ident, x, y, y_p, y_f, _, *abbrev*, r, α	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
	$\mid$	$\texttt{array}\, int\, \tau$		array of length int of element type τ
	$\mid$	$\tau *$		pointer to type τ
$int,\ \_$	$::=$			OCaml fixed-width integer
	$\mid$	i		literal integer
	$\mid$	n		literal integer
tag	$::=$			OCaml type for struct/union tag
	$\mid$	$ident$		
$\beta,\ \_$	$::=$			base types
	$\mid$	$\texttt{unit}$		unit
	$\mid$	$\texttt{bool}$		boolean
	$\mid$	$\texttt{integer}$		integer
	$\mid$	$\texttt{real}$		rational numbers?
	$\mid$	$\texttt{loc}$		location
	$\mid$	$\texttt{array}\, \beta$		array
	$\mid$	$\texttt{list}\, \beta$		list
	$\mid$	$\overline{\beta_i}^{\,i}$		tuple
	$\mid$	$\texttt{struct}\, tag$		struct
	$\mid$	$\texttt{set}\, \beta$		set
	$\mid$	$\texttt{opt}\, (\beta)$		option
	$\mid$	$\beta \to \beta'$		parameter types
	$\mid$	β_τ	M	of a C type
$binop$	$::=$			binary operators
	$\mid$	$\texttt{+}$		addition
	$\mid$	$\texttt{-}$		subtraction
	$\mid$	$\texttt{*}$		multiplication
	$\mid$	$\texttt{/}$		division

	|	`rem_t`	modulus
	|	`rem_f`	remainder
	|	`^`	exponentiation
	|	`=`	equality, defined both for integer and C types
	|	`!=`	inequality, similiarly defined
	|	`>`	greater than, similarly defined
	|	`<`	less than, similarly defined
	|	`>=`	greater than or equal to, similarly defined
	|	`<=`	less than or equal to, similarly defined
	|	`/\`	conjucntion
	|	`\/`	disjunction
$binop_{arith}$	::=		arithmentic binary operators
	|	`+`	
	|	`-`	
	|	`*`	
	|	`/`	
	|	`rem_t`	
	|	`rem_f`	
	|	`^`	
$binop_{rel}$	::=		relational binary operators
	|	`=`	
	|	`!=`	
	|	`>`	
	|	`<`	
	|	`>=`	
	|	`<=`	
$binop_{bool}$	::=		boolean binary operators
	|	`/\`	

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

	|	`Cons`	list cons
	|	`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_β	::=			type annotated optional identifier
	|	$\_ {:} \beta$	binders = {}	
	|	*ident*:β	binders = *ident*	
pattern	::=			
	|	*ident_opt_β*	binders = binders(*ident_opt_β*)	
	|	$\mathit{ctor\_val}(\overline{\mathit{pattern}_i}^{\,i})$	binders = $\mathrm{binders}(\overline{\mathit{pattern}_i}^{\,i})$	
z	::=			OCaml arbitrary-width integer
	|	i	M	literal integer
	|	*mem_int*	M	
	|	size_of(τ)	M	size of a C type
	|	$\mathrm{offset\_of}_{\mathit{tag}}(\mathit{member})$	M	offset of a struct member
	|	**ptr_size**	M	size of a pointer
	|	$\mathrm{max\_int}_{\tau}$	M	maximum value of int of type τ
	|	$\mathrm{min\_int}_{\tau}$	M	minimum value of int of type τ
$\mathbb{Q}$, q, _	::=			OCaml type for rational numbers
	|	$\frac{\mathit{int}_1}{\mathit{int}_2}$		
lit	::=			
	|	*ident*		
	|	**unit**		
	|	*bool*		
	|	z		
	|	$\mathbb{Q}$		

$\mathit{ident\_or\_pattern}$	$::=$			
	$\mid$	ident	binders = ident	
	$\mid$	$\mathit{pattern}$	binders = binders($\mathit{pattern}$)	
$\mathit{array\_prop}$	$::=$			array property formulas
	$\mid$	$\forall\, \overline{\mathit{ident}_i}^{\,i}\, .\, \mathit{term}_1 \rightarrow \mathit{term}_2$	bind $\overline{\mathit{ident}_i}^{\,i}$ in term_1 bind $\overline{\mathit{ident}_i}^{\,i}$ in term_2	
$\mathit{bool\_op}$	$::=$			
	$\mid$	$\neg\, \mathit{term}$		
	$\mid$	$\mathit{term}_1 = \mathit{term}_2$		
	$\mid$	$\mathit{term}_1 \rightarrow \mathit{term}_2$		
	$\mid$	$\bigwedge(\, \overline{\mathit{term}_i}^{\,i}\,)$		
	$\mid$	$\bigvee(\, \overline{\mathit{term}_i}^{\,i}\,)$		
	$\mid$	$\mathit{array\_prop}$		
	$\mid$	$\mathit{term}_1\ \mathit{binop}_{\mathit{bool}}\ \mathit{term}_2$	M	
	$\mid$	$\texttt{if}\ \mathit{term}_1\ \texttt{then}\ \mathit{term}_2\ \texttt{else}\ \mathit{term}_3$		
$\mathit{arith\_op}$	$::=$			
	$\mid$	$\mathit{term}_1 + \mathit{term}_2$		
	$\mid$	$\mathit{term}_1 - \mathit{term}_2$		
	$\mid$	$\mathit{term}_1 \times \mathit{term}_2$		
	$\mid$	$\mathit{term}_1 / \mathit{term}_2$		
	$\mid$	$\mathit{term}_1\ \texttt{rem\_t}\ \mathit{term}_2$		
	$\mid$	$\mathit{term}_1\ \texttt{rem\_f}\ \mathit{term}_2$		
	$\mid$	$\mathit{term}_1\ \hat{}\ \mathit{term}_2$		
	$\mid$	$\mathit{term}_1\ \mathit{binop}_{\mathit{arith}}\ \mathit{term}_2$	M	
$\mathit{cmp\_op}$	$::=$			
	$\mid$	$\mathit{term}_1 < \mathit{term}_2$		less than
	$\mid$	$\mathit{term}_1 \leqslant \mathit{term}_2$		less than or equal

$$
\begin{array}{lcll}
 & | & term_1 \; binop_{rel} \; term_2 & \mathsf{M} \\
 & & & \\
list\_op & ::= & & \\
 & | & \texttt{nil} & \\
 & | & term_1 :: term_2 & \\
 & | & \texttt{tl}\, term & \\
 & | & term^{(int)} & \\
 & & & \\
tuple\_op & ::= & & \\
 & | & (\, \overline{term_i}^{\,i}\,) & \\
 & | & term^{(int)} & \\
 & & & \\
pointer\_op & ::= & & \\
 & | & mem\_ptr & \\
 & | & term_1 +_{\mathrm{ptr}} term_2 & \\
 & | & \texttt{cast\_int\_to\_ptr}\, term & \\
 & | & \texttt{cast\_ptr\_to\_int}\, term & \\
 & & & \\
array\_op & ::= & & \\
 & | & [|\, \overline{term_i}^{\,i}\, |] & \\
 & | & term_1[term_2] & \\
 & | & \texttt{const}\, term & \\
 & | & term_1[term_2] := term_3 & \\
 & & & \\
param\_op & ::= & & \\
 & | & ident{:}\beta.\, term & \\
 & | & term(term_1, \, .., term_n) & \\
 & & & \\
struct\_op & ::= & & \\
 & | & term.member & \\
\end{array}
$$

ct_pred	::=			
	|	$\texttt{representable}\,(\tau, term)$		
	|	$\texttt{aligned}\,(\tau, term)$		
	|	$\texttt{alignedI}\,(term_1, term_2)$		
$term,\ \_,\ iguard$	::=			
	|	lit		
	|	$arith\_op$		
	|	$bool\_op$		
	|	cmp_op		
	|	$tuple\_op$		
	|	$struct\_op$		
	|	$pointer\_op$		
	|	$list\_op$		
	|	$array\_op$		
	|	ct_pred		
	|	$param\_op$		
	|	$(term)$	S	parentheses
	|	$\sigma(term)$	M	simul-sub σ in $term$
	|	$pval$	M	
$pexpr$	::=			pure expressions
	|	$pval$		pure values
	|	$ctor\_expr(\overline{pval_i}^{\,i})$		data constructor application
	|	$\texttt{array\_shift}\,(pval_1, \tau, pval_2)$		pointer array shift
	|	$\texttt{member\_shift}\,(pval, ident, member)$		pointer struct/union member shift
	|	$\texttt{not}\,(pval)$		boolean not
	|	$pval_1\ binop\ pval_2$		binary operations
	|	$\texttt{memberof}\,(ident, member, pval)$		C struct/union member access
	|	$name(\overline{pval_i}^{\,i})$		pure function call
	|	$\texttt{assert\_undef}\,(pval,\ UB\_name)$		

	$\mid$ `bool_to_integer` $(pval)$		
	$\mid$ `conv_int` $(\tau, pval)$		
	$\mid$ `wrapI` $(\tau, pval)$		
$tpexpr$	::=		top-level pure expressions
	$\mid$ $tpval$		top-level pure values
	$\mid$ `case` $pval$ `of` $\overline{\mid\ tpexpr\_case\_branch_i}^{\,i}$ `end`		pattern matching
	$\mid$ `let` $ident\_or\_pattern = pexpr$ `in` $tpexpr$	bind binders($ident\_or\_pattern$) in $tpexpr$	pure let
	$\mid$ `let` $ident\_or\_pattern{:}(y_1{:}\beta_1.\ term_1) = tpexpr_1$ `in` $tpexpr_2$	bind binders($ident\_or\_pattern$) in $tpexpr_2$ bind y_1 in $term_1$	annoted pure let
	$\mid$ `if` $pval$ `then` $tpexpr_1$ `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	::=		
	$\mid$ `dynamic`		
	$\mid$ `static` τ		
$bool$, _	::=		OCaml booleans
	$\mid$ `true`		
	$\mid$ `false`		
$points\_to$, pt	::=		points-to separation logic pred
	$\mid$ $term_1 \overset{init}{\mapsto}_\tau term_2$		
$qpoints\_to$, qpt	::=		quantified (integer-indexed) p
	$\mid$ $\ast x.\ iguard; term_1 + x \times \text{size\_of}(\tau) \overset{init}{\mapsto}_\tau term_2$		
res_term	::=		resource terms

	|	$\texttt{emp}$		empty heap
	|	$\mathit{points\_to}$		single-cell heap
	|	$\mathit{qpoints\_to}$		contiguous-cell heap
	|	ident		variable
	|	$\langle \mathit{res\_term}_1, \mathit{res\_term}_2 \rangle$		seperating-conjunction pair
	|	$\texttt{pack}\,(\mathit{pval}, \mathit{res\_term})$		packing for existentials
	|	$\texttt{fold}\,(\mathit{res\_term})$		fold into recursive res. pred.
	|	$\texttt{explode}\,(\mathit{res\_term}{:}\mathit{pt})$		transform points-to-array into quantified points-to
	|	$\texttt{implode}\,(\mathit{res\_term}{:}\mathit{qpt}, \mathit{int})$		transform quantified points-to into points-to-array
	|	$\texttt{break}\,(\mathit{res\_term}{:}\mathit{qpt}, \mathit{int})$		split a qpt into a qpt and a pt
	|	$\texttt{glue}\,(\mathit{res\_term}_1{:}\mathit{qpt}, \mathit{res\_term}_2{:}\mathit{pt})$		join a qpt and a pt into a qpt
	|	$\sigma(\mathit{res\_term})$	M	substitution for resource terms
$\mathit{mem\_action}$	::=			memory actions
	|	$\texttt{create}\,(\mathit{pval}, \tau)$		
	|	$\texttt{create\_readonly}\,(\mathit{pval}_1, \tau, \mathit{pval}_2)$		
	|	$\texttt{alloc}\,(\mathit{pval}_1, \mathit{pval}_2)$		
	|	$\texttt{kill}\,(\mathit{m\_kill\_kind}, \mathit{pval}, \mathit{pt})$		
	|	$\texttt{store}\,(\mathit{bool}, \tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{mem\_order}, \mathit{pt})$		true means store is locking
	|	$\texttt{load}\,(\tau, \mathit{pval}, \mathit{mem\_order}, \mathit{pt})$		
	|	$\texttt{rmw}\,(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3, \mathit{mem\_order}_1, \mathit{mem\_order}_2)$		
	|	$\texttt{fence}\,(\mathit{mem\_order})$		
	|	$\texttt{cmp\_exch\_strong}\,(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3, \mathit{mem\_order}_1, \mathit{mem\_order}_2)$		
	|	$\texttt{cmp\_exch\_weak}\,(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3, \mathit{mem\_order}_1, \mathit{mem\_order}_2)$		
	|	$\texttt{linux\_fence}\,(\mathit{linux\_mem\_order})$		
	|	$\texttt{linux\_load}\,(\tau, \mathit{pval}, \mathit{linux\_mem\_order})$		
	|	$\texttt{linux\_store}\,(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{linux\_mem\_order})$		
	|	$\texttt{linux\_rmw}\,(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{linux\_mem\_order})$		
$\mathit{polarity}$	::=			polarities for memory actions
	|			(pos) sequenced by `let weak` and `let strong`

	$\mid$	`neg`		only sequenced by `let strong`
pol_mem_action	::=			memory actions with polarity
	$\mid$	*polarity mem_action*		
mem_op	::=			operations involving the memory state
	$\mid$	$pval_1\ binop_{rel}\ pval_2$		pointer relational binary operations
	$\mid$	$pval_1 -_{\tau} pval_2$		pointer subtraction
	$\mid$	`intFromPtr` $(\tau_1, \tau_2, pval)$		cast of pointer value to integer value
	$\mid$	`ptrFromInt` $(\tau_1, \tau_2, pval)$		cast of integer value to pointer value
	$\mid$	`ptrValidForDeref` $(\tau, pval, pt)$		dereferencing validity predicate
	$\mid$	`ptrWellAligned` $(\tau, pval)$		
	$\mid$	`ptrArrayShift` $(pval_1, \tau, pval_2)$		
	$\mid$	`memcpy` $(pval_1, pval_2, pval_3)$		
	$\mid$	`memcmp` $(pval_1, pval_2, pval_3)$		
	$\mid$	`realloc` $(pval_1, pval_2, pval_3)$		
	$\mid$	`va_start` $(pval_1, pval_2)$		
	$\mid$	`va_copy` $(pval)$		
	$\mid$	`va_arg` $(pval, \tau)$		
	$\mid$	`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
	$\mid$	`done` *spine*		end of top-level expression

	|	$\texttt{undef}\ \mathit{UB\_name}$		undefined behaviour
$\mathit{res\_pattern}$	::=			resource terms
	|	$\texttt{emp}$	$\text{binders} = \{\}$	empty heap
	|	ident	$\text{binders} = \mathit{ident}$	variable
	|	$\texttt{fold}\,(\mathit{res\_pattern})$	$\text{binders} = \{\}$	unfold (recursive) predicate
	|	$\langle \mathit{res\_pattern}_1, \mathit{res\_pattern}_2 \rangle$	$\text{binders} = \text{binders}(\mathit{res\_pattern}_1) \cup \text{binders}(\mathit{res\_pattern}_2)$	seperating-conjunction pair
	|	$\texttt{pack}\,(\mathit{ident}, \mathit{res\_pattern})$	$\text{binders} = \mathit{ident} \cup \text{binders}(\mathit{res\_pattern})$	packing for existentials
$\mathit{ret\_pattern}$	::=			return pattern
	|	$\texttt{comp}\,\mathit{ident\_or\_pattern}$	$\text{binders} = \text{binders}(\mathit{ident\_or\_pattern})$	computational variable
	|	$\texttt{log}\,\mathit{ident}$	$\text{binders} = \mathit{ident}$	logical variable
	|	$\texttt{res}\,\mathit{res\_pattern}$	$\text{binders} = \text{binders}(\mathit{res\_pattern})$	resource variable
$\mathit{init},$	::=			initialisation status
	|	$\checkmark$		initialised
	|	$\times$		uninitalised
res	::=			resources
	|	$\texttt{emp}$		empty heap
	|	$\mathit{points\_to}$		points-to heap pred.
	|	$\mathit{qpoints\_to}$		quantified (integer-indexed) points-to heap pred.
	|	$\mathit{res}_1 * \mathit{res}_2$		seperating conjunction
	|	$\exists\, \mathit{ident}{:}\beta.\ \mathit{res}$		existential
	|	$\mathit{term} \wedge \mathit{res}$		logical conjuction
	|	$\texttt{if}\,\mathit{term}\,\texttt{then}\,\mathit{res}_1\,\texttt{else}\,\mathit{res}_2$		ordered disjuction
	|	$\alpha(\overline{\mathit{pval}_i}^{\,i})$		predicate
	|	$\sigma(\mathit{res})$	M	simul-sub σ in res
$\mathit{ret},\ \_$	::=			return types
	|	$\Sigma\, \mathit{ident}{:}\beta.\ \mathit{ret}$		return a computational value

	|	$\exists\, ident{:}\beta.\ ret$		return a logical value
	|	$res \otimes ret$		return a resource value
	|	$term \wedge ret$		return a predicate (post-condition
	|	$\mathtt{I}$		end return list
	|	$\sigma(ret)$	M	simul-sub σ in ret
seq_expr	::=			sequential (effectful) expressions
	|	$\mathtt{ccall}\,(\tau, ident, spine)$		C function call
	|	$\mathtt{pcall}\,(name, spine)$		procedure call
seq_texpr	::=			sequential top-level (effectful) expres
	|	$tval$		(effectful) top-level values
	|	$\mathtt{run}\, ident\, \overline{pval_i}^{\,i}$		run from label
	|	$\mathtt{let}\, ident\_or\_pattern = pexpr\, \mathtt{in}\, texpr$	bind binders($ident\_or\_pattern$) in $texpr$	pure let
	|	$\mathtt{let}\, ident\_or\_pattern{:}(y_1{:}\beta_1.\ term_1) = tpexpr\, \mathtt{in}\, texpr$	bind binders($ident\_or\_pattern$) in $texpr$ bind y_1 in $term_1$	annotated pure let
	|	$\mathtt{let}\, \overline{ret\_pattern_i}^{\,i} = seq\_expr\, \mathtt{in}\, texpr$	bind binders($\overline{ret\_pattern_i}^{\,i}$) in $texpr$	bind return patterns
	|	$\mathtt{let}\, \overline{ret\_pattern_i}^{\,i}{:}ret = texpr_1\, \mathtt{in}\, texpr_2$	bind binders($\overline{ret\_pattern_i}^{\,i}$) in $texpr_2$	annotated bind return patterns
	|	$\mathtt{case}\, pval\, \mathtt{of}\, \overline{\mid texpr\_case\_branch_i}^{\,i}\, \mathtt{end}$		pattern matching
	|	$\mathtt{if}\, pval\, \mathtt{then}\, texpr_1\, \mathtt{else}\, texpr_2$		conditional
	|	$\mathtt{bound}\,[int](is\_texpr)$		limit scope of indet seq behaviour
$texpr\_case\_branch$	::=			top-level case expression branch
	|	$pattern \Rightarrow texpr$	bind binders($pattern$) in $texpr$	top-level case expression branch
is_expr	::=			indet seq (effectful) expressions
	|	$tval$		(effectful) top-level values
	|	$\mathtt{memop}\,(mem\_op)$		pointer op involving memory
	|	pol_mem_action		memory action
is_texpr	::=			indet seq top-level (effectful) express

	$\mid$	$\texttt{let}\,\texttt{weak}\,\overline{ret\_pattern_i}^{\,i} = is\_expr\ \texttt{in}\ texpr$	$\mathsf{bind}\ \mathrm{binders}(\overline{ret\_pattern_i}^{\,i})\ \mathsf{in}\ texpr$	weak sequencing
	$\mid$	$\texttt{let}\,\texttt{strong}\,\overline{ret\_pattern_i}^{\,i} = is\_expr\ \texttt{in}\ texpr$	$\mathsf{bind}\ \mathrm{binders}(\overline{ret\_pattern_i}^{\,i})\ \mathsf{in}\ texpr$	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
	$\mid$	$\Sigma\, ident{:}\beta.\ pure\_ret$		
	$\mid$	$term \wedge pure\_ret$		
	$\mid$	$\texttt{I}$		
$\mathcal{C}$	$::=$			computational var env
	$\mid$	$\cdot$		
	$\mid$	$\mathcal{C}, ident{:}\beta$		
	$\mid$	$\overline{\mathcal{C}_i}^{\,i}$		

$\mathcal{L}$	$::=$			logical var env
	$\mid$	$\cdot$		
	$\mid$	$\overline{\mathcal{L}_i}^{\,i}$		
	$\mid$	$\mathcal{L}, \mathit{ident}{:}\beta$		
Φ	$::=$			constraints env
	$\mid$	$\cdot$		
	$\mid$	Φ, term		
	$\mid$	$\overline{\Phi_i}^{\,i}$		
$\mathcal{R}$	$::=$			resources env
	$\mid$	$\cdot$		
	$\mid$	$\mathcal{R}, \mathit{ident}{:}\mathit{res}$		
	$\mid$	$\overline{\mathcal{R}_i}^{\,i}$		
$\sigma,\ \psi$	$::=$			substitutions
	$\mid$	$\cdot$		
	$\mid$	$\mathit{spine\_elem}/\mathit{ident}, \sigma$		
	$\mid$	$\mathit{term}/\mathit{ident}, \sigma$		
	$\mid$	$\overline{\sigma_i}^{\,i}$		
	$\mid$	$\sigma(\psi)$	M	apply σ to all elements in ψ
typing	$::=$			
	$\mid$	$\mathtt{smt}\,(\Phi \Rightarrow \mathit{term})$		
	$\mid$	$\mathit{ident}{:}\beta \in \mathcal{C}$		
	$\mid$	$\mathit{ident}{:}\beta \in \mathcal{L}$		
	$\mid$	$\mathtt{struct}\,\mathit{tag}\ \&\,\overline{\mathit{member}_i{:}\tau_i}^{\,i} \in \mathtt{Globals}$		
	$\mid$	$\alpha \equiv \overline{x_i{:}\beta_i}^{\,i} \mapsto \mathit{res} \in \mathtt{Globals}$		recursive resource predicate
	$\mid$	$\overline{\mathcal{C}_i; \mathcal{L}_i; \Phi_i \vdash \mathit{mem\_val}_i \Rightarrow \mathtt{mem}\,\beta_i}^{\,i}$		dependent on memory object model
	$\mid$	$\overline{\mathcal{C}_j; \mathcal{L}_j \mid \overline{\mathit{ident}_{i\,j}}^{\,i} \vdash \mathtt{guarded}\,(\mathit{term}_j)}^{\,j}$		
	$\mid$	$\overline{\mathcal{C}_j; \mathcal{L}_j \mid \overline{\mathit{ident}_{i\,j}}^{\,i} \vdash \mathtt{vconstr}\,(\mathit{term}_j)}^{\,j}$		

	$\vert$	$\mathit{ident} \in \mathcal{C};\mathcal{L}$	
	$\vert$	$\mathit{ident} \in \overline{\mathit{ident}_i}^{\,i}$	
opsem	::=		
	$\vert$	$\forall\, i < j.\ \mathtt{not}\,(\mathit{pattern}_i = \mathit{pval} \rightsquigarrow \sigma_i)$	
	$\vert$	$\mathtt{fresh}\,(\mathit{mem\_ptr})$	
	$\vert$	term	
	$\vert$	$\mathit{pval}{:}\beta$	
$\mathit{formula}$	::=		
	$\vert$	$\mathit{judgement}$	
	$\vert$	typing	
	$\vert$	opsem	
	$\vert$	$\mathit{res} \equiv \mathit{res}'$	
	$\vert$	$\mathit{term} \equiv \mathit{term}'$	
	$\vert$	$\mathit{name}{:}\mathit{pure\_arg} \equiv \overline{x_i}^{\,i} \mapsto \mathit{tpexpr} \in \mathtt{Globals}$	
	$\vert$	$\mathit{name}{:}\mathit{arg} \equiv \overline{x_i}^{\,i} \mapsto \mathit{texpr} \in \mathtt{Globals}$	
$\mathit{heap},\ h,\ f$	::=		heaps
	$\vert$	$\cdot$	
	$\vert$	$h + \{\mathit{points\_to}\}$	
	$\vert$	$h + f$	[O] convenient for the soundness proof
$\mathit{wf\_jtyp}$	::=		
	$\vert$	$\mathcal{C};\mathcal{L} \vdash \mathtt{guarded\_e}\,(\mathit{term})$	
	$\vert$	$\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \mathtt{guarded}\,(\mathit{term})$	
	$\vert$	$\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \mathtt{guarded}\,(\mathit{term})$	
	$\vert$	$\mathcal{C};\mathcal{L} \vdash \mathtt{well\_formed}\,(\mathit{array\_prop})$	
$\mathit{lemma\_jtype}$	::=		
	$\vert$	$\overline{x_i}^{\,i} :: \mathit{arg} \rightsquigarrow \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \mid \mathit{ret}$	

$$
\begin{array}{lcl}
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \sqsubseteq \mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}' \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash (\sigma){:}(\mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}') \\
 & & \\
\mathit{res\_jtype} & ::= & \\
 & | & \Phi \vdash \mathit{res} \ \equiv \ \mathit{res}' \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{res} \\
 & | & h{:}\mathcal{R} \\
 & & \\
\mathit{object\_value\_jtype} & ::= & \\
 & | & \mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \ \mathtt{obj}\,\beta \\
 & & \\
\mathit{pval\_jtype} & ::= & \\
 & | & \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta \\
 & & \\
\mathit{spine\_jtype} & ::= & \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} \ :: \mathit{arg} \gg \sigma;\mathit{ret} \\
 & & \\
\mathit{pexpr\_jtype} & ::= & \\
 & | & \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pexpr} \Rightarrow \mathit{ident}{:}\beta.\ \mathit{term} \\
 & & \\
\mathit{comp\_pattern\_jtype} & ::= & \\
 & | & \mathit{pattern}{:}\beta \rightsquigarrow \mathcal{C}\,\mathtt{with}\,\mathit{term} \\
 & | & \mathit{ident\_or\_pattern}{:}\beta \rightsquigarrow \mathcal{C}\,\mathtt{with}\,\mathit{term} \\
 & & \\
\mathit{res\_pattern\_jtype} & ::= & \\
 & | & \Phi \vdash \mathit{res}' \ = \ \mathtt{strip\_ifs}\,(\mathit{res}) \\
 & | & \Phi \vdash \mathit{res}\ \mathtt{as}\ \mathit{res\_pattern} \rightsquigarrow \mathcal{L}';\Phi';\mathcal{R}' \\
 & | & \Phi \vdash \mathit{res\_pattern}{:}\mathit{res} \rightsquigarrow \mathcal{L}';\Phi';\mathcal{R}' \\
 & & \\
\mathit{ret\_pattern\_jtype} & ::= & \\
 & | & \Phi \vdash \ \overline{\mathit{ret\_pattern}_i}^{\,i} \ {:}\mathit{ret} \rightsquigarrow \mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}'
\end{array}
$$

$$
\begin{array}{lll}
\mathit{tpval\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpval} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term} \\[1em]
\mathit{tpexpr\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpexpr} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term} \\[1em]
\mathit{action\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_action} \Rightarrow \mathit{ret} \\[1em]
\mathit{memop\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_op} \Rightarrow \mathit{ret} \\[1em]
\mathit{tval\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{tval} \Leftarrow \mathit{ret} \\[1em]
\mathit{seq\_expr\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_expr} \Rightarrow \mathit{ret} \\[1em]
\mathit{is\_expr\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_expr} \Rightarrow \mathit{ret} \\[1em]
\mathit{texpr\_jtype} & ::= & \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret} \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret} \\
& | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{texpr} \Leftarrow \mathit{ret} \\[1em]
\mathit{subs\_jtype} & ::= & \\
& | & \mathit{pattern} = \mathit{pval} \leadsto \sigma \\
& | & \mathit{ident\_or\_pattern} = \mathit{pval} \leadsto \sigma \\
& | & \mathit{res\_pattern} = \mathit{res\_term} \leadsto \sigma \\
& | & \overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \leadsto \sigma
\end{array}
$$

$$
\begin{array}{lll}
 & | & \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret \\
 & & \\
pure\_opsem\_jtype & ::= & \\
 & | & \langle pexpr \rangle \longrightarrow \langle pexpr' \rangle \\
 & | & \langle pexpr \rangle \longrightarrow \langle tpexpr{:}(y{:}\beta.\, term) \rangle \\
 & | & \langle tpexpr \rangle \longrightarrow \langle tpexpr' \rangle \\
 & & \\
opsem\_jtype & ::= & \\
 & | & \langle h; seq\_expr \rangle \longrightarrow \langle h'; texpr{:}ret \rangle \\
 & | & \langle h; seq\_texpr \rangle \longrightarrow \langle h'; texpr \rangle \\
 & | & \langle h; mem\_op \rangle \longrightarrow \langle h'; tval \rangle \\
 & | & \langle h; mem\_action \rangle \longrightarrow \langle h'; tval \rangle \\
 & | & \langle h; is\_expr \rangle \longrightarrow \langle h'; is\_expr' \rangle \\
 & | & \langle h; is\_texpr \rangle \longrightarrow \langle h'; texpr \rangle \\
 & | & \langle h; texpr \rangle \longrightarrow \langle h'; texpr' \rangle
\end{array}
$$

$\boxed{\mathcal{C}; \mathcal{L} \vdash \ \texttt{guarded\_e}\,(term)}$

$$
\frac{ident \ \in \ \mathcal{C}; \mathcal{L}}{\mathcal{C}; \mathcal{L} \vdash \ \texttt{guarded\_e}\,(ident)} \quad \textsc{Wf\_Guarded\_Eexpr\_EVar}
$$

$$
\frac{ident \ \in \ \mathcal{C}; \mathcal{L}}{\mathcal{C}; \mathcal{L} \vdash \ \texttt{guarded\_e}\,(z \times ident)} \quad \textsc{Wf\_Guarded\_Eexpr\_Scaled\_EVar}
$$

$$
\frac{\begin{array}{c}\mathcal{C}; \mathcal{L} \vdash \ \texttt{guarded\_e}\,(term_1) \\ \mathcal{C}; \mathcal{L} \vdash \ \texttt{guarded\_e}\,(term_2)\end{array}}{\mathcal{C}; \mathcal{L} \vdash \ \texttt{guarded\_e}\,(term_1 + term_2)} \quad \textsc{Wf\_Guarded\_Eexpr\_Plus}
$$

$\boxed{\mathcal{C}; \mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \ \texttt{guarded}\,(term)}$

$$\frac{\mathcal{C};\mathcal{L} \vdash \ \texttt{guarded\_e}\,(\mathit{term})}{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term})} \quad \text{WF\_GUARDED\_EXPR\_EEXPR}$$

$$\frac{\mathit{ident} \in \overline{\mathit{ident}_i}^{\,i}}{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{ident})} \quad \text{WF\_GUARDED\_EXPR\_UVAR}$$

$$\boxed{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term})}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term}) \\ \mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term}')\end{array}}{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term} \leqslant \mathit{term}')} \quad \text{WF\_GUARDED\_LEQ}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term}) \\ \mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term}')\end{array}}{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term} = \mathit{term}')} \quad \text{WF\_GUARDED\_EQ}$$

$$\frac{\overline{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term}_j)}^{\,j}}{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\bigvee(\overline{\mathit{term}_j}^{\,j}))} \quad \text{WF\_GUARDED\_OR}$$

$$\frac{\overline{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term}_j)}^{\,j}}{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\bigwedge(\overline{\mathit{term}_j}^{\,j}))} \quad \text{WF\_GUARDED\_AND}$$

$$\frac{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\mathit{term})}{\mathcal{C};\mathcal{L} \mid \ \overline{\mathit{ident}_i}^{\,i} \ \vdash \ \texttt{guarded}\,(\neg\, \mathit{term})} \quad \text{WF\_GUARDED\_NEG}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(term_1)\\ \mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(term_2)\\ \mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(term_3)\end{array}}{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(\texttt{if}\,term_1\,\texttt{then}\,term_2\,\texttt{else}\,term_3)}\ \textsc{Wf\_Guarded\_ITE}$$

$\boxed{\mathcal{C};\mathcal{L} \vdash \texttt{well\_formed}\,(array\_prop)}$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(term_1)\\ \mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{vconstr}\,(term_2)\end{array}}{\mathcal{C};\mathcal{L} \vdash \texttt{well\_formed}\,(\forall\,\overline{ident_i}^{\,i}\,.\,term_1 \to term_2)}\ \textsc{Wf\_Array\_Prop\_Base}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash(\sigma){:}(\mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}')\\ \mathtt{smt}\,(\Phi\Rightarrow\mathit{term})\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash(\sigma){:}(\mathcal{C}';\mathcal{L}';\Phi',\mathit{term};\mathcal{R}')}\quad\textsc{Ty\_Subs\_Cons\_Phi}$$

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

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

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

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

$$\frac{\mathtt{smt}\,(\Phi\Rightarrow(\mathit{iguard}\to\mathit{iguard}')\wedge(\mathit{iguard}'\to\mathit{iguard})\wedge(\mathit{term}_1=\mathit{term}_1')\wedge(\mathit{term}_2=\mathit{term}_2'))}{\Phi\vdash\circledast x.\ \mathit{iguard};\mathit{term}_1+x\times\mathrm{size\_of}(\tau)\overset{\mathit{init}}{\mapsto}_\tau\mathit{term}_2\ \equiv\ \circledast x.\ \mathit{iguard}';\mathit{term}_1'+x\times\mathrm{size\_of}(\tau)\overset{\mathit{init}}{\mapsto}_\tau\mathit{term}_2'}\quad\textsc{Ty\_Res\_Eq\_QPointsTo}$$

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

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

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

$$\frac{\begin{array}{l}\mathtt{smt}\,(\Phi \Rightarrow (\mathit{term}_1 \to \mathit{term}_2) \wedge (\mathit{term}_2 \to \mathit{term}_1))\\ \Phi \vdash \mathit{res}_{11} \;\equiv\; \mathit{res}_{21}\\ \Phi \vdash \mathit{res}_{21} \;\equiv\; \mathit{res}_{22}\end{array}}{\Phi \vdash \mathtt{if}\ \mathit{term}_1\ \mathtt{then}\ \mathit{res}_{11}\ \mathtt{else}\ \mathit{res}_{12} \;\equiv\; \mathtt{if}\ \mathit{term}_2\ \mathtt{then}\ \mathit{res}_{21}\ \mathtt{else}\ \mathit{res}_{22}} \quad \textsc{Ty\_Res\_Eq\_OrdDisj}$$

$$\frac{}{\Phi \vdash \alpha(\overline{\mathit{pval}_i}^{\,i}) \;\equiv\; \alpha(\overline{\mathit{pval}_i}^{\,i})} \quad \textsc{Ty\_Res\_Eq\_Pred}$$

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

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

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

$$\frac{\begin{array}{l}\Phi \vdash \mathit{qpoints\_to} \;\equiv\; \mathit{qpoints\_to}'\\ \Phi \vdash \mathit{qpoints\_to}' \;\equiv\; \mathit{qpoints\_to}''\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot, \_{:}\mathit{qpoints\_to} \vdash \mathit{qpoints\_to}' \Leftarrow \mathit{qpoints\_to}''} \quad \textsc{Ty\_Res\_QPointsTo}$$

$$\frac{\begin{array}{l}\Phi \vdash \mathit{res}_1' = \mathtt{strip\_ifs}\,(\mathit{res}_1)\\ \Phi \vdash \mathit{res}_2' = \mathtt{strip\_ifs}\,(\mathit{res}_2)\\ \Phi \vdash \mathit{res}_1' \;\equiv\; \mathit{res}_2'\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot, r{:}\mathit{res}_1 \vdash r \Leftarrow \mathit{res}_2} \quad \textsc{Ty\_Res\_Var}$$

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

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

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

$$\frac{\begin{array}{l}\alpha \ \equiv\ \overline{x_i{:}\beta_i}^{\,i} \ \mapsto \mathit{res} \in \mathtt{Globals}\\ \overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_i \Rightarrow \beta_i}^{\,i}\\ \Phi \vdash \mathit{res}' = \mathtt{strip\_ifs}\,(\overline{\mathit{pval}_i/x_i,\cdot}^{\,i}(\mathit{res}))\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{res}'\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{fold}\,(\mathit{res\_term}) \Leftarrow \alpha(\overline{\mathit{pval}_i}^{\,i})}\ \textsc{Ty\_Res\_Fold}$$

$$\frac{\begin{array}{l}\mathit{pt} \ \equiv\ \mathit{term}_1 \overset{\mathit{init}}{\mapsto}_{\mathtt{array}\,n\,\tau} \mathit{term}_2\\ \mathit{qpt} \ \equiv\ \circledast x.\, 0 \leqslant x \wedge x \leqslant n+1; \mathit{term}_1 + x \times \text{size\_of}(\tau) \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2[x]\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{pt}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{explode}\,(\mathit{res\_term}{:}\mathit{pt}) \Leftarrow \mathit{qpt}}\ \textsc{Ty\_Res\_Explode}$$

$$\frac{\begin{array}{l}\mathit{qpt} \ \equiv\ \circledast x.\, \mathit{iguard}; \mathit{term}_1 + x \times \text{size\_of}(\tau) \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2\\ \mathit{pt} \ \equiv\ \mathit{term}_1' \overset{\mathit{init}}{\mapsto}_{\mathtt{array}\,n\,\tau} \mathit{term}_2'\\ \mathit{iguard}' = (0 \leqslant x \wedge x \leqslant n+1)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \circledast x.\, \mathit{iguard}; \mathit{term}_1 + x \times \text{size\_of}(\tau) \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2\\ \mathtt{smt}\,(\Phi \Rightarrow (\mathit{iguard} \to \mathit{iguard}') \wedge (\mathit{iguard}' \to \mathit{iguard}) \wedge (\mathit{term}_1 = \mathit{term}_1') \wedge (\mathit{term}_2 = \mathit{term}_2'[x]))\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{implode}\,(\mathit{res\_term}{:}\mathit{qpt}, n) \Leftarrow \mathit{pt}}\ \textsc{Ty\_Res\_Implode}$$

$$\frac{\begin{array}{l} qpt \equiv \circledast x.\ iguard; term_1 + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau term_2 \\ qpt' \equiv \circledast x.\ iguard'; term_1 + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau term_2 \\ pt \equiv term_1 + i \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau i/x, \cdot(term_2) \\ \mathtt{smt}\,(\Phi, iguard' \Rightarrow x\ \mathtt{!=}\ i) \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash res\_term \Leftarrow qpt \end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathtt{break}\,(res\_term{:}qpt, i) \Leftarrow qpt' * pt} \quad \textsc{Ty\_Res\_Break}$$

$$\frac{\begin{array}{l} qpt' \equiv \circledast x.\ iguard'; term_1 + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau term_2' \\ pt \equiv term_1'' \overset{init}{\mapsto}_\tau term_2'' \\ i \equiv (term_1'' - term_1)/\mathrm{size\_of}(\tau) \\ qpt \equiv \circledast x.\ iguard; term_1 + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau \mathtt{if}\ x = i\ \mathtt{then}\ term_2''\ \mathtt{else}\ term_2' \\ \mathtt{smt}\,(\Phi \Rightarrow (iguard \to (iguard' \lor x = i)) \land ((iguard' \lor x = i) \to iguard)) \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1 \vdash res\_term_1 \Leftarrow qpt' \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_2 \vdash res\_term_2 \Leftarrow pt \end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R}_2 \vdash \mathtt{glue}\,(res\_term_1{:}qpt', res\_term_2{:}pt) \Leftarrow qpt} \quad \textsc{Ty\_Res\_Glue}$$

$\boxed{h{:}\mathcal{R}}$

$$\frac{}{\cdot{:}\cdot} \quad \textsc{Ty\_Heap\_Emp}$$

$$\frac{\begin{array}{l} h{:}\mathcal{R} \\ \cdot; \cdot; \cdot; \mathcal{R}' \vdash pt \Leftarrow pt \end{array}}{h + \{pt\}{:}\mathcal{R}, \mathcal{R}'} \quad \textsc{Ty\_Heap\_PointsTo}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{value}_i \Rightarrow \beta}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash \beta[\,\overline{\mathit{value}_i}^{\,i}\,] \Rightarrow \texttt{list}\,\beta} \quad \textsc{Ty\_Pval\_List}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{value}_i \Rightarrow \beta_i}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash (\,\overline{\mathit{value}_i}^{\,i}\,) \Rightarrow \overline{\beta_i}^{\,i}} \quad \textsc{Ty\_Pval\_Tuple}$$

$$\frac{\texttt{smt}\,(\Phi \Rightarrow \texttt{false})}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{error}\,(\mathit{string}, \mathit{pval}) \Rightarrow \beta} \quad \textsc{Ty\_Pval\_Error}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Nil}\,\beta(\,) \Rightarrow \texttt{list}\,\beta} \quad \textsc{Ty\_Pval\_Ctor\_Nil}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_2 \Rightarrow \texttt{list}\,\beta\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Cons}(\mathit{pval}_1, \mathit{pval}_2) \Rightarrow \texttt{list}\,\beta} \quad \textsc{Ty\_Pval\_Ctor\_Cons}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_i \Rightarrow \beta_i}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Tuple}(\,\overline{\mathit{pval}_i}^{\,i}\,) \Rightarrow \overline{\beta_i}^{\,i}} \quad \textsc{Ty\_Pval\_Ctor\_Tuple}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_i \Rightarrow \beta}^{\,i}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Array}(\,\overline{\mathit{pval}_i}^{\,i}\,) \Rightarrow \texttt{array}\,\beta} \quad \textsc{Ty\_Pval\_Ctor\_Array}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Specified}(\mathit{pval}) \Rightarrow \beta} \quad \textsc{Ty\_Pval\_Ctor\_Specified}$$

$$\frac{\begin{array}{l}\texttt{struct}\, tag \,\&\, \overline{member_i{:}\tau_i}^{\,i} \in \texttt{Globals} \\ \overline{\mathcal{C};\mathcal{L};\Phi \vdash pval_i \Rightarrow \beta_{\tau_i}}^{\,i}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash (\,\texttt{struct}\, tag)\{\,\overline{.\,member_i = pval_i}^{\,i}\,\} \Rightarrow \texttt{struct}\, tag} \quad \text{TY\_PVAL\_STRUCT}$$

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

$$\frac{}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \;::ret \gg \cdot; ret} \quad \text{TY\_SPINE\_EMPTY}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash x = pval,\, \overline{x_i = spine\_elem_i}^{\,i} :: \Pi\, x{:}\beta.\; arg \gg pval/x, \sigma; ret} \quad \text{TY\_SPINE\_COMP}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash x = pval,\, \overline{x_i = spine\_elem_i}^{\,i} :: \forall\, x{:}\beta.\; arg \gg pval/x, \sigma; ret} \quad \text{TY\_SPINE\_LOG}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1 \vdash res\_term \Leftarrow res \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}_2 \vdash \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1,\mathcal{R}_2 \vdash x = res\_term,\, \overline{x_i = spine\_elem_i}^{\,i} :: res \multimap arg \gg res\_term/x, \sigma; ret} \quad \text{TY\_SPINE\_RES}$$

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow term) \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = spine\_elem_i}^{\,i} :: term \supset arg \gg \sigma; ret} \quad \text{TY\_SPINE\_PHI}$$

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

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \beta}{\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow y{:}\beta.\; y = pval} \quad \text{TY\_PE\_VAL}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \texttt{loc}\\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{integer}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{array\_shift}\,(pval_1, \tau, pval_2) \Rightarrow y{:}\texttt{loc}.\ y = pval_1 +_{\mathrm{ptr}} (pval_2 \times \mathrm{size\_of}(\tau))} \quad \textsc{Ty\_PE\_Array\_Shift}$$

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

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \texttt{integer}\\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{integer}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash pval_1\ binop_{arith}\ pval_2 \Rightarrow y{:}\texttt{integer}.\ y = (pval_1\ binop_{arith}\ pval_2)} \quad \textsc{Ty\_PE\_Arith\_Binop}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \texttt{integer}\\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{integer}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash pval_1\ binop_{rel}\ pval_2 \Rightarrow y{:}\texttt{bool}.\ y = (pval_1\ binop_{rel}\ pval_2)} \quad \textsc{Ty\_PE\_Rel\_Binop}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \texttt{bool}\\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{bool}\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash pval_1\ binop_{bool}\ pval_2 \Rightarrow y{:}\texttt{bool}.\ y = (pval_1\ binop_{bool}\ pval_2)} \quad \textsc{Ty\_PE\_Bool\_Binop}$$

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

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

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

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

$\boxed{\mathit{pattern}{:}\beta \leadsto \mathcal{C}\,\texttt{with}\,\mathit{term}}$

$$\frac{}{\_{:}\beta{:}\beta \leadsto \cdot\,\texttt{with}\,\_} \quad \textsc{Ty\_Pat\_Comp\_No\_Sym\_Annot}$$

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

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

$$\frac{\begin{array}{l}\mathit{pattern}_1{:}\beta \leadsto \mathcal{C}_1\,\texttt{with}\,\mathit{term}_1\\ \mathit{pattern}_2{:}\texttt{list}\,\beta \leadsto \mathcal{C}_2\,\texttt{with}\,\mathit{term}_2\end{array}}{\texttt{Cons}(\mathit{pattern}_1,\mathit{pattern}_2){:}\texttt{list}\,\beta \leadsto \mathcal{C}_1,\mathcal{C}_2\,\texttt{with}\,\mathit{term}_1 :: \mathit{term}_2} \quad \textsc{Ty\_Pat\_Comp\_Cons}$$

$$\frac{\overline{\mathit{pattern}_i{:}\beta_i \leadsto \mathcal{C}_i\,\texttt{with}\,\mathit{term}_i}^{\,i}}{\texttt{Tuple}(\overline{\mathit{pattern}_i}^{\,i}){:}\overline{\beta_i}^{\,i} \leadsto \overline{\mathcal{C}_i}^{\,i}\,\texttt{with}\,(\overline{\mathit{term}_i}^{\,i})} \quad \textsc{Ty\_Pat\_Comp\_Tuple}$$

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

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

$\boxed{ident\_or\_pattern{:}\beta \leadsto \mathcal{C}\ \mathtt{with}\ term}$

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

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

$\boxed{\Phi \vdash res' = \mathtt{strip\_ifs}\,(res)}$

$$\frac{}{\Phi \vdash \mathtt{emp} = \mathtt{strip\_ifs}\,(\mathtt{emp})}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_Empty}$$

$$\frac{}{\Phi \vdash pt = \mathtt{strip\_ifs}\,(pt)}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_PointsTo}$$

$$\frac{}{\Phi \vdash res_1 * res_2 = \mathtt{strip\_ifs}\,(res_1 * res_2)}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_SepConj}$$

$$\frac{}{\Phi \vdash \exists\, x{:}\beta.\ res = \mathtt{strip\_ifs}\,(\exists\, x{:}\beta.\ res)}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_Exists}$$

$$\frac{}{\Phi \vdash term \wedge res = \mathtt{strip\_ifs}\,(term \wedge res)}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_TermConj}$$

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow \mathit{term})\\ \Phi \vdash \mathit{res}_1' = \texttt{strip\_ifs}\,(\mathit{res}_1')\end{array}}{\Phi \vdash \mathit{res}_1' = \texttt{strip\_ifs}\,(\texttt{if}\,\mathit{term}\,\texttt{then}\,\mathit{res}_1\,\texttt{else}\,\mathit{res}_2)}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_True}$$

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow \neg\,\mathit{term})\\ \Phi \vdash \mathit{res}_2' = \texttt{strip\_ifs}\,(\mathit{res}_2)\end{array}}{\Phi \vdash \mathit{res}_2' = \texttt{strip\_ifs}\,(\texttt{if}\,\mathit{term}\,\texttt{then}\,\mathit{res}_1\,\texttt{else}\,\mathit{res}_2)}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_False}$$

$$\frac{}{\Phi \vdash \texttt{if}\,\mathit{term}\,\texttt{then}\,\mathit{res}_1\,\texttt{else}\,\mathit{res}_2 = \texttt{strip\_ifs}\,(\texttt{if}\,\mathit{term}\,\texttt{then}\,\mathit{res}_1\,\texttt{else}\,\mathit{res}_2)}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_UnderDet}$$

$$\frac{}{\Phi \vdash \alpha(\overline{\mathit{pval}_i}^{\,i}) = \texttt{strip\_ifs}\,(\alpha(\overline{\mathit{pval}_i}^{\,i}))}\quad\textsc{Ty\_Pat\_Res\_StripIfs\_Pred}$$

$$\boxed{\Phi \vdash \mathit{res}\;\texttt{as}\;\mathit{res\_pattern} \rightsquigarrow \mathcal{L}';\Phi';\mathcal{R}'}$$

$$\frac{}{\Phi \vdash \texttt{emp}\;\texttt{as}\;\texttt{emp} \rightsquigarrow \cdot;\cdot;\cdot}\quad\textsc{Ty\_Pat\_Res\_Match\_Empty}$$

$$\frac{}{\Phi \vdash \mathit{res}\;\texttt{as}\;r \rightsquigarrow \cdot;\cdot;\cdot, r{:}\mathit{res}}\quad\textsc{Ty\_Pat\_Res\_Match\_Var}$$

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

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

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

$$\frac{\begin{array}{c}\alpha \equiv \overline{x_i{:}\beta_i}^{\,i} \mapsto \mathit{res} \in \texttt{Globals}\\ \Phi \vdash \mathit{res\_pattern}{:}\overline{\mathit{pval}_i/x_i,\cdot}^{\,i}(\mathit{res}) \leadsto \mathcal{L}';\Phi';\mathcal{R}'\end{array}}{\Phi \vdash \alpha(\,\overline{\mathit{pval}_i}^{\,i}\,)\,\texttt{as}\,\texttt{fold}\,(\mathit{res\_pattern}) \leadsto \mathcal{L}';\Phi';\mathcal{R}'}\quad \textsc{Ty\_Pat\_Res\_Match\_Fold}$$

$$\boxed{\Phi \vdash \mathit{res\_pattern}{:}\mathit{res} \leadsto \mathcal{L}';\Phi';\mathcal{R}'}$$

$$\frac{\begin{array}{c}\Phi \vdash \mathit{res}' = \texttt{strip\_ifs}\,(\mathit{res})\\ \Phi \vdash \mathit{res}'\,\texttt{as}\,\mathit{res\_pattern} \leadsto \mathcal{L}';\Phi';\mathcal{R}'\end{array}}{\Phi \vdash \mathit{res\_pattern}{:}\mathit{res} \leadsto \mathcal{L}';\Phi';\mathcal{R}'}\quad \textsc{Ty\_Pat\_Res\_Strip\_Ifs}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\mathcal{C}; \mathcal{L}; \Phi \vdash pval \Rightarrow \texttt{integer}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot \vdash \texttt{create}\,(pval, \tau) \Rightarrow \Sigma\, y_p{:}\texttt{loc}.\ \texttt{representable}\,(\tau*, y_p) \wedge \texttt{alignedI}\,(pval, y_p) \wedge \exists\, y{:}\beta_\tau.\ y_p(1) \overset{\times}{\mapsto}_\tau y \otimes \texttt{I}} \quad \textsc{Ty\_Action\_Create}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \Phi \vdash pval_0 \Rightarrow \texttt{loc}\\ \texttt{smt}\,(\Phi \Rightarrow pval_0 = pval_1)\\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash pval_1(1) \overset{\checkmark}{\mapsto}_\tau pval_2 \Leftarrow pval_1(1) \overset{\checkmark}{\mapsto}_\tau pval_2\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{load}\,(\tau, pval_0, \_, pval_1(1) \overset{\checkmark}{\mapsto}_\tau pval_2) \Rightarrow \Sigma\, y{:}\beta_\tau.\ y = pval_2 \wedge pval_1(1) \overset{\checkmark}{\mapsto}_\tau pval_2 \otimes \texttt{I}} \quad \textsc{Ty\_Action\_Load}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \Phi \vdash pval_0 \Rightarrow \texttt{loc}\\ \mathcal{C}; \mathcal{L}; \Phi \vdash pval_1 \Rightarrow \beta_\tau\\ \texttt{smt}\,(\Phi \Rightarrow \texttt{representable}\,(\tau, pval_1))\\ \texttt{smt}\,(\Phi \Rightarrow pval_2 = pval_0)\\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash pval_2(1) \mapsto_\tau \_ \Leftarrow pval_2(1) \mapsto_\tau \_\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{store}\,(\_, \tau, pval_0, pval_1, \_, pval_2(1) \mapsto_\tau \_) \Rightarrow \Sigma\, \_{:}\texttt{unit}.\ pval_2(1) \overset{\checkmark}{\mapsto}_\tau pval_1 \otimes \texttt{I}} \quad \textsc{Ty\_Action\_Store}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \Phi \vdash pval_0 \Rightarrow \texttt{loc}\\ \texttt{smt}\,(\Phi \Rightarrow pval_0 = pval_1)\\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash pval_1(1) \mapsto_\tau \_ \Leftarrow pval_1(1) \mapsto_\tau \_\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{kill}\,(\texttt{static}\,\tau, pval_0, pval_1(1) \mapsto_\tau \_) \Rightarrow \Sigma\, \_{:}\texttt{unit}.\ \texttt{I}} \quad \textsc{Ty\_Action\_Kill\_Static}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c}\mathit{ident}{:}\mathit{arg}\ \equiv\ \overline{x_i}^{\,i}\ \mapsto \mathit{texpr}\ \in\ \mathtt{Globals}\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i}\ :: \mathit{arg} \gg \sigma; \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{ccall}\,(\tau, \mathit{ident}, \overline{\mathit{spine\_elem}_i}^{\,i}) \Rightarrow \sigma(\mathit{ret})} \quad \textsc{Ty\_Seq\_E\_CCall}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\overline{\mathit{pattern}_i = \mathit{pval}_i \rightsquigarrow \sigma_i}^{\,i}}{\mathtt{Array}(\,\overline{\mathit{pattern}_i}^{\,i}\,) = \mathtt{Array}(\,\overline{\mathit{pval}_i}^{\,i}\,) \rightsquigarrow \overline{\sigma_i}^{\,i}} \quad \textsc{Subs\_Decons\_Value\_Array}$$

$$\frac{\mathit{pattern} = \mathit{pval} \rightsquigarrow \sigma}{\mathtt{Specified}(\mathit{pattern}) = \mathit{pval} \rightsquigarrow \sigma} \quad \textsc{Subs\_Decons\_Value\_Specified}$$

$\boxed{\mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma}$

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

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

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

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

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

$$\frac{\begin{array}{c}\mathit{res\_pattern}_1 = \mathit{res\_term}_1 \leadsto \sigma_1\\ \mathit{res\_pattern}_2 = \mathit{res\_term}_2 \leadsto \sigma_2\end{array}}{\langle \mathit{res\_pattern}_1, \mathit{res\_pattern}_2 \rangle = \langle \mathit{res\_term}_1, \mathit{res\_term}_2 \rangle \leadsto \sigma_1, \sigma_2} \quad \textsc{Subs\_Decons\_Res\_Pair}$$

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

$$\frac{\mathit{res\_pattern} = \mathit{res\_term} \leadsto \sigma}{\texttt{fold}\,(\mathit{res\_pattern}) = \mathit{res\_term} \leadsto \sigma} \quad \textsc{Subs\_Decons\_Res\_Fold}$$

$$\boxed{\overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \leadsto \sigma}$$

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

$$\frac{\begin{array}{l}\mathit{ident\_or\_pattern} = \mathit{pval} \leadsto \sigma\\ \overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \leadsto \psi\end{array}}{\texttt{comp}\,\mathit{ident\_or\_pattern} = \mathit{pval}, \overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \leadsto \sigma, \psi} \quad \textsc{Subs\_Decons\_Ret\_Comp}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} abbrev_1 \equiv \mathrm{max\_int}_\tau - \mathrm{min\_int}_\tau + 1 \\ abbrev_2 \equiv pval\, \texttt{rem\_f}\; abbrev_1 \\ mem\_int' \equiv \texttt{if}\; abbrev_2 \leqslant \mathrm{max\_int}_\tau\; \texttt{then}\; abbrev_2\; \texttt{else}\; abbrev_2 - abbrev_1 \end{array}}{\langle \texttt{wrapI}\,(\tau, mem\_int)\rangle \longrightarrow \langle mem\_int'\rangle} \quad \textsc{Op\_PE\_PE\_WrapI}$$

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\langle \texttt{if}\ \texttt{False}\ \texttt{then}\ tpexpr_1\ \texttt{else}\ tpexpr_2 \rangle \longrightarrow \langle tpexpr_2 \rangle} \quad \text{OP\_TPE\_TPE\_IF\_FALSE}$$

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

$$\frac{\begin{array}{l} ident{:}arg \ \equiv\ \overline{x_i}^{\,i} \mapsto texpr \in \texttt{Globals} \\ \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret \end{array}}{\langle h; \texttt{ccall}\,(\tau, ident, \overline{spine\_elem_i}^{\,i}) \rangle \longrightarrow \langle h; \sigma(texpr){:}\sigma(ret) \rangle} \quad \text{OP\_SE\_TE\_CCALL}$$

$$\frac{\begin{array}{l} name{:}arg \ \equiv\ \overline{x_i}^{\,i} \mapsto texpr \in \texttt{Globals} \\ \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret \end{array}}{\langle h; \texttt{pcall}\,(name, \overline{spine\_elem_i}^{\,i}) \rangle \longrightarrow \langle h; \sigma(texpr){:}\sigma(ret) \rangle} \quad \text{OP\_SE\_TE\_PCALL}$$

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

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

$$\frac{\begin{array}{l} pattern_j = pval \rightsquigarrow \sigma_j \\ \forall\, i < j.\ \texttt{not}\,(pattern_i = pval \rightsquigarrow \sigma_i) \end{array}}{\langle h; \texttt{case}\, pval\, \texttt{of}\, \overline{\mid pattern_i \Rightarrow texpr_i}^{\,i}\ \texttt{end} \rangle \longrightarrow \langle h; \sigma_j(texpr_j) \rangle} \quad \text{OP\_STE\_TE\_CASE}$$

$$\frac{ident\_or\_pattern = pval \rightsquigarrow \sigma}{\langle h; \texttt{let}\, ident\_or\_pattern = pval\, \texttt{in}\, texpr \rangle \longrightarrow \langle h; \sigma(texpr) \rangle} \quad \text{OP\_STE\_TE\_LETP\_SUB}$$

$$\frac{\langle pexpr \rangle \longrightarrow \langle pexpr' \rangle}{\langle h; \texttt{let}\, ident\_or\_pattern = pexpr\, \texttt{in}\, texpr \rangle \longrightarrow \langle h; \texttt{let}\, ident\_or\_pattern = pexpr'\, \texttt{in}\, texpr \rangle} \quad \text{OP\_STE\_TE\_LETP\_LETP}$$

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

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

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

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

$$\frac{\langle h; seq\_expr\rangle \longrightarrow \langle h; texpr_1{:}ret\rangle}{\langle h; \texttt{let}\, \overline{ret\_pattern_i}^{\,i} = seq\_expr \,\texttt{in}\, texpr_2\rangle \longrightarrow \langle h; \texttt{let}\, \overline{ret\_pattern_i}^{\,i}{:}ret = texpr_1 \,\texttt{in}\, texpr_2\rangle} \quad \textsc{Op\_STE\_TE\_Let\_LetT}$$

$$\frac{\langle h; texpr_1\rangle \longrightarrow \langle h'; texpr_1'\rangle}{\langle h; \texttt{let}\, \overline{ret\_pattern_i}^{\,i}{:}ret = texpr_1 \,\texttt{in}\, texpr_2\rangle \longrightarrow \langle h'; \texttt{let}\, \overline{ret\_pattern_i}^{\,i}{:}ret = texpr_1' \,\texttt{in}\, texpr_2\rangle} \quad \textsc{Op\_STE\_TE\_LetT\_LetT}$$

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

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

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

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

$$\frac{\mathit{bool\_value} \ \equiv \ \mathit{mem\_int}_1 \ \mathit{binop}_{rel} \ \mathit{mem\_int}_2}{\langle h; \mathit{mem\_int}_1 \ \ \mathit{binop}_{rel} \ \ \mathit{mem\_int}_2 \rangle \longrightarrow \langle h; \texttt{done}\, \mathit{bool\_value} \rangle} \quad \textsc{Op\_Memop\_TVal\_Rel\_Binop}$$

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

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

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

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

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

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

$$\frac{\begin{array}{l} \texttt{fresh}\, (\mathit{mem\_ptr}) \\ \texttt{representable}\, (\tau *, \mathit{mem\_ptr}) \\ \texttt{alignedI}\, (\mathit{mem\_int}, \mathit{mem\_ptr}) \\ \mathit{pval}{:}\beta_\tau \end{array}}{\langle h; \texttt{create}\, (\mathit{mem\_int}, \tau) \rangle \longrightarrow \langle h + \{\mathit{mem\_ptr}(1) \overset{\times}{\mapsto}_\tau \mathit{pval}\}; \texttt{done}\, \mathit{mem\_ptr}, \mathit{pval}, \mathit{mem\_ptr}(1) \overset{\times}{\mapsto}_\tau \mathit{pval} \rangle} \quad \textsc{Op\_Action\_Tval\_Create}$$

$$\frac{}{\langle h + \{\mathit{mem\_ptr}(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}\}; \texttt{load}\, (\tau, \mathit{mem\_ptr}, \_, \mathit{mem\_ptr}(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}) \rangle \longrightarrow \langle h + \{\mathit{mem\_ptr}(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}\}; \texttt{done}\, \mathit{pval}, \mathit{mem\_ptr}(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval} \rangle} \quad \textsc{Op\_Action\_Tval\_Loa}$$

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

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

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

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

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

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

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

$$\frac{\overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \sigma}{\langle h; \mathtt{let}\,\mathtt{strong}\,\overline{ret\_pattern_i}^{\,i} = \mathtt{done}\,\overline{spine\_elem_i}^{\,i}\,\mathtt{in}\,texpr\rangle \longrightarrow \langle h; \sigma(texpr)\rangle} \quad \text{OP\_ISTE\_ISTE\_LETS\_SUB}$$

$$\frac{\langle h; is\_expr\rangle \longrightarrow \langle h'; is\_expr'\rangle}{\langle h; \mathtt{let}\,\mathtt{strong}\,\overline{ret\_pattern_i}^{\,i} = is\_expr\,\mathtt{in}\,texpr\rangle \longrightarrow \langle h'; \mathtt{let}\,\mathtt{strong}\,\overline{ret\_pattern_i}^{\,i} = is\_expr'\,\mathtt{in}\,texpr\rangle} \quad \text{OP\_ISTE\_ISTE\_LETS\_LETS}$$

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

$$\frac{\langle h; seq\_texpr\rangle \longrightarrow \langle h; texpr\rangle}{\langle h; seq\_texpr\rangle \longrightarrow \langle h; texpr\rangle} \quad \text{OP\_TE\_TE\_SEQ}$$

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

```
Definition rules:        231 good    0 bad
Definition rule clauses: 535 good    0 bad
```