\vert$	$\texttt{alignedI}\,(term_1, term_2)$
$term,\ \_$	::=	

	$\vert$	lit		
	$\vert$	$\mathit{arith\_op}$		
	$\vert$	$\mathit{bool\_op}$		
	$\vert$	$\mathit{cmp\_op}$		
	$\vert$	$\mathit{tuple\_op}$		
	$\vert$	$\mathit{struct\_op}$		
	$\vert$	$\mathit{pointer\_op}$		
	$\vert$	$\mathit{list\_op}$		
	$\vert$	$\mathit{array\_op}$		
	$\vert$	$\mathit{ct\_pred}$		
	$\vert$	$\mathit{param\_op}$		
	$\vert$	(term)	S	parentheses
	$\vert$	$\sigma(\mathit{term})$	M	simul-sub σ in term
	$\vert$	pval	M	
pexpr	$::=$			pure expressions
	$\vert$	pval		pure values
	$\vert$	$\mathit{ctor\_expr}(\,\overline{\mathit{pval}_i}^{\,i}\,)$		data constructor application
	$\vert$	$\texttt{array\_shift}\,(\mathit{pval}_1, \tau, \mathit{pval}_2)$		pointer array shift
	$\vert$	$\texttt{member\_shift}\,(\mathit{pval}, \mathit{ident}, \mathit{member})$		pointer struct/union member shift
	$\vert$	$\texttt{not}\,(\mathit{pval})$		boolean not
	$\vert$	$\mathit{pval}_1\ \mathit{binop}\ \mathit{pval}_2$		binary operations
	$\vert$	$\texttt{memberof}\,(\mathit{ident}, \mathit{member}, \mathit{pval})$		C struct/union member access
	$\vert$	$\mathit{name}(\,\overline{\mathit{pval}_i}^{\,i}\,)$		pure function call
	$\vert$	$\texttt{assert\_undef}\,(\mathit{pval},\ \mathit{UB\_name})$		
	$\vert$	$\texttt{bool\_to\_integer}\,(\mathit{pval})$		
	$\vert$	$\texttt{conv\_int}\,(\tau, \mathit{pval})$		
	$\vert$	$\texttt{wrapI}\,(\tau, \mathit{pval})$		
tpexpr	$::=$			top-level pure expressions

	$\mid$	$tpval$		top-level pure values
	$\mid$	$\mathtt{case}\ pval\ \mathtt{of}\ \overline{\mid tpexpr\_case\_branch_i}^{i}\ \mathtt{end}$		pattern matching
	$\mid$	$\mathtt{let}\ ident\_or\_pattern = pexpr\ \mathtt{in}\ tpexpr$	bind binders($ident\_or\_pattern$) in $tpexpr$	pure let
	$\mid$	$\mathtt{let}\ ident\_or\_pattern{:}(y_1{:}\beta_1.\ term_1) = tpexpr_1\ \mathtt{in}\ tpexpr_2$	bind binders($ident\_or\_pattern$) in $tpexpr_2$ bind y_1 in $term_1$	pure let
	$\mid$	$\mathtt{if}\ pval\ \mathtt{then}\ tpexpr_1\ \mathtt{else}\ tpexpr_2$		pure if
	$\mid$	$\sigma(tpexpr)$	M	simul-sub σ in $tpexpr$
$tpexpr\_case\_branch$	$::=$			pure top-level case expression
	$\mid$	$pattern \Rightarrow tpexpr$	bind binders($pattern$) in $tpexpr$	top-level case expression br
m_kill_kind	$::=$			
	$\mid$	$\mathtt{dynamic}$		
	$\mid$	$\mathtt{static}\ \tau$		
$bool$, _	$::=$			OCaml booleans
	$\mid$	$\mathtt{true}$		
	$\mid$	$\mathtt{false}$		
int, _	$::=$			OCaml fixed-width integer
	$\mid$	i		literal integer
res_term	$::=$			resource terms
	$\mid$	$\mathtt{emp}$		empty heap
	$\mid$	$points\_to$		single-cell heap
	$\mid$	$ident$		variable
	$\mid$	$\langle res\_term_1, res\_term_2\rangle$		seperating-conjunction pair
	$\mid$	$\mathtt{pack}\ (pval, res\_term)$		packing for existentials
	$\mid$	$\sigma(res\_term)$	M	substitution for resource te

mem_action	::=		memory actions
	|	`create` $(pval, \tau)$	
	|	`create_readonly` $(pval_1, \tau, pval_2)$	
	|	`alloc` $(pval_1, pval_2)$	
	|	`kill` $(m\_kill\_kind, pval, pt)$	
	|	`store` $(bool, \tau, pval_1, pval_2, mem\_order, pt)$	true means store is locking
	|	`load` $(\tau, pval, mem\_order, pt)$	
	|	`rmw` $(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$	
	|	`fence` (mem_order)	
	|	`cmp_exch_strong` $(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$	
	|	`cmp_exch_weak` $(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$	
	|	`linux_fence` $(linux\_mem\_order)$	
	|	`linux_load` $(\tau, pval, linux\_mem\_order)$	
	|	`linux_store` $(\tau, pval_1, pval_2, linux\_mem\_order)$	
	|	`linux_rmw` $(\tau, pval_1, pval_2, linux\_mem\_order)$	
$polarity$	::=		polarities for memory actions
	|		(pos) sequenced by `let weak` and `let strong`
	|	`neg`	only sequenced by `let strong`
pol_mem_action	::=		memory actions with polarity
	|	$polarity\ mem\_action$	
mem_op	::=		operations involving the memory state
	|	$pval_1\ \ binop_{rel}\ \ pval_2$	pointer relational binary operations
	|	$pval_1\ -_\tau\ pval_2$	pointer subtraction
	|	`intFromPtr` $(\tau_1, \tau_2, pval)$	cast of pointer value to integer value
	|	`ptrFromInt` $(\tau_1, \tau_2, pval)$	cast of integer value to pointer value
	|	`ptrValidForDeref` $(\tau, pval, pt)$	dereferencing validity predicate
	|	`ptrWellAligned` $(\tau, pval)$	

	$\mid$	$\texttt{ptrArrayShift}\,(pval_1, \tau, pval_2)$		
	$\mid$	$\texttt{memcpy}\,(pval_1, pval_2, pval_3)$		
	$\mid$	$\texttt{memcmp}\,(pval_1, pval_2, pval_3)$		
	$\mid$	$\texttt{realloc}\,(pval_1, pval_2, pval_3)$		
	$\mid$	$\texttt{va\_start}\,(pval_1, pval_2)$		
	$\mid$	$\texttt{va\_copy}\,(pval)$		
	$\mid$	$\texttt{va\_arg}\,(pval, \tau)$		
	$\mid$	$\texttt{va\_end}\,(pval)$		
$spine\_elem$	$::=$			spine element
	$\mid$	$pval$		pure or logical value
	$\mid$	res_term		resource value
	$\mid$	$\sigma(spine\_elem)$	M	substitution for spine elements / return values
$spine$	$::=$			spine
	$\mid$	$\overline{spine\_elem_i}^{\,i}$		
$tval$	$::=$			(effectful) top-level values
	$\mid$	$\texttt{done}\, spine$		end of top-level expression
	$\mid$	$\texttt{undef}\ UB\_name$		undefined behaviour
$res\_pattern$	$::=$			resource terms
	$\mid$	$\texttt{emp}$	binders $= \{\}$	empty heap
	$\mid$	pt	binders $= \{\}$	single-cell heap
	$\mid$	$ident$	binders $= ident$	variable
	$\mid$	$\langle res\_pattern_1, res\_pattern_2\rangle$	binders $=$ binders$(res_pattern_1) \cup$ binders$(res_pattern_2)$	seperating-conjunction pair
	$\mid$	$\texttt{pack}\,(ident, res\_pattern)$	binders $= ident \cup$ binders$(res_pattern)$	packing for existentials
$ret\_pattern$	$::=$			return pattern
	$\mid$	$\texttt{comp}\, ident\_or\_pattern$	binders $=$ binders$(ident_or_pattern)$	computational variable

	$\vert$	$\texttt{log}\, \mathit{ident}$	binders = ident	logical variable
	$\vert$	$\texttt{res}\, \mathit{res\_pattern}$	binders = binders($\mathit{res\_pattern}$)	resource variable
init,	::=			initialisation status
	$\vert$	$\checkmark$		initialised
	$\vert$	$\times$		uninitalised
$\mathit{points\_to}$, pt	::=			points-to separation logic predicate
	$\vert$	$\mathit{term}_1 \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2$		
res	::=			resources
	$\vert$	$\texttt{emp}$		empty heap
	$\vert$	$\mathit{points\_to}$		points-top heap pred.
	$\vert$	$\mathit{res}_1 * \mathit{res}_2$		seperating conjunction
	$\vert$	$\exists\, \mathit{ident}{:}\beta.\ \mathit{res}$		existential
	$\vert$	$\mathit{term} \wedge \mathit{res}$		logical conjuction
	$\vert$	$\sigma(\mathit{res})$	M	simul-sub σ in res
ret, _	::=			return types
	$\vert$	$\Sigma\, \mathit{ident}{:}\beta.\ \mathit{ret}$		return a computational value
	$\vert$	$\exists\, \mathit{ident}{:}\beta.\ \mathit{ret}$		return a logical value
	$\vert$	$\mathit{res} \otimes \mathit{ret}$		return a resource value
	$\vert$	$\mathit{term} \wedge \mathit{ret}$		return a predicate (post-condition)
	$\vert$	$\texttt{I}$		end return list
	$\vert$	$\sigma(\mathit{ret})$	M	simul-sub σ in ret
$\mathit{seq\_expr}$	::=			sequential (effectful) expressions
	$\vert$	$\texttt{ccall}\,(\tau, \mathit{pval}, \mathit{spine})$		C function call
	$\vert$	$\texttt{pcall}\,(\mathit{name}, \mathit{spine})$		procedure call

seq_texpr	$::=$			sequential top-level (effectful) expres
	$\mid$	$tval$		(effectful) top-level values
	$\mid$	$\mathtt{run}\ ident\ \overline{pval_i}^{\,i}$		run from label
	$\mid$	$\mathtt{let}\ ident\_or\_pattern = pexpr\ \mathtt{in}\ texpr$	bind binders($ident\_or\_pattern$) in $texpr$	pure let
	$\mid$	$\mathtt{let}\ ident\_or\_pattern{:}(y_1{:}\beta_1.\ term_1) = tpexpr\ \mathtt{in}\ texpr$	bind binders($ident\_or\_pattern$) in $texpr$ bind y_1 in $term_1$	pure let
	$\mid$	$\mathtt{let}\ \overline{ret\_pattern_i}^{\,i} = seq\_expr\ \mathtt{in}\ texpr$	bind binders($\overline{ret\_pattern_i}^{\,i}$) in $texpr$	bind return patterns
	$\mid$	$\mathtt{let}\ \overline{ret\_pattern_i}^{\,i}{:}ret = texpr_1\ \mathtt{in}\ texpr_2$	bind binders($\overline{ret\_pattern_i}^{\,i}$) in $texpr_2$	annotated bind return patterns
	$\mid$	$\mathtt{case}\ pval\ \mathtt{of}\ \overline{\mid texpr\_case\_branch_i}^{\,i}\ \mathtt{end}$		pattern matching
	$\mid$	$\mathtt{if}\ pval\ \mathtt{then}\ texpr_1\ \mathtt{else}\ texpr_2$		conditional
	$\mid$	$\mathtt{bound}\ [int](is\_texpr)$		limit scope of indet seq behaviour
$texpr\_case\_branch$	$::=$			top-level case expression branch
	$\mid$	$pattern \Rightarrow texpr$	bind binders($pattern$) in $texpr$	top-level case expression branch
is_expr	$::=$			indet seq (effectful) expressions
	$\mid$	$tval$		(effectful) top-level values
	$\mid$	$\mathtt{memop}\ (mem\_op)$		pointer op involving memory
	$\mid$	pol_mem_action		memory action
is_texpr	$::=$			indet seq top-level (effectful) expres
	$\mid$	$\mathtt{let}\ \mathtt{weak}\ \overline{ret\_pattern_i}^{\,i} = is\_expr\ \mathtt{in}\ texpr$	bind binders($\overline{ret\_pattern_i}^{\,i}$) in $texpr$	weak sequencing
	$\mid$	$\mathtt{let}\ \mathtt{strong}\ \overline{ret\_pattern_i}^{\,i} = is\_expr\ \mathtt{in}\ texpr$	bind binders($\overline{ret\_pattern_i}^{\,i}$) in $texpr$	strong sequencing
$texpr$	$::=$			top-level (effectful) expressions
	$\mid$	seq_texpr		sequential (effectful) expressions
	$\mid$	is_texpr		indet seq (effectful) expressions
	$\mid$	$\sigma(texpr)$	M	simul-sub σ in $texpr$
arg	$::=$			argument/function types

	$\mid$	$\Pi\, ident{:}\beta.\ arg$		
	$\mid$	$\forall\, ident{:}\beta.\ arg$		
	$\mid$	$res \multimap arg$		
	$\mid$	$term \supset arg$		
	$\mid$	ret		
	$\mid$	$\sigma(arg)$	M	simul-sub σ in arg
$pure\_arg$	$::=$			pure argument/function types
	$\mid$	$\Pi\, ident{:}\beta.\ pure\_arg$		
	$\mid$	$term \supset pure\_arg$		
	$\mid$	$pure\_ret$		
$pure\_ret$	$::=$			pure return types
	$\mid$	$\Sigma\, ident{:}\beta.\ pure\_ret$		
	$\mid$	$term \wedge pure\_ret$		
	$\mid$	$\mathtt{I}$		
$\mathcal{C}$	$::=$			computational var env
	$\mid$	$\cdot$		
	$\mid$	$\mathcal{C}, ident{:}\beta$		
	$\mid$	$\overline{\mathcal{C}_i}^{\,i}$		
$\mathcal{L}$	$::=$			logical var env
	$\mid$	$\cdot$		
	$\mid$	$\overline{\mathcal{L}_i}^{\,i}$		
	$\mid$	$\mathcal{L}, ident{:}\beta$		
Φ	$::=$			constraints env
	$\mid$	$\cdot$		
	$\mid$	$\Phi, term$		

	$\vert$	$\overline{\Phi_i}^{\,i}$		
$\mathcal{R}$	$::=$			resources env
	$\vert$	$\cdot$		
	$\vert$	$\mathcal{R}, \mathit{res}$		
	$\vert$	$\mathcal{R}, \mathit{ident}{:}\mathit{res}$		
	$\vert$	$\overline{\mathcal{R}_i}^{\,i}$		
$\sigma,\ \psi$	$::=$			substitutions
	$\vert$	$\cdot$		
	$\vert$	$\mathit{spine\_elem}/\mathit{ident}, \sigma$		
	$\vert$	$\mathit{term}/\mathit{ident}, \sigma$		
	$\vert$	$\overline{\sigma_i}^{\,i}$		
	$\vert$	$\sigma(\psi)$	M	apply σ to all elements in ψ
typing	$::=$			
	$\vert$	$\texttt{smt}\,(\Phi \Rightarrow \mathit{term})$		
	$\vert$	$\mathit{ident}{:}\beta\ \in\ \mathcal{C}$		
	$\vert$	$\mathit{ident}{:}\beta\ \in\ \mathcal{L}$		
	$\vert$	$\mathit{ident}{:}\,\texttt{struct}\,\mathit{tag}\ \&\,\overline{\mathit{member}_i{:}\tau_i}^{\,i}\ \in\ \texttt{Globals}$		
	$\vert$	$\overline{\mathcal{C}_i; \mathcal{L}_i; \Phi_i \vdash \mathit{mem\_val}_i \Rightarrow \texttt{mem}\,\beta_i}^{\,i}$		dependent on memory object model
opsem	$::=$			
	$\vert$	$\forall\, i < j.\ \texttt{not}\,(\mathit{pattern}_i = \mathit{pval} \rightsquigarrow \sigma_i)$		
	$\vert$	$\texttt{fresh}\,(\mathit{mem\_ptr})$		
	$\vert$	term		
	$\vert$	$\mathit{pval}{:}\beta$		
$\mathit{formula}$	$::=$			
	$\vert$	$\mathit{judgement}$		

	$\vert$	$typing$	
	$\vert$	$opsem$	
	$\vert$	$term \equiv term'$	
	$\vert$	$name{:}pure\_arg \equiv \overline{x_i}^{\,i} \mapsto tpexpr \in \texttt{Globals}$	
	$\vert$	$pval{:}arg \equiv \overline{x_i}^{\,i} \mapsto texpr \in \texttt{Globals}$	
$heap,\ h$	$::=$		heaps
	$\vert$	$\cdot$	
	$\vert$	$h + \{points\_to\}$	
$object\_value\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash object\_value \Rightarrow \texttt{obj}\,\beta$	
$pval\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash pval \Rightarrow \beta$	
res_jtype	$::=$		
	$\vert$	$\Phi \vdash res \equiv res'$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash res\_term \Leftarrow res$	
$spine\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret$	
$pexpr\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash pexpr \Rightarrow ident{:}\beta.\ term$	
$comp\_pattern\_jtype$	$::=$		
	$\vert$	$pattern{:}\beta \rightsquigarrow \mathcal{C}\ \texttt{with}\ term$	
	$\vert$	$ident\_or\_pattern{:}\beta \rightsquigarrow \mathcal{C}\ \texttt{with}\ term$	

$$
\begin{array}{lcl}
\mathit{res\_pattern\_jtype} & ::= & \\
 & | & \mathit{res\_pattern}{:}\mathit{res} \rightsquigarrow \mathcal{L}; \Phi; \mathcal{R} \\[2ex]
\mathit{ret\_pattern\_jtype} & ::= & \\
 & | & \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \\[2ex]
\mathit{tpval\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpval} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term} \\[2ex]
\mathit{tpexpr\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpexpr} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term} \\[2ex]
\mathit{action\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_action} \Rightarrow \mathit{ret} \\[2ex]
\mathit{memop\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_op} \Rightarrow \mathit{ret} \\[2ex]
\mathit{tval\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{tval} \Leftarrow \mathit{ret} \\[2ex]
\mathit{seq\_expr\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_expr} \Rightarrow \mathit{ret} \\[2ex]
\mathit{is\_expr\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_expr} \Rightarrow \mathit{ret} \\[2ex]
\mathit{texpr\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret} \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}
\end{array}
$$

$$
\begin{array}{lll}
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{texpr} \Leftarrow \mathit{ret} \\
\\
\mathit{subs\_jtype} & ::= & \\
 & | & \mathit{pattern} = \mathit{pval} \rightsquigarrow \sigma \\
 & | & \mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma \\
 & | & \mathit{res\_pattern} = \mathit{res\_term} \rightsquigarrow \sigma \\
 & | & \overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \rightsquigarrow \sigma \\
 & | & \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret} \\
\\
\mathit{pure\_opsem\_jtype} & ::= & \\
 & | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{pexpr}' \rangle \\
 & | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\, \mathit{term}) \rangle \\
 & | & \langle \mathit{tpexpr} \rangle \longrightarrow \langle \mathit{tpexpr}' \rangle \\
\\
\mathit{opsem\_jtype} & ::= & \\
 & | & \langle h; \mathit{seq\_expr} \rangle \longrightarrow \langle h'; \mathit{texpr}{:}\mathit{ret} \rangle \\
 & | & \langle h; \mathit{seq\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
 & | & \langle h; \mathit{mem\_op} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
 & | & \langle h; \mathit{mem\_action} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
 & | & \langle h; \mathit{is\_expr} \rangle \longrightarrow \langle h'; \mathit{is\_expr}' \rangle \\
 & | & \langle h; \mathit{is\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
 & | & \langle h; \mathit{texpr} \rangle \longrightarrow \langle h'; \mathit{texpr}' \rangle \\
\\
\mathit{lemma\_jtype} & ::= & \\
 & | & \overline{x_i}^{\,i} :: \mathit{arg} \rightsquigarrow \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \mid \mathit{ret} \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \sqsubseteq \mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}' \\
 & | & \mathcal{C};\mathcal{L};\mathcal{R} \vdash (\sigma){:}(\mathcal{C}';\mathcal{L}';\mathcal{R}')
\end{array}
$$

$\boxed{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \texttt{obj}\ \beta}$

$$
\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{mem\_int} \Rightarrow \texttt{obj}\ \texttt{integer}} \quad \textsc{Ty\_Pval\_Obj\_Int}
$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{mem\_ptr} \Rightarrow \texttt{obj}\,\texttt{loc}} \quad \textsc{Ty\_Pval\_Obj\_Ptr}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{loaded\_value}_i \Rightarrow \beta}^{i}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{array}\,(\,\overline{\mathit{loaded\_value}_i}^{i}\,) \Rightarrow \texttt{obj}\,\texttt{array}\,\beta} \quad \textsc{Ty\_Pval\_Obj\_Arr}$$

$$\frac{\begin{array}{l}\mathit{ident}{:}\,\texttt{struct}\,\mathit{tag}\ \&\,\overline{\mathit{member}_i{:}\tau_i}^{i} \in \texttt{Globals}\\ \overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{mem\_val}_i \Rightarrow \texttt{mem}\,\beta_{\tau_i}}^{i}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash (\,\texttt{struct}\,\mathit{tag})\{\,\overline{.\,\mathit{member}_i{:}\tau_i = \mathit{mem\_val}_i}^{i}\,\} \Rightarrow \texttt{obj}\,\texttt{struct}\,\mathit{tag}} \quad \textsc{Ty\_Pval\_Obj\_Struct}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta}$

$$\frac{x{:}\beta\ \in\ \mathcal{C}}{\mathcal{C};\mathcal{L};\Phi \vdash x \Rightarrow \beta} \quad \textsc{Ty\_Pval\_Var\_Comp}$$

$$\frac{x{:}\beta\ \in\ \mathcal{L}}{\mathcal{C};\mathcal{L};\Phi \vdash x \Rightarrow \beta} \quad \textsc{Ty\_Pval\_Var\_Log}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \texttt{obj}\,\beta}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \beta} \quad \textsc{Ty\_Pval\_Obj}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \texttt{obj}\,\beta}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{specified}\,\mathit{object\_value} \Rightarrow \beta} \quad \textsc{Ty\_Pval\_Loaded}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Unit} \Rightarrow \texttt{unit}} \quad \textsc{Ty\_Pval\_Unit}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{True} \Rightarrow \texttt{bool}} \quad \textsc{Ty\_Pval\_True}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{False} \Rightarrow \texttt{bool}} \quad \text{TY\_PVAL\_FALSE}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash value_i \Rightarrow \beta}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash \beta[\,\overline{value_i}^{\,i}\,] \Rightarrow \texttt{list}\,\beta} \quad \text{TY\_PVAL\_LIST}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash value_i \Rightarrow \beta_i}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash (\,\overline{value_i}^{\,i}\,) \Rightarrow \overline{\beta_i}^{\,i}} \quad \text{TY\_PVAL\_TUPLE}$$

$$\frac{\texttt{smt}\,(\Phi \Rightarrow \texttt{false})}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{error}\,(string, pval) \Rightarrow \beta} \quad \text{TY\_PVAL\_ERROR}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Nil}\,\beta(\,) \Rightarrow \texttt{list}\,\beta} \quad \text{TY\_PVAL\_CTOR\_NIL}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{list}\,\beta\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Cons}(pval_1, pval_2) \Rightarrow \texttt{list}\,\beta} \quad \text{TY\_PVAL\_CTOR\_CONS}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash pval_i \Rightarrow \beta_i}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Tuple}(\,\overline{pval_i}^{\,i}\,) \Rightarrow \overline{\beta_i}^{\,i}} \quad \text{TY\_PVAL\_CTOR\_TUPLE}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash pval_i \Rightarrow \beta}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Array}(\,\overline{pval_i}^{\,i}\,) \Rightarrow \texttt{array}\,\beta} \quad \text{TY\_PVAL\_CTOR\_ARRAY}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \beta}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Specified}(pval) \Rightarrow \beta} \quad \text{TY\_PVAL\_CTOR\_SPECIFIED}$$

$$\frac{\begin{array}{l} \mathit{ident}\mathord{:}\,\texttt{struct}\,\mathit{tag}\;\&\;\overline{\mathit{member}_i\mathord{:}\tau_i}^{\,i} \;\in\; \texttt{Globals} \\ \overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_i \Rightarrow \beta_{\tau_i}}^{\,i} \end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash (\,\texttt{struct}\,\mathit{tag})\{\,\overline{.\,\mathit{member}_i = \mathit{pval}_i}^{\,i}\,\} \Rightarrow \texttt{struct}\,\mathit{tag}} \quad \textsc{Ty\_Pval\_Struct}$$

$\boxed{\Phi \vdash \mathit{res} \;\equiv\; \mathit{res}'}$

$$\frac{}{\Phi \vdash \texttt{emp} \;\equiv\; \texttt{emp}} \quad \textsc{Ty\_Res\_Eq\_Emp}$$

$$\frac{\texttt{smt}\,(\Phi \Rightarrow (\mathit{term}_1 = \mathit{term}_1') \wedge (\mathit{term}_2 = \mathit{term}_2'))}{\Phi \vdash \mathit{term}_1 \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2 \;\equiv\; \mathit{term}_1' \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2'} \quad \textsc{Ty\_Res\_Eq\_PointsTo}$$

$$\frac{\begin{array}{l} \Phi \vdash \mathit{res}_1 \;\equiv\; \mathit{res}_1' \\ \Phi \vdash \mathit{res}_2 \;\equiv\; \mathit{res}_2' \end{array}}{\Phi \vdash \mathit{res}_1 * \mathit{res}_2 \;\equiv\; \mathit{res}_1' * \mathit{res}_2'} \quad \textsc{Ty\_Res\_Eq\_SepConj}$$

$$\frac{\Phi \vdash \mathit{res} \;\equiv\; \mathit{res}'}{\Phi \vdash \exists\,\mathit{ident}\mathord{:}\beta.\;\mathit{res} \;\equiv\; \exists\,\mathit{ident}\mathord{:}\beta.\;\mathit{res}'} \quad \textsc{Ty\_Res\_Eq\_Exists}$$

$$\frac{\begin{array}{l} \texttt{smt}\,(\Phi, \mathit{term} \Rightarrow \mathit{term}') \\ \texttt{smt}\,(\Phi, \mathit{term}' \Rightarrow \mathit{term}) \\ \Phi \vdash \mathit{res} \;\equiv\; \mathit{res}' \end{array}}{\Phi \vdash \mathit{term} \wedge \mathit{res} \;\equiv\; \mathit{term}' \wedge \mathit{res}'} \quad \textsc{Ty\_Res\_Eq\_Term}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{res}}$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{emp} \Leftarrow \texttt{emp}} \quad \textsc{Ty\_Res\_Emp}$$

$$\frac{\Phi \vdash \mathit{points\_to} \ \equiv \ \mathit{points\_to}'}{\mathcal{C};\mathcal{L};\Phi;\cdot,\mathit{points\_to} \vdash \mathit{points\_to}' \Leftarrow \mathit{points\_to}'} \quad \textsc{Ty\_Res\_PointsTo}$$

$$\frac{\Phi \vdash \mathit{res} \ \equiv \ \mathit{res}'}{\mathcal{C};\mathcal{L};\Phi;\cdot,r{:}\mathit{res} \vdash r \Leftarrow \mathit{res}'} \quad \textsc{Ty\_Res\_Var}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1 \vdash \mathit{res\_term}_1 \Leftarrow \mathit{res}_1 \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}_2 \vdash \mathit{res\_term}_2 \Leftarrow \mathit{res}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1,\mathcal{R}_2 \vdash \langle \mathit{res\_term}_1, \mathit{res\_term}_2\rangle \Leftarrow \mathit{res}_1 * \mathit{res}_2} \quad \textsc{Ty\_Res\_SepConj}$$

$$\frac{\begin{array}{l}\mathtt{smt}\,(\Phi \Rightarrow \mathit{term}) \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{res}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{term} \wedge \mathit{res}} \quad \textsc{Ty\_Res\_Conj}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term}_2 \Leftarrow \mathit{pval}/y,\cdot(\mathit{res})\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{pack}\,(\mathit{pval}, \mathit{res\_term}_2) \Leftarrow \exists\, y{:}\beta.\ \mathit{res}} \quad \textsc{Ty\_Res\_Pack}$$

$$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma;\mathit{ret}}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \ ::\mathit{ret} \gg \cdot;\mathit{ret}} \quad \textsc{Ty\_Spine\_Empty}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma;\mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash x = \mathit{pval},\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \Pi\, x{:}\beta.\ \mathit{arg} \gg \mathit{pval}/x,\sigma;\mathit{ret}} \quad \textsc{Ty\_Spine\_Comp}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma;\mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash x = \mathit{pval},\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \forall\, x{:}\beta.\ \mathit{arg} \gg \mathit{pval}/x,\sigma;\mathit{ret}} \quad \textsc{Ty\_Spine\_Log}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1 \vdash \mathit{res\_term} \Leftarrow \mathit{res} \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}_2 \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1,\mathcal{R}_2 \vdash x = \mathit{res\_term},\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{res} \multimap \mathit{arg} \gg \mathit{res\_term}/x, \sigma; \mathit{ret}} \quad \textsc{Ty\_Spine\_Res}$$

$$\frac{\begin{array}{c}\mathtt{smt}\,(\Phi \Rightarrow \mathit{term}) \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{term} \supset \mathit{arg} \gg \sigma; \mathit{ret}} \quad \textsc{Ty\_Spine\_Phi}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pexpr} \Rightarrow \mathit{ident}{:}\beta.\ \mathit{term}}$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow y{:}\beta.\ y = \mathit{pval}} \quad \textsc{Ty\_PE\_Val}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \mathtt{loc} \\ \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_2 \Rightarrow \mathtt{integer}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{array\_shift}\,(\mathit{pval}_1, \tau, \mathit{pval}_2) \Rightarrow y{:}\mathtt{loc}.\ y = \mathit{pval}_1 +_{\mathrm{ptr}} (\mathit{pval}_2 \times \mathrm{size\_of}(\tau))} \quad \textsc{Ty\_PE\_Array\_Shift}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \mathtt{loc} \\ \_{:}\,\mathtt{struct}\,\mathit{tag}\ \&\,\overline{\mathit{member}_i{:}\tau_i}^{\,i} \in \mathtt{Globals}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{member\_shift}\,(\mathit{pval}, \mathit{tag}, \mathit{member}_j) \Rightarrow y{:}\mathtt{loc}.\ y = \mathit{pval} +_{\mathrm{ptr}} \mathrm{offset\_of}_{\mathit{tag}}(\mathit{member}_j)} \quad \textsc{Ty\_PE\_Member\_Shift}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \mathtt{bool}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{not}\,(\mathit{pval}) \Rightarrow y{:}\mathtt{bool}.\ y = \neg\,\mathit{pval}} \quad \textsc{Ty\_PE\_Not}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \mathtt{integer} \\ \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_2 \Rightarrow \mathtt{integer}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1\ \mathit{binop}_{\mathit{arith}}\ \mathit{pval}_2 \Rightarrow y{:}\mathtt{integer}.\ y = (\mathit{pval}_1\ \mathit{binop}_{\mathit{arith}}\ \mathit{pval}_2)} \quad \textsc{Ty\_PE\_Arith\_Binop}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \mathtt{integer} \qquad \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_2 \Rightarrow \mathtt{integer}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1\ \mathit{binop}_{\mathit{rel}}\ \mathit{pval}_2 \Rightarrow y{:}\mathtt{bool}.\ y = (\mathit{pval}_1\ \mathit{binop}_{\mathit{rel}}\ \mathit{pval}_2)} \quad \textsc{Ty\_PE\_Rel\_Binop}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \mathtt{bool} \qquad \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_2 \Rightarrow \mathtt{bool}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1\ \mathit{binop}_{\mathit{bool}}\ \mathit{pval}_2 \Rightarrow y{:}\mathtt{bool}.\ y = (\mathit{pval}_1\ \mathit{binop}_{\mathit{bool}}\ \mathit{pval}_2)} \quad \textsc{Ty\_PE\_Bool\_Binop}$$

$$\frac{\mathit{name}{:}\mathit{pure\_arg} \equiv \overline{x_i}^{\,i} \mapsto \mathit{tpexpr} \in \mathtt{Globals} \qquad \mathcal{C};\mathcal{L};\Phi;\cdot \vdash \overline{x_i = \mathit{pval}_i}^{\,i} :: \mathit{pure\_arg} \gg \sigma; \Sigma\, y{:}\beta.\ \mathit{term} \wedge \mathtt{I}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{name}(\,\overline{\mathit{pval}_i}^{\,i}\,) \Rightarrow y{:}\beta.\ \sigma(\mathit{term})} \quad \textsc{Ty\_PE\_Call}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \mathtt{bool} \qquad \mathtt{smt}\,(\Phi \Rightarrow \mathit{pval})}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{assert\_undef}\,(\mathit{pval},\ \mathit{UB\_name}) \Rightarrow y{:}\mathtt{unit}.\ y = \mathtt{unit}} \quad \textsc{Ty\_PE\_Assert\_Undef}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \mathtt{bool}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{bool\_to\_integer}\,(\mathit{pval}) \Rightarrow y{:}\mathtt{integer}.\ y = \mathtt{if}\ \mathit{pval}\ \mathtt{then}\ 1\ \mathtt{else}\ 0} \quad \textsc{Ty\_PE\_Bool\_To\_Integer}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \mathtt{integer} \qquad \mathit{abbrev}_1 \equiv \mathrm{max\_int}_\tau - \mathrm{min\_int}_\tau + 1 \qquad \mathit{abbrev}_2 \equiv \mathit{pval}\ \mathtt{rem\_f}\ \mathit{abbrev}_1}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{wrapI}\,(\tau, \mathit{pval}) \Rightarrow y{:}\beta.\ y = \mathtt{if}\ \mathit{abbrev}_2 \leq \mathrm{max\_int}_\tau\ \mathtt{then}\ \mathit{abbrev}_2\ \mathtt{else}\ \mathit{abbrev}_2 - \mathit{abbrev}_1} \quad \textsc{Ty\_PE\_WrapI}$$

$\boxed{\mathit{pattern}{:}\beta \rightsquigarrow \mathcal{C}\ \mathtt{with}\ \mathit{term}}$

$$\frac{}{\_{:}\beta{:}\beta \rightsquigarrow \cdot\ \mathtt{with}\ \_} \quad \textsc{Ty\_Pat\_Comp\_No\_Sym\_Annot}$$

$$\frac{}{x{:}\beta{:}\beta \rightsquigarrow \cdot, x{:}\beta\ \texttt{with}\ x}\quad \textsc{Ty\_Pat\_Comp\_Sym\_Annot}$$

$$\frac{}{\texttt{Nil}\ \beta(\,){:}\texttt{list}\ \beta \rightsquigarrow \cdot\ \texttt{with}\ \texttt{nil}}\quad \textsc{Ty\_Pat\_Comp\_Nil}$$

$$\frac{\begin{array}{c} pattern_1{:}\beta \rightsquigarrow \mathcal{C}_1\ \texttt{with}\ term_1 \\ pattern_2{:}\texttt{list}\ \beta \rightsquigarrow \mathcal{C}_2\ \texttt{with}\ term_2 \end{array}}{\texttt{Cons}(pattern_1, pattern_2){:}\texttt{list}\ \beta \rightsquigarrow \mathcal{C}_1, \mathcal{C}_2\ \texttt{with}\ term_1 :: term_2}\quad \textsc{Ty\_Pat\_Comp\_Cons}$$

$$\frac{\overline{pattern_i{:}\beta_i \rightsquigarrow \mathcal{C}_i\ \texttt{with}\ term_i}^{\,i}}{\texttt{Tuple}(\overline{pattern_i}^{\,i}){:}\overline{\beta_i}^{\,i} \rightsquigarrow \overline{\mathcal{C}_i}^{\,i}\ \texttt{with}\ (\overline{term_i}^{\,i})}\quad \textsc{Ty\_Pat\_Comp\_Tuple}$$

$$\frac{\overline{pattern_i{:}\beta \rightsquigarrow \mathcal{C}_i\ \texttt{with}\ term_i}^{\,i}}{\texttt{Array}(\overline{pattern_i}^{\,i}){:}\texttt{array}\ \beta \rightsquigarrow \overline{\mathcal{C}_i}^{\,i}\ \texttt{with}\ [|\ \overline{term_i}^{\,i}\ |]}\quad \textsc{Ty\_Pat\_Comp\_Array}$$

$$\frac{pattern{:}\beta \rightsquigarrow \mathcal{C}\ \texttt{with}\ term}{\texttt{Specified}(pattern){:}\beta \rightsquigarrow \mathcal{C}\ \texttt{with}\ term}\quad \textsc{Ty\_Pat\_Comp\_Specified}$$

$\boxed{ident\_or\_pattern{:}\beta \rightsquigarrow \mathcal{C}\ \texttt{with}\ term}$

$$\frac{}{x{:}\beta \rightsquigarrow \cdot, x{:}\beta\ \texttt{with}\ x}\quad \textsc{Ty\_Pat\_Sym\_Or\_Pattern\_Sym}$$

$$\frac{pattern{:}\beta \rightsquigarrow \mathcal{C}\ \texttt{with}\ term}{pattern{:}\beta \rightsquigarrow \mathcal{C}\ \texttt{with}\ term}\quad \textsc{Ty\_Pat\_Sym\_Or\_Pattern\_Pattern}$$

$\boxed{res\_pattern{:}res \rightsquigarrow \mathcal{L}; \Phi; \mathcal{R}}$

$$\frac{}{\texttt{emp}{:}\texttt{emp} \rightsquigarrow \cdot; \cdot; \cdot}\quad \textsc{Ty\_Pat\_Res\_Empty}$$

$$\frac{}{\mathit{points\_to}{:}\mathit{points\_to} \rightsquigarrow \cdot; \cdot; \cdot, \mathit{points\_to}} \quad \textsc{Ty\_Pat\_Res\_PointsTo}$$

$$\frac{}{r{:}\mathit{res} \rightsquigarrow \cdot; \cdot; \cdot, r{:}\mathit{res}} \quad \textsc{Ty\_Pat\_Res\_Var}$$

$$\frac{\begin{array}{c}\mathit{res\_pattern}_1{:}\mathit{res}_1 \rightsquigarrow \mathcal{L}_1; \Phi_1; \mathcal{R}_1 \\ \mathit{res\_pattern}_2{:}\mathit{res}_2 \rightsquigarrow \mathcal{L}_2; \Phi_2; \mathcal{R}_2\end{array}}{\langle \mathit{res\_pattern}_1, \mathit{res\_pattern}_2 \rangle{:}\mathit{res}_1 * \mathit{res}_2 \rightsquigarrow \mathcal{L}_1, \mathcal{L}_2; \Phi_1, \Phi_2; \mathcal{R}_1, \mathcal{R}_2} \quad \textsc{Ty\_Pat\_Res\_SepConj}$$

$$\frac{\mathit{res\_pattern}{:}\mathit{res} \rightsquigarrow \mathcal{L}; \Phi; \mathcal{R}}{\mathit{res\_pattern}{:}\mathit{term} \wedge \mathit{res} \rightsquigarrow \mathcal{L}; \Phi, \mathit{term}; \mathcal{R}} \quad \textsc{Ty\_Pat\_Res\_Conj}$$

$$\frac{\mathit{res\_pattern}{:}x/y, \cdot(\mathit{res}) \rightsquigarrow \mathcal{L}; \Phi; \mathcal{R}}{\texttt{pack}\,(x, \mathit{res\_pattern}){:}\exists\, y{:}\beta.\ \mathit{res} \rightsquigarrow \mathcal{L}, x{:}\beta; \Phi; \mathcal{R}} \quad \textsc{Ty\_Pat\_Res\_Pack}$$

$\boxed{\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}}$

$$\frac{}{{:}\texttt{I} \rightsquigarrow \cdot; \cdot; \cdot; \cdot} \quad \textsc{Ty\_Pat\_Ret\_Empty}$$

$$\frac{\begin{array}{l}\mathit{ident\_or\_pattern}{:}\beta \rightsquigarrow \mathcal{C}_1\ \texttt{with}\ \mathit{term}_1 \\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{term}_1/y, \cdot(\mathit{ret}) \rightsquigarrow \mathcal{C}_2; \mathcal{L}_2; \Phi_2; \mathcal{R}_2\end{array}}{\texttt{comp}\,\mathit{ident\_or\_pattern}, \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\Sigma\, y{:}\beta.\ \mathit{ret} \rightsquigarrow \mathcal{C}_1, \mathcal{C}_2; \mathcal{L}_2; \Phi_1, \Phi_2; \mathcal{R}_2} \quad \textsc{Ty\_Pat\_Ret\_Comp}$$

$$\frac{\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}}{\texttt{log}\,y, \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\exists\, y{:}\beta.\ \mathit{ret} \rightsquigarrow \mathcal{C}; \mathcal{L}, y{:}\beta; \Phi; \mathcal{R}} \quad \textsc{Ty\_Pat\_Ret\_Log}$$

$$\frac{\begin{array}{l} \mathit{res\_pattern}{:}\mathit{res} \leadsto \mathcal{L}_1; \Phi_1; \mathcal{R}_1 \\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} \leadsto \mathcal{C}_2; \mathcal{L}_2; \Phi_2; \mathcal{R}_2 \end{array}}{\texttt{res}\,\mathit{res\_pattern}, \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{res} \otimes \mathit{ret} \leadsto \mathcal{C}_2; \mathcal{L}_1, \mathcal{L}_2; \Phi_1, \Phi_2; \mathcal{R}_1, \mathcal{R}_2} \quad \textsc{Ty\_Pat\_Ret\_Res}$$

$$\frac{\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} \leadsto \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}}{\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{term} \wedge \mathit{ret} \leadsto \mathcal{C}; \mathcal{L}; \Phi, \mathit{term}; \mathcal{R}} \quad \textsc{Ty\_Pat\_Ret\_Phi}$$

$\boxed{\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpval} \Leftarrow \mathit{ident}{:}\beta.\ \mathit{term}}$

$$\frac{\texttt{smt}\,(\Phi \Rightarrow \texttt{false})}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{undef}\ \ \mathit{UB\_name} \Leftarrow y{:}\beta.\ \mathit{term}} \quad \textsc{Ty\_TPVal\_Undef}$$

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta \\ \texttt{smt}\,(\Phi \Rightarrow \mathit{pval}/y, \cdot(\mathit{term})) \end{array}}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{done}\,\mathit{pval} \Leftarrow y{:}\beta.\ \mathit{term}} \quad \textsc{Ty\_TPVal\_Done}$$

$\boxed{\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpexpr} \Leftarrow \mathit{ident}{:}\beta.\ \mathit{term}}$

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \texttt{bool} \\ \mathcal{C}; \mathcal{L}; \Phi, \mathit{pval} = \texttt{true} \vdash \mathit{tpexpr}_1 \Leftarrow y{:}\beta.\ \mathit{term} \\ \mathcal{C}; \mathcal{L}; \Phi, \mathit{pval} = \texttt{false} \vdash \mathit{tpexpr}_2 \Leftarrow y{:}\beta.\ \mathit{term} \end{array}}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{if}\,\mathit{pval}\,\texttt{then}\,\mathit{tpexpr}_1\,\texttt{else}\,\mathit{tpexpr}_2 \Leftarrow y{:}\beta.\ \mathit{term}} \quad \textsc{Ty\_TPE\_If}$$

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pexpr} \Rightarrow y_1{:}\beta_1.\ \mathit{term}_1 \\ \mathit{ident\_or\_pattern}{:}\beta_1 \leadsto \mathcal{C}_1\,\texttt{with}\,\mathit{term} \\ \mathcal{C}, \mathcal{C}_1; \mathcal{L}; \Phi, \mathit{term}/y_1, \cdot(\mathit{term}_1) \vdash \mathit{tpexpr} \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2 \end{array}}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}\,\texttt{in}\,\mathit{tpexpr} \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2} \quad \textsc{Ty\_TPE\_Let}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi\vdash \mathit{tpexpr}_1 \Leftarrow y_1{:}\beta_1.\ \mathit{term}_1\\ \mathit{ident\_or\_pattern}{:}\beta_1 \rightsquigarrow \mathcal{C}_1\ \mathtt{with}\ \mathit{term}\\ \mathcal{C},\mathcal{C}_1;\mathcal{L};\Phi,\mathit{term}/y_1,\cdot(\mathit{term}_1)\vdash \mathit{tpexpr}\Leftarrow y_2{:}\beta_2.\ \mathit{term}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi\vdash \mathtt{let}\ \mathit{ident\_or\_pattern}{:}(y_1{:}\beta_1.\ \mathit{term}_1) = \mathit{tpexpr}_1\ \mathtt{in}\ \mathit{tpexpr}_2 \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2}\ \textsc{Ty\_TPE\_LetT}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval}\Rightarrow\beta_1\\ \overline{\mathit{pattern}_i{:}\beta_1\rightsquigarrow \mathcal{C}_i\ \mathtt{with}\ \mathit{term}_i}^{i}\\ \overline{\mathcal{C},\mathcal{C}_i;\mathcal{L};\Phi,\mathit{term}_i=\mathit{pval},\Phi_i\vdash \mathit{tpexpr}_i\Leftarrow y_2{:}\beta_2.\ \mathit{term}_2}^{i}\end{array}}{\mathcal{C};\mathcal{L};\Phi\vdash \mathtt{case}\ \mathit{pval}\ \mathtt{of}\ \overline{\mid \mathit{pattern}_i\Rightarrow \mathit{tpexpr}_i}^{i}\ \mathtt{end}\Leftarrow y_2{:}\beta_2.\ \mathit{term}_2}\ \textsc{Ty\_TPE\_Case}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{mem\_action}\Rightarrow \mathit{ret}}$

$$\frac{\mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval}\Rightarrow \mathtt{integer}}{\mathcal{C};\mathcal{L};\Phi;\cdot\vdash \mathtt{create}\,(\mathit{pval},\tau)\Rightarrow \Sigma\, y_p{:}\mathtt{loc}.\ \mathtt{representable}\,(\tau*,y_p)\wedge \mathtt{alignedI}\,(\mathit{pval},y_p)\wedge \exists\, y{:}\beta_\tau.\ y_p \overset{\times}{\mapsto}_\tau y\otimes \mathtt{I}}\ \textsc{Ty\_Action\_Create}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval}_0\Rightarrow \mathtt{loc}\\ \mathtt{smt}\,(\Phi\Rightarrow \mathit{pval}_0=\mathit{pval}_1)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{pval}_1\overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2\Leftarrow \mathit{pval}_1\overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathtt{load}\,(\tau,\mathit{pval}_0,\_,\mathit{pval}_1\overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2)\Rightarrow \Sigma\, y{:}\beta_\tau.\ y=\mathit{pval}_2\wedge \mathit{pval}_1\overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2\otimes\mathtt{I}}\ \textsc{Ty\_Action\_Load}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval}_0\Rightarrow \mathtt{loc}\\ \mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval}_1\Rightarrow \beta_\tau\\ \mathtt{smt}\,(\Phi\Rightarrow \mathtt{representable}\,(\tau,\mathit{pval}_1))\\ \mathtt{smt}\,(\Phi\Rightarrow \mathit{pval}_2=\mathit{pval}_0)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{pval}_2\mapsto_\tau \_\Leftarrow \mathit{pval}_2\mapsto_\tau \_\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathtt{store}\,(\_,\tau,\mathit{pval}_0,\mathit{pval}_1,\_,\mathit{pval}_2\mapsto_\tau \_)\Rightarrow \Sigma\,\_{:}\mathtt{unit}.\ \mathit{pval}_2\overset{\checkmark}{\mapsto}_\tau \mathit{pval}_1\otimes\mathtt{I}}\ \textsc{Ty\_Action\_Store}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval_0 \Rightarrow \texttt{loc}\\ \texttt{smt}\,(\Phi \Rightarrow pval_0 = pval_1)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash pval_1 \mapsto_\tau \_ \Leftarrow pval_1 \mapsto_\tau \_\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{kill}\,(\texttt{static}\,\tau, pval_0, pval_1 \mapsto_\tau \_) \Rightarrow \Sigma\,\_{:}\texttt{unit}.\,\texttt{I}} \quad \textsc{Ty\_Action\_Kill\_Static}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash mem\_op \Rightarrow ret}$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \texttt{loc}\\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{loc}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash pval_1\ binop_{rel}\ pval_2 \Rightarrow \Sigma\,y{:}\texttt{bool}.\ y = (pval_1\ binop_{rel}\ pval_2) \wedge \texttt{I}} \quad \textsc{Ty\_Memop\_Rel\_Binop}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \texttt{loc}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{intFromPtr}\,(\tau_1, \tau_2, pval) \Rightarrow \Sigma\,y{:}\texttt{integer}.\ y = \texttt{cast\_ptr\_to\_int}\,pval \wedge \texttt{I}} \quad \textsc{Ty\_Memop\_IntFromPtr}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \texttt{integer}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{ptrFromInt}\,(\tau_1, \tau_2, pval) \Rightarrow \Sigma\,y{:}\texttt{loc}.\ y = \texttt{cast\_int\_to\_ptr}\,pval \wedge \texttt{I}} \quad \textsc{Ty\_Memop\_PtrFromInt}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval_0 \Rightarrow \texttt{loc}\\ \texttt{smt}\,(\Phi \Rightarrow pval_1 = pval_0)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash pval_1 \overset{\checkmark}{\mapsto}_\tau \_ \Leftarrow pval_1 \overset{\checkmark}{\mapsto}_\tau \_\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{ptrValidForDeref}\,(\tau, pval_0, pval_1 \overset{\checkmark}{\mapsto}_\tau \_) \Rightarrow \Sigma\,y{:}\texttt{bool}.\ y = \texttt{aligned}\,(\tau, pval_1) \wedge pval_1 \overset{\checkmark}{\mapsto}_\tau \_ \otimes \texttt{I}} \quad \textsc{Ty\_Memop\_PtrValidForDeref}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \texttt{loc}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{ptrWellAligned}\,(\tau, pval) \Rightarrow \Sigma\,y{:}\texttt{bool}.\ y = \texttt{aligned}\,(\tau, pval) \wedge \texttt{I}} \quad \textsc{Ty\_Memop\_PtrWellAligned}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \texttt{loc}\\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{integer}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{ptrArrayShift}\,(pval_1, \tau, pval_2) \Rightarrow \Sigma\,y{:}\texttt{loc}.\ y = pval_1 +_{\text{ptr}} (pval_2 \times \text{size\_of}(\tau)) \wedge \texttt{I}} \quad \textsc{Ty\_Memop\_PtrArrayShift}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash tval \Leftarrow ret}$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{done}\ \ \Leftarrow \texttt{I}} \quad \textsc{Ty\_TVal\_I}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{done}\,\overline{spine\_elem_i}^{\,i} \Leftarrow pval/y,\cdot(ret)\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{done}\,pval,\ \overline{spine\_elem_i}^{\,i} \Leftarrow \Sigma\,y{:}\beta.\ ret} \quad \textsc{Ty\_TVal\_Comp}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{done}\,\overline{spine\_elem_i}^{\,i} \Leftarrow pval/y,\cdot(ret)\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{done}\,pval,\ \overline{spine\_elem_i}^{\,i} \Leftarrow \exists\,y{:}\beta.\ ret} \quad \textsc{Ty\_TVal\_Log}$$

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow term) \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{done}\,spine \Leftarrow ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{done}\,spine \Leftarrow term \wedge ret} \quad \textsc{Ty\_TVal\_Phi}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1 \vdash res\_term \Leftarrow res \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}_2 \vdash \texttt{done}\,\overline{spine\_elem_i}^{\,i} \Leftarrow ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1,\mathcal{R}_2 \vdash \texttt{done}\,res\_term,\ \overline{spine\_elem}^{\,i} \Leftarrow res \otimes ret} \quad \textsc{Ty\_TVal\_Res}$$

$$\frac{\texttt{smt}\,(\Phi \Rightarrow \texttt{false})}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{undef}\ \ UB\_name \Leftarrow ret} \quad \textsc{Ty\_TVal\_UB}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash seq\_expr \Rightarrow ret}$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \texttt{loc} \\ pval{:}arg\ \equiv\ \overline{x_i}^{\,i} \mapsto texpr\ \in \texttt{Globals} \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash\ \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{ccall}\,(\tau, pval, \overline{spine\_elem_i}^{\,i}) \Rightarrow \sigma(ret)} \quad \textsc{Ty\_Seq\_E\_CCall}$$

$$\frac{\begin{array}{l} name{:}arg \;\equiv\; \overline{x_i}^{\,i} \mapsto texpr \;\in\; \texttt{Globals} \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret \end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{pcall}\,(name, \overline{spine\_elem_i}^{\,i}) \Rightarrow \sigma(ret)} \quad \textsc{Ty\_Seq\_E\_Proc}$$

$$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash is\_expr \Rightarrow ret}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash mem\_op \Rightarrow ret}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{memop}\,(mem\_op) \Rightarrow ret} \quad \textsc{Ty\_Is\_E\_Memop}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash mem\_action \Rightarrow ret}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \; mem\_action \Rightarrow ret} \quad \textsc{Ty\_Is\_E\_Action}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash mem\_action \Rightarrow ret}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{neg}\, mem\_action \Rightarrow ret} \quad \textsc{Ty\_Is\_E\_Neg\_Action}$$

$$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash seq\_texpr \Leftarrow ret}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash tval \Leftarrow ret}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash tval \Leftarrow ret} \quad \textsc{Ty\_Seq\_TE\_TVal}$$

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L};\Phi \vdash pexpr \Rightarrow y{:}\beta.\ term \\ ident\_or\_pattern{:}\beta \rightsquigarrow \mathcal{C}_1\ \texttt{with}\, term_1 \\ \mathcal{C},\mathcal{C}_1;\mathcal{L};\Phi, term_1/y, \cdot(term);\mathcal{R} \vdash texpr \Leftarrow ret \end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{let}\, ident\_or\_pattern = pexpr\, \texttt{in}\, texpr \Leftarrow ret} \quad \textsc{Ty\_Seq\_TE\_LetP}$$

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L};\Phi \vdash tpexpr \Leftarrow y{:}\beta.\ term \\ ident\_or\_pattern{:}\beta \rightsquigarrow \mathcal{C}_1\ \texttt{with}\, term_1 \\ \mathcal{C},\mathcal{C}_1;\mathcal{L};\Phi, term_1/y, \cdot(term);\mathcal{R} \vdash texpr \Leftarrow ret \end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{let}\, ident\_or\_pattern{:}(y{:}\beta.\ term) = tpexpr\, \texttt{in}\, texpr \Leftarrow ret} \quad \textsc{Ty\_Seq\_TE\_LetPT}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}' \vdash \mathit{seq\_expr} \Rightarrow \mathit{ret}_1\\ \overline{\mathit{ret\_pattern}_i}^{\,i} : \mathit{ret}_1 \leadsto \mathcal{C}_1;\mathcal{L}_1;\Phi_1;\mathcal{R}_1\\ \mathcal{C},\mathcal{C}_1;\mathcal{L},\mathcal{L}_1;\Phi,\Phi_1;\mathcal{R},\mathcal{R}_1 \vdash \mathit{texpr} \Leftarrow \mathit{ret}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}',\mathcal{R} \vdash \mathtt{let}\, \overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{seq\_expr}\, \mathtt{in}\, \mathit{texpr} \Leftarrow \mathit{ret}_2} \quad \textsc{Ty\_Seq\_TE\_Let}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}' \vdash \mathit{texpr}_1 \Leftarrow \mathit{ret}_1\\ \overline{\mathit{ret\_pattern}_i}^{\,i} : \mathit{ret}_1 \leadsto \mathcal{C}_1;\mathcal{L}_1;\Phi_1;\mathcal{R}_1\\ \mathcal{C},\mathcal{C}_1;\mathcal{L},\mathcal{L}_1;\Phi,\Phi_1;\mathcal{R},\mathcal{R}_1 \vdash \mathit{texpr}_2 \Leftarrow \mathit{ret}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}',\mathcal{R} \vdash \mathtt{let}\, \overline{\mathit{ret\_pattern}_i}^{\,i} : \mathit{ret}_1 = \mathit{texpr}_1\, \mathtt{in}\, \mathit{texpr}_2 \Leftarrow \mathit{ret}_2} \quad \textsc{Ty\_Seq\_TE\_LetT}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta_1\\ \overline{\mathit{pattern}_i : \beta_1 \leadsto \mathcal{C}_i\, \mathtt{with}\, \mathit{term}_i}^{\,i}\\ \overline{\mathcal{C},\mathcal{C}_i;\mathcal{L};\Phi,\mathit{term}_i = \mathit{pval},\Phi_i;\mathcal{R} \vdash \mathit{texpr}_i \Leftarrow \mathit{ret}}^{\,i}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{case}\, \mathit{pval}\, \mathtt{of}\, \overline{\mid \mathit{pattern}_i \Rightarrow \mathit{texpr}_i}^{\,i}\, \mathtt{end} \Leftarrow \mathit{ret}} \quad \textsc{Ty\_Seq\_TE\_Case}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \mathtt{bool}\\ \mathcal{C};\mathcal{L};\Phi,\mathit{pval} = \mathtt{true};\mathcal{R} \vdash \mathit{texpr}_1 \Leftarrow \mathit{ret}\\ \mathcal{C};\mathcal{L};\Phi,\mathit{pval} = \mathtt{false};\mathcal{R} \vdash \mathit{texpr}_2 \Leftarrow \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{if}\, \mathit{pval}\, \mathtt{then}\, \mathit{texpr}_1\, \mathtt{else}\, \mathit{texpr}_2 \Leftarrow \mathit{ret}} \quad \textsc{Ty\_Seq\_TE\_If}$$

$$\frac{\begin{array}{l}\mathit{ident}:\mathit{arg} \equiv \overline{x_i}^{\,i} \mapsto \mathit{texpr} \in \mathtt{Globals}\\ \mathcal{C};\mathcal{L};\Phi;\cdot \vdash \overline{x_i = \mathit{pval}_i}^{\,i} :: \mathit{arg} \gg \sigma;\mathtt{false} \wedge \mathtt{I}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \mathtt{run}\, \mathit{ident}\, \overline{\mathit{pval}_i}^{\,i} \Leftarrow \mathtt{false} \wedge \mathtt{I}} \quad \textsc{Ty\_Seq\_TE\_Run}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{bound}\,[\mathit{int}](\mathit{is\_texpr}) \Leftarrow \mathit{ret}} \quad \textsc{Ty\_Seq\_TE\_Bound}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}}$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{is\_expr} \Rightarrow \mathit{ret}_1\\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret}_1 \leadsto \mathcal{C}_1;\mathcal{L}_1;\Phi_1;\mathcal{R}_1\\ \mathcal{C},\mathcal{C}_1;\mathcal{L},\mathcal{L}_1;\Phi,\Phi_1;\mathcal{R},\mathcal{R}_1\vdash \mathit{texpr} \Leftarrow \mathit{ret}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \texttt{let}\,\texttt{strong}\,\overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{is\_expr}\,\texttt{in}\,\mathit{texpr} \Leftarrow \mathit{ret}_2}\quad \textsc{Ty\_Is\_TE\_LetS}$$

$\boxed{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{texpr} \Leftarrow \mathit{ret}}$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}}\quad \textsc{Ty\_TE\_Is}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret}}\quad \textsc{Ty\_TE\_Seq}$$

$\boxed{\mathit{pattern} = \mathit{pval} \leadsto \sigma}$

$$\frac{}{\_{:}\_ = \mathit{pval} \leadsto \cdot}\quad \textsc{Subs\_Decons\_Value\_No\_Sym\_Annot}$$

$$\frac{}{x{:}\_ = \mathit{pval} \leadsto \mathit{pval}/x,\cdot}\quad \textsc{Subs\_Decons\_Value\_Sym\_Annot}$$

$$\frac{\begin{array}{c}\mathit{pattern}_1 = \mathit{pval}_1 \leadsto \sigma_1\\ \mathit{pattern}_2 = \mathit{pval}_2 \leadsto \sigma_2\end{array}}{\texttt{Cons}(\mathit{pattern}_1,\mathit{pattern}_2) = \texttt{Cons}(\mathit{pval}_1,\mathit{pval}_2) \leadsto \sigma_1,\sigma_2}\quad \textsc{Subs\_Decons\_Value\_Cons}$$

$$\frac{\overline{\mathit{pattern}_i = \mathit{pval}_i \leadsto \sigma_i}^{\,i}}{\texttt{Tuple}(\,\overline{\mathit{pattern}_i}^{\,i}\,) = \texttt{Tuple}(\,\overline{\mathit{pval}_i}^{\,i}\,) \leadsto \overline{\sigma_i}^{\,i}}\quad \textsc{Subs\_Decons\_Value\_Tuple}$$

$$\frac{\overline{pattern_i = pval_i \leadsto \sigma_i}^{\,i}}{\texttt{Array}(\,\overline{pattern_i}^{\,i}\,) = \texttt{Array}(\,\overline{pval_i}^{\,i}\,) \leadsto \overline{\sigma_i}^{\,i}} \quad \textsc{Subs\_Decons\_Value\_Array}$$

$$\frac{pattern = pval \leadsto \sigma}{\texttt{Specified}(pattern) = pval \leadsto \sigma} \quad \textsc{Subs\_Decons\_Value\_Specified}$$

$\boxed{ident\_or\_pattern = pval \leadsto \sigma}$

$$\frac{}{x = pval \leadsto pval/x, \cdot} \quad \textsc{Subs\_Decons\_Value'\_Sym}$$

$$\frac{pattern = pval \leadsto \sigma}{pattern = pval \leadsto \sigma} \quad \textsc{Subs\_Decons\_Value'\_Pattern}$$

$\boxed{res\_pattern = res\_term \leadsto \sigma}$

$$\frac{}{\texttt{emp} = \texttt{emp} \leadsto \cdot} \quad \textsc{Subs\_Decons\_Res\_Emp}$$

$$\frac{}{pt = pt \leadsto \cdot} \quad \textsc{Subs\_Decons\_Res\_Points\_to}$$

$$\frac{}{ident = res\_term \leadsto res\_term/ident, \cdot} \quad \textsc{Subs\_Decons\_Res\_Var}$$

$$\frac{\begin{array}{c} res\_pattern_1 = res\_term_1 \leadsto \sigma_1 \\ res\_pattern_2 = res\_term_2 \leadsto \sigma_2 \end{array}}{\langle res\_pattern_1, res\_pattern_2 \rangle = \langle res\_term_1, res\_term_2 \rangle \leadsto \sigma_1, \sigma_2} \quad \textsc{Subs\_Decons\_Res\_Pair}$$

$$\frac{res\_pattern = res\_term \leadsto \sigma}{\texttt{pack}\,(ident, res\_pattern) = \texttt{pack}\,(pval, res\_term) \leadsto pval/ident, \sigma} \quad \textsc{Subs\_Decons\_Res\_Pack}$$

$$\boxed{\overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \sigma}$$

$$\frac{}{\rightsquigarrow \cdot}\quad \textsc{Subs\_Decons\_Ret\_Empty}$$

$$\frac{\begin{array}{c} ident\_or\_pattern = pval \rightsquigarrow \sigma \\ \overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \psi \end{array}}{\texttt{comp}\, ident\_or\_pattern = pval,\ \overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \sigma, \psi}\quad \textsc{Subs\_Decons\_Ret\_Comp}$$

$$\frac{\overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \psi}{\texttt{log}\, ident = pval,\ \overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow pval/ident, \psi}\quad \textsc{Subs\_Decons\_Ret\_Log}$$

$$\frac{\begin{array}{c} res\_pattern = res\_term \rightsquigarrow \sigma \\ \overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \psi \end{array}}{\texttt{res}\, res\_pattern = res\_term,\ \overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \sigma, \psi}\quad \textsc{Subs\_Decons\_Ret\_Res}$$

$$\boxed{\overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret}$$

$$\frac{}{::ret \gg \cdot; ret}\quad \textsc{Subs\_Decons\_Arg\_Empty}$$

$$\frac{\overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret}{x = pval,\ \overline{x_i = spine\_elem_i}^{\,i} :: \Pi\, x{:}\beta.\ arg \gg pval/x, \sigma; ret}\quad \textsc{Subs\_Decons\_Arg\_Comp}$$

$$\frac{\overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret}{x = pval,\ \overline{x_i = spine\_elem_i}^{\,i} :: \forall\, x{:}\beta.\ arg \gg pval/x, \sigma; ret}\quad \textsc{Subs\_Decons\_Arg\_Log}$$

$$\frac{\overline{x_i = spine\_elem_i}^{\,i} \ :: arg \gg \sigma; ret}{x = res\_term,\ \overline{x_i = spine\_elem_i}^{\,i} \ :: res \multimap arg \gg res\_term/x, \sigma; ret} \quad \textsc{Subs\_Decons\_Arg\_Res}$$

$$\frac{\overline{x_i = spine\_elem_i}^{\,i} \ :: arg \gg \sigma; ret}{\overline{x_i = spine\_elem_i}^{\,i} \ :: term \supset arg \gg \sigma; ret} \quad \textsc{Subs\_Decons\_Arg\_Phi}$$

$\boxed{\langle pexpr \rangle \longrightarrow \langle pexpr' \rangle}$

$$\frac{mem\_ptr' \ \equiv \ mem\_ptr +_{\text{ptr}} mem\_int \times \text{size\_of}(\tau)}{\langle \texttt{array\_shift}\,(mem\_ptr, \tau, mem\_int) \rangle \longrightarrow \langle mem\_ptr' \rangle} \quad \textsc{Op\_PE\_PE\_ArrayShift}$$

$$\frac{mem\_ptr' \ \equiv \ mem\_ptr +_{\text{ptr}} \text{offset\_of}_{tag}(member)}{\langle \texttt{member\_shift}\,(mem\_ptr, tag, member) \rangle \longrightarrow \langle mem\_ptr' \rangle} \quad \textsc{Op\_PE\_PE\_MemberShift}$$

$$\frac{}{\langle \texttt{not}\,(\texttt{True}) \rangle \longrightarrow \langle \texttt{False} \rangle} \quad \textsc{Op\_PE\_PE\_Not\_True}$$

$$\frac{}{\langle \texttt{not}\,(\texttt{False}) \rangle \longrightarrow \langle \texttt{True} \rangle} \quad \textsc{Op\_PE\_PE\_Not\_False}$$

$$\frac{mem\_int \ \equiv \ mem\_int_1 \ binop_{arith} \ mem\_int_2}{\langle mem\_int_1 \ \ binop_{arith} \ \ mem\_int_2 \rangle \longrightarrow \langle mem\_int \rangle} \quad \textsc{Op\_PE\_PE\_Arith\_Binop}$$

$$\frac{bool\_value \ \equiv \ mem\_int_1 \ binop_{rel} \ mem\_int_2}{\langle mem\_int_1 \ \ binop_{rel} \ \ mem\_int_2 \rangle \longrightarrow \langle bool\_value \rangle} \quad \textsc{Op\_PE\_PE\_Rel\_Binop}$$

$$\frac{bool\_value \ \equiv \ bool\_value_1 \ binop_{bool} \ bool\_value_2}{\langle bool\_value_1 \ \ binop_{bool} \ \ bool\_value_2 \rangle \longrightarrow \langle bool\_value \rangle} \quad \textsc{Op\_PE\_PE\_Bool\_Binop}$$

$$\frac{}{\langle \texttt{assert\_undef}\,(\texttt{True},\ \mathit{UB\_name})\rangle \longrightarrow \langle \texttt{Unit}\rangle}\ \textsc{Op\_PE\_PE\_Assert\_Undef}$$

$$\frac{}{\langle \texttt{bool\_to\_integer}\,(\texttt{True})\rangle \longrightarrow \langle 1\rangle}\ \textsc{Op\_PE\_PE\_Bool\_To\_Integer\_True}$$

$$\frac{}{\langle \texttt{bool\_to\_integer}\,(\texttt{False})\rangle \longrightarrow \langle 0\rangle}\ \textsc{Op\_PE\_PE\_Bool\_To\_Integer\_False}$$

$$\frac{\begin{array}{l} \mathit{abbrev}_1 \equiv \mathrm{max\_int}_\tau - \mathrm{min\_int}_\tau + 1 \\ \mathit{abbrev}_2 \equiv \mathit{pval}\,\texttt{rem\_f}\,\mathit{abbrev}_1 \\ \mathit{mem\_int}' \equiv \texttt{if}\ \mathit{abbrev}_2 \leq \mathrm{max\_int}_\tau\ \texttt{then}\ \mathit{abbrev}_2\ \texttt{else}\ \mathit{abbrev}_2 - \mathit{abbrev}_1 \end{array}}{\langle \texttt{wrapI}\,(\tau, \mathit{mem\_int})\rangle \longrightarrow \langle \mathit{mem\_int}'\rangle}\ \textsc{Op\_PE\_PE\_WrapI}$$

$\boxed{\langle \mathit{pexpr}\rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\,\mathit{term})\rangle}$

$$\frac{\begin{array}{c} \mathit{name}{:}\mathit{pure\_arg} \equiv \overline{x_i}^{\,i} \mapsto \mathit{tpexpr} \in \texttt{Globals} \\ \overline{x_i = \mathit{pval}_i}^{\,i} :: \mathit{pure\_arg} \gg \sigma; \Sigma\, y{:}\beta.\,\mathit{term} \wedge \texttt{I} \end{array}}{\langle \mathit{name}(\overline{\mathit{pval}_i}^{\,i})\rangle \longrightarrow \langle \sigma(\mathit{tpexpr}){:}(y{:}\beta.\,\sigma(\mathit{term}))\rangle}\ \textsc{Op\_PE\_TPE\_Call}$$

$\boxed{\langle \mathit{tpexpr}\rangle \longrightarrow \langle \mathit{tpexpr}'\rangle}$

$$\frac{\begin{array}{c} \mathit{pattern}_j = \mathit{pval} \rightsquigarrow \sigma_j \\ \forall\, i < j.\ \texttt{not}\,(\mathit{pattern}_i = \mathit{pval} \rightsquigarrow \sigma_i) \end{array}}{\langle \texttt{case}\,\mathit{pval}\,\texttt{of}\,\overline{\mid \mathit{pattern}_i \Rightarrow \mathit{tpexpr}_i}^{\,i}\ \texttt{end}\rangle \longrightarrow \langle \sigma_j(\mathit{tpexpr}_j)\rangle}\ \textsc{Op\_TPE\_TPE\_Case}$$

$$\frac{\mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma}{\langle \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pval}\,\texttt{in}\,\mathit{tpexpr}\rangle \longrightarrow \langle \sigma(\mathit{tpexpr})\rangle}\ \textsc{Op\_TPE\_TPE\_Let\_Sub}$$

$$\frac{\langle \mathit{pexpr}\rangle \longrightarrow \langle \mathit{pexpr}'\rangle}{\langle \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}\,\texttt{in}\,\mathit{tpexpr}\rangle \longrightarrow \langle \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}'\,\texttt{in}\,\mathit{tpexpr}\rangle}\ \textsc{Op\_TPE\_TPE\_Let\_Let}$$

$$\frac{\langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{tpexpr}_1{:}(y{:}\beta.\ \mathit{term}) \rangle}{\langle \texttt{let}\, \mathit{ident\_or\_pattern} = \mathit{pexpr}\ \texttt{in}\, \mathit{tpexpr}_2 \rangle \longrightarrow \langle \texttt{let}\, \mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \mathit{tpexpr}_1\ \texttt{in}\, \mathit{tpexpr}_2 \rangle} \quad \textsc{Op\_TPE\_TPE\_Let\_LetT}$$

$$\frac{\mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma}{\langle \texttt{let}\, \mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \texttt{done}\, \mathit{pval}\ \texttt{in}\, \mathit{tpexpr} \rangle \longrightarrow \langle \sigma(\mathit{tpexpr}) \rangle} \quad \textsc{Op\_TPE\_TPE\_LetT\_Sub}$$

$$\frac{\langle \mathit{tpexpr}_1 \rangle \longrightarrow \langle \mathit{tpexpr}'_1 \rangle}{\langle \texttt{let}\, \mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \mathit{tpexpr}_1\ \texttt{in}\, \mathit{tpexpr}_2 \rangle \longrightarrow \langle \texttt{let}\, \mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \mathit{tpexpr}'_1\ \texttt{in}\, \mathit{tpexpr}_2 \rangle} \quad \textsc{Op\_TPE\_TPE\_LetT\_LetT}$$

$$\frac{}{\langle \texttt{if True then}\, \mathit{tpexpr}_1\ \texttt{else}\, \mathit{tpexpr}_2 \rangle \longrightarrow \langle \mathit{tpexpr}_1 \rangle} \quad \textsc{Op\_TPE\_TPE\_If\_True}$$

$$\frac{}{\langle \texttt{if False then}\, \mathit{tpexpr}_1\ \texttt{else}\, \mathit{tpexpr}_2 \rangle \longrightarrow \langle \mathit{tpexpr}_2 \rangle} \quad \textsc{Op\_TPE\_TPE\_If\_False}$$

$$\boxed{\langle h; \mathit{seq\_expr} \rangle \longrightarrow \langle h'; \mathit{texpr}{:}\mathit{ret} \rangle}$$

$$\frac{\begin{array}{l} \mathit{mem\_ptr}{:}\mathit{arg} \ \equiv\ \overline{x_i}^{\,i} \ \mapsto \mathit{texpr} \ \in\ \texttt{Globals} \\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} \ :: \mathit{arg} \gg \sigma; \mathit{ret} \end{array}}{\langle h; \texttt{ccall}\,(\tau, \mathit{mem\_ptr}, \overline{\mathit{spine\_elem}_i}^{\,i}) \rangle \longrightarrow \langle h; \sigma(\mathit{texpr}){:}\sigma(\mathit{ret}) \rangle} \quad \textsc{Op\_SE\_TE\_CCall}$$

$$\frac{\begin{array}{l} \mathit{name}{:}\mathit{arg} \ \equiv\ \overline{x_i}^{\,i} \ \mapsto \mathit{texpr} \ \in\ \texttt{Globals} \\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} \ :: \mathit{arg} \gg \sigma; \mathit{ret} \end{array}}{\langle h; \texttt{pcall}\,(\mathit{name}, \overline{\mathit{spine\_elem}_i}^{\,i}) \rangle \longrightarrow \langle h; \sigma(\mathit{texpr}){:}\sigma(\mathit{ret}) \rangle} \quad \textsc{Op\_SE\_TE\_PCall}$$

$$\boxed{\langle h; \mathit{seq\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle}$$

$$\frac{\begin{array}{l} \mathit{ident}{:}\mathit{arg} \ \equiv\ \overline{x_i}^{\,i} \ \mapsto \mathit{texpr} \ \in\ \texttt{Globals} \\ \overline{x_i = \mathit{pval}_i}^{\,i} \ :: \mathit{arg} \gg \sigma; \texttt{false} \wedge \texttt{I} \end{array}}{\langle h; \texttt{run}\, \mathit{ident}\, \overline{\mathit{pval}_i}^{\,i} \rangle \longrightarrow \langle h; \sigma(\mathit{texpr}) \rangle} \quad \textsc{Op\_STE\_TE\_Run}$$

$$\frac{\begin{array}{c}\mathit{pattern}_j = \mathit{pval} \rightsquigarrow \sigma_j \\ \forall\, i < j.\ \texttt{not}\,(\mathit{pattern}_i = \mathit{pval} \rightsquigarrow \sigma_i)\end{array}}{\langle h; \texttt{case}\,\mathit{pval}\,\texttt{of}\,\overline{\mid \mathit{pattern}_i \Rightarrow \mathit{texpr}_i}^{\,i}\ \texttt{end}\rangle \longrightarrow \langle h; \sigma_j(\mathit{texpr}_j)\rangle}\ \textsc{Op\_STE\_TE\_Case}$$

$$\frac{\mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma}{\langle h; \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pval}\,\texttt{in}\,\mathit{texpr}\rangle \longrightarrow \langle h; \sigma(\mathit{texpr})\rangle}\ \textsc{Op\_STE\_TE\_LetP\_Sub}$$

$$\frac{\langle \mathit{pexpr}\rangle \longrightarrow \langle \mathit{pexpr}'\rangle}{\langle h; \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}\,\texttt{in}\,\mathit{texpr}\rangle \longrightarrow \langle h; \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}'\,\texttt{in}\,\mathit{texpr}\rangle}\ \textsc{Op\_STE\_TE\_LetP\_LetP}$$

$$\frac{\langle \mathit{pexpr}\rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\ \mathit{term})\rangle}{\langle h; \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}\,\texttt{in}\,\mathit{texpr}\rangle \longrightarrow \langle h; \texttt{let}\,\mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \mathit{tpexpr}\,\texttt{in}\,\mathit{texpr}\rangle}\ \textsc{Op\_STE\_TE\_LetP\_LetTP}$$

$$\frac{\mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma}{\langle h; \texttt{let}\,\mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \texttt{done}\,\mathit{pval}\,\texttt{in}\,\mathit{texpr}\rangle \longrightarrow \langle h; \sigma(\mathit{texpr})\rangle}\ \textsc{Op\_STE\_TE\_LetTP\_Sub}$$

$$\frac{\langle \mathit{tpexpr}\rangle \longrightarrow \langle \mathit{tpexpr}'\rangle}{\langle h; \texttt{let}\,\mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \mathit{tpexpr}\,\texttt{in}\,\mathit{texpr}\rangle \longrightarrow \langle h; \texttt{let}\,\mathit{ident\_or\_pattern}{:}(y{:}\beta.\ \mathit{term}) = \mathit{tpexpr}'\,\texttt{in}\,\mathit{texpr}\rangle}\ \textsc{Op\_STE\_TE\_LetTP\_LetTP}$$

$$\frac{\overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \rightsquigarrow \sigma}{\langle h; \texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} = \texttt{done}\,\overline{\mathit{spine\_elem}_i}^{\,i}\,\texttt{in}\,\mathit{texpr}\rangle \longrightarrow \langle h; \sigma(\mathit{texpr})\rangle}\ \textsc{Op\_STE\_TE\_LetT\_Sub}$$

$$\frac{\langle h; \mathit{seq\_expr}\rangle \longrightarrow \langle h; \mathit{texpr}_1{:}\mathit{ret}\rangle}{\langle h; \texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{seq\_expr}\,\texttt{in}\,\mathit{texpr}_2\rangle \longrightarrow \langle h; \texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} = \mathit{texpr}_1\,\texttt{in}\,\mathit{texpr}_2\rangle}\ \textsc{Op\_STE\_TE\_Let\_LetT}$$

$$\frac{\langle h; \mathit{texpr}_1\rangle \longrightarrow \langle h'; \mathit{texpr}_1'\rangle}{\langle h; \texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} = \mathit{texpr}_1\,\texttt{in}\,\mathit{texpr}_2\rangle \longrightarrow \langle h'; \texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} = \mathit{texpr}_1'\,\texttt{in}\,\mathit{texpr}_2\rangle}\ \textsc{Op\_STE\_TE\_LetT\_LetT}$$

$$\frac{}{\langle h; \texttt{if True then}\, texpr_1 \,\texttt{else}\, texpr_2\rangle \longrightarrow \langle h; texpr_1\rangle} \quad \textsc{Op\_STE\_TE\_If\_True}$$

$$\frac{}{\langle h; \texttt{if False then}\, texpr_1 \,\texttt{else}\, texpr_2\rangle \longrightarrow \langle h; texpr_2\rangle} \quad \textsc{Op\_STE\_TE\_If\_False}$$

$$\frac{}{\langle h; \texttt{bound}\,[int](is\_texpr)\rangle \longrightarrow \langle h; is\_texpr\rangle} \quad \textsc{Op\_STE\_TE\_Bound}$$

$\boxed{\langle h; mem\_op\rangle \longrightarrow \langle h'; tval\rangle}$

$$\frac{bool\_value \;\equiv\; mem\_int_1 \; binop_{rel} \; mem\_int_2}{\langle h; mem\_int_1 \;\; binop_{rel} \;\; mem\_int_2\rangle \longrightarrow \langle h; \texttt{done}\, bool\_value\rangle} \quad \textsc{Op\_Memop\_TVal\_Rel\_Binop}$$

$$\frac{mem\_int \;\equiv\; \texttt{cast\_ptr\_to\_int}\, mem\_ptr}{\langle h; \texttt{intFromPtr}\,(\tau_1, \tau_2, mem\_ptr)\rangle \longrightarrow \langle h; \texttt{done}\, mem\_int\rangle} \quad \textsc{Op\_Memop\_TVal\_IntFromPtr}$$

$$\frac{mem\_ptr \;\equiv\; \texttt{cast\_ptr\_to\_int}\, mem\_int}{\langle h; \texttt{ptrFromInt}\,(\tau_1, \tau_2, mem\_int)\rangle \longrightarrow \langle h; \texttt{done}\, mem\_ptr\rangle} \quad \textsc{Op\_Memop\_TVal\_PtrFromInt}$$

$$\frac{bool\_value \;\equiv\; \texttt{aligned}\,(\tau, mem\_ptr)}{\langle h + \{mem\_ptr \overset{\checkmark}{\mapsto}_\tau \_\}; \texttt{ptrValidForDeref}\,(\tau, mem\_ptr, mem\_ptr \overset{\checkmark}{\mapsto}_\tau \_)\rangle \longrightarrow \langle h + \{mem\_ptr \overset{\checkmark}{\mapsto}_\tau \_\}; \texttt{done}\, bool\_value, mem\_ptr \overset{\checkmark}{\mapsto}_\tau \_\rangle} \quad \textsc{Op\_Memop\_TVal\_PtrValid}$$

$$\frac{bool\_value \;\equiv\; \texttt{aligned}\,(\tau, mem\_ptr)}{\langle h; \texttt{ptrWellAligned}\,(\tau, mem\_ptr)\rangle \longrightarrow \langle h; \texttt{done}\, bool\_value\rangle} \quad \textsc{Op\_Memop\_TVal\_PtrWellAligned}$$

$$\frac{mem\_ptr' \;\equiv\; mem\_ptr +_{\mathrm{ptr}} (mem\_int \times \mathrm{size\_of}(\tau))}{\langle h; \texttt{ptrArrayShift}\,(mem\_ptr, \tau, mem\_int)\rangle \longrightarrow \langle h; \texttt{done}\, mem\_ptr'\rangle} \quad \textsc{Op\_Memop\_TVal\_PtrArrayShift}$$

$\boxed{\langle h; mem\_action\rangle \longrightarrow \langle h'; tval\rangle}$

$$\frac{\begin{array}{l}\texttt{fresh}\,(mem\_ptr)\\ \texttt{representable}\,(\tau*, mem\_ptr)\\ \texttt{alignedI}\,(mem\_int, mem\_ptr)\\ pval{:}\beta_\tau\end{array}}{\langle h; \texttt{create}\,(mem\_int, \tau)\rangle \longrightarrow \langle h + \{mem\_ptr \overset{\times}{\mapsto}_\tau pval\}; \texttt{done}\, mem\_ptr, pval, mem\_ptr \overset{\times}{\mapsto}_\tau pval\rangle} \quad \text{OP\_ACTION\_TVAL\_CREATE}$$

$$\frac{}{\langle h + \{mem\_ptr \overset{\checkmark}{\mapsto}_\tau pval\}; \texttt{load}\,(\tau, mem\_ptr, \_, mem\_ptr \overset{\checkmark}{\mapsto}_\tau pval)\rangle \longrightarrow \langle h + \{mem\_ptr \overset{\checkmark}{\mapsto}_\tau pval\}; \texttt{done}\, pval, mem\_ptr \overset{\checkmark}{\mapsto}_\tau pval\rangle} \quad \text{OP\_ACTION\_TVAL\_LOAD}$$

$$\frac{}{\langle h + \{mem\_ptr \overset{\checkmark}{\mapsto}_\tau \_\}; \texttt{store}\,(\_, \tau, mem\_ptr, pval, \_, mem\_ptr \overset{\checkmark}{\mapsto}_\tau \_)\rangle \longrightarrow \langle h + \{mem\_ptr \overset{\checkmark}{\mapsto}_\tau pval\}; \texttt{done}\,\texttt{Unit}, mem\_ptr \overset{\checkmark}{\mapsto}_\tau pval\rangle} \quad \text{OP\_ACTION\_TVAL\_STORE}$$

$$\frac{}{\langle h + \{mem\_ptr \mapsto_\tau \_\}; \texttt{kill}\,(\texttt{static}\,\tau, mem\_ptr, mem\_ptr \mapsto_\tau \_)\rangle \longrightarrow \langle h; \texttt{done}\,\texttt{Unit}\rangle} \quad \text{OP\_ACTION\_TVAL\_KILL\_STATIC}$$

$\boxed{\langle h; is\_expr\rangle \longrightarrow \langle h'; is\_expr'\rangle}$

$$\frac{\langle h; mem\_op\rangle \longrightarrow \langle h; tval\rangle}{\langle h; \texttt{memop}\,(mem\_op)\rangle \longrightarrow \langle h; tval\rangle} \quad \text{OP\_ISE\_ISE\_MEMOP}$$

$$\frac{\langle h; mem\_action\rangle \longrightarrow \langle h'; tval\rangle}{\langle h;\ mem\_action\rangle \longrightarrow \langle h'; tval\rangle} \quad \text{OP\_ISE\_ISE\_ACTION}$$

$$\frac{\langle h; mem\_action\rangle \longrightarrow \langle h'; tval\rangle}{\langle h; \texttt{neg}\, mem\_action\rangle \longrightarrow \langle h'; tval\rangle} \quad \text{OP\_ISE\_ISE\_NEG\_ACTION}$$

$\boxed{\langle h; is\_texpr\rangle \longrightarrow \langle h'; texpr\rangle}$

$$\frac{\overline{ret\_pattern_i = spine\_elem_i}^{\,i} \leadsto \sigma}{\langle h; \texttt{let strong}\, \overline{ret\_pattern_i}^{\,i} = \texttt{done}\, \overline{spine\_elem_i}^{\,i}\, \texttt{in}\, texpr\rangle \longrightarrow \langle h; \sigma(texpr)\rangle} \quad \textsc{Op\_IsTE\_IsTE\_LetS\_Sub}$$

$$\frac{\langle h; is\_expr\rangle \longrightarrow \langle h'; is\_expr'\rangle}{\langle h; \texttt{let strong}\, \overline{ret\_pattern_i}^{\,i} = is\_expr\, \texttt{in}\, texpr\rangle \longrightarrow \langle h'; \texttt{let strong}\, \overline{ret\_pattern_i}^{\,i} = is\_expr'\, \texttt{in}\, texpr\rangle} \quad \textsc{Op\_IsTE\_IsTE\_LetS\_LetS}$$

$\boxed{\langle h; texpr\rangle \longrightarrow \langle h'; texpr'\rangle}$

$$\frac{\langle h; seq\_texpr\rangle \longrightarrow \langle h; texpr\rangle}{\langle h; seq\_texpr\rangle \longrightarrow \langle h; texpr\rangle} \quad \textsc{Op\_TE\_TE\_Seq}$$

$$\frac{\langle h; is\_texpr\rangle \longrightarrow \langle h'; texpr\rangle}{\langle h; is\_texpr\rangle \longrightarrow \langle h'; texpr\rangle} \quad \textsc{Op\_TE\_TE\_Is}$$

$\boxed{\overline{x_i}^{\,i} :: arg \leadsto \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}$

$$\frac{}{::ret \leadsto \cdot; \cdot; \cdot; \cdot \mid ret} \quad \textsc{Arg\_Env\_Ret}$$

$$\frac{\overline{x_i}^{\,i} :: arg \leadsto \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{x,\, \overline{x_i}^{\,i} :: \Pi\, x{:}\beta.\ arg \leadsto \mathcal{C}, x{:}\beta; \mathcal{L}; \Phi; \mathcal{R} \mid ret} \quad \textsc{Arg\_Env\_Comp}$$

$$\frac{\overline{x_i}^{\,i} :: arg \leadsto \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{x,\, \overline{x_i}^{\,i} :: \forall\, x{:}\beta.\ arg \leadsto \mathcal{C}; \mathcal{L}, x{:}\beta; \Phi; \mathcal{R} \mid ret} \quad \textsc{Arg\_Env\_Log}$$

$$\frac{\overline{x_i}^{\,i} :: arg \leadsto \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{\overline{x_i}^{\,i} :: term \supset arg \leadsto \mathcal{C}; \mathcal{L}; \Phi, term; \mathcal{R} \mid ret} \quad \textsc{Arg\_Env\_Phi}$$

$$\frac{\overline{x_i}^{\,i} :: arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{x,\ \overline{x_i}^{\,i} :: res \multimap arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}, x{:}res \mid ret} \quad \textsc{Arg\_Env\_Res}$$

$\boxed{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}$

$$\frac{}{\cdot; \cdot; \cdot; \cdot \sqsubseteq \cdot; \cdot; \cdot; \cdot} \quad \textsc{Weak\_Empty}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}, x{:}\beta; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}', x{:}\beta; \mathcal{L}'; \Phi'; \mathcal{R}'} \quad \textsc{Weak\_Cons\_Comp}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}, x{:}\beta; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}', x{:}\beta; \Phi'; \mathcal{R}'} \quad \textsc{Weak\_Cons\_Log}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi, term; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi', term; \mathcal{R}'} \quad \textsc{Weak\_Cons\_Phi}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}, res \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}', res} \quad \textsc{Weak\_Cons\_Res\_Anon}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}, x{:}res \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}', x{:}res} \quad \textsc{Weak\_Cons\_Res\_Named}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}', x{:}\beta; \mathcal{L}'; \Phi'; \mathcal{R}'} \quad \textsc{Weak\_Skip\_Comp}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}', x{:}\beta; \Phi'; \mathcal{R}'} \quad \textsc{Weak\_Skip\_Log}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi', term; \mathcal{R}'} \quad \textsc{Weak\_Skip\_Phi}$$

$\boxed{\mathcal{C}; \mathcal{L}; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \mathcal{R}')}$

$$\frac{}{\mathcal{C}; \mathcal{L}; \cdot \vdash (\cdot){:}(\cdot; \cdot; \cdot)} \quad \textsc{Ty\_Subs\_Empty}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \mathcal{R}') \\ \mathcal{C}; \mathcal{L}; \Phi \vdash pval \Rightarrow \beta\end{array}}{\mathcal{C}; \mathcal{L}; \mathcal{R} \vdash (pval/x, \sigma){:}(\mathcal{C}', x{:}\beta; \mathcal{L}'; \mathcal{R}')} \quad \textsc{Ty\_Subs\_Cons\_Comp}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \mathcal{R}') \\ \mathcal{C}; \mathcal{L}; \Phi \vdash pval \Rightarrow \beta\end{array}}{\mathcal{C}; \mathcal{L}; \mathcal{R} \vdash (pval/x, \sigma){:}(\mathcal{C}'; \mathcal{L}', x{:}\beta; \mathcal{R}')} \quad \textsc{Ty\_Subs\_Cons\_Log}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \mathcal{R}') \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1 \vdash res\_term \Leftarrow res\end{array}}{\mathcal{C}; \mathcal{L}; \mathcal{R}, \mathcal{R}_1 \vdash (res\_term/x, \sigma){:}(\mathcal{C}'; \mathcal{L}'; \mathcal{R}', x{:}res)} \quad \textsc{Ty\_Subs\_Cons\_Res\_Named}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \mathcal{R}') \\ \mathcal{C}; \mathcal{L}; \Phi; \overline{\cdot, res_i}^{\,i} \vdash res\_term \Leftarrow res\end{array}}{\mathcal{C}; \mathcal{L}; \mathcal{R}, \overline{\cdot, res_i}^{\,i} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \mathcal{R}', \overline{\cdot, res_i}^{\,i})} \quad \textsc{Ty\_Subs\_Cons\_Res\_Anon}$$

```
Definition rules:        199 good    0 bad
Definition rule clauses: 444 good    0 bad
```