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

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

	$\vert$	\/		
mem_int	$::=$			memory integer value
	$\vert$	1	M	
	$\vert$	0	M	
$object\_value$	$::=$			C object values (inhabitants of object types), which can be read/stored
	$\vert$	mem_int		integer value
	$\vert$	mem_ptr		pointer value
	$\vert$	$\texttt{array}\,(\,\overline{loaded\_value_i}^{\,i}\,)$		C array value
	$\vert$	$(\,\texttt{struct}\,ident)\{\,\overline{.\,member_i{:}\tau_i = mem\_val_i}^{\,i}\,\}$		C struct value
	$\vert$	$(\,\texttt{union}\,ident)\{\,.\,member = mem\_val\}$		C union value
$loaded\_value$	$::=$			potentially unspecified C object values
	$\vert$	$\texttt{specified}\,object\_value$		specified loaded value
$value$	$::=$			Core values
	$\vert$	$object\_value$		C object value
	$\vert$	$loaded\_value$		loaded C object value
	$\vert$	$\texttt{Unit}$		unit
	$\vert$	$\texttt{True}$		boolean true
	$\vert$	$\texttt{False}$		boolean false
	$\vert$	$\beta[\,\overline{value_i}^{\,i}\,]$		list
	$\vert$	$(\,\overline{value_i}^{\,i}\,)$		tuple
$bool\_value$	$::=$			Core booleans
	$\vert$	$\texttt{True}$		boolean true
	$\vert$	$\texttt{False}$		boolean false
$ctor\_val$	$::=$			data constructors
	$\vert$	$\texttt{Nil}\,\beta$		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
$ident\_opt\_\beta$	::=			type annotated optional identifier
	|	$\_{:}\beta$	binders = $\{\}$	
	|	$ident{:}\beta$	binders = $ident$	
$pattern$	::=			
	|	$ident\_opt\_\beta$	binders = binders($ident\_opt\_\beta$)	
	|	$ctor\_val(\overline{pattern_i}^{\,i})$	binders = binders($\overline{pattern_i}^{\,i}$)	
z	::=			OCaml arbitrary-width integer
	|	i	M	literal integer
	|	mem_int	M	
	|	size_of(τ)	M	size of a C type
	|	offset_of$_{tag}$($member$)	M	offset of a struct member
	|	`ptr_size`	M	size of a pointer
	|	max_int$_\tau$	M	maximum value of int of type τ
	|	min_int$_\tau$	M	minimum value of int of type τ
$\mathbb{Q}$, q, $\_$	::=			OCaml type for rational numbers
	|	$\frac{int_1}{int_2}$		
lit	::=			
	|	$ident$		
	|	`unit`		
	|	$bool$		
	|	z		
	|	$\mathbb{Q}$		

$ident\_or\_pattern$	$::=$			
	$\|$	$ident$	binders $=$ $ident$	
	$\|$	$pattern$	binders $=$ binders($pattern$)	
$array\_prop$	$::=$			array property formulas
	$\|$	$\forall\, \overline{ident_i}^{\,i}\,.\, term_1 \rightarrow term_2$	bind $\overline{ident_i}^{\,i}$ in $term_1$ bind $\overline{ident_i}^{\,i}$ in $term_2$	
$bool\_op$	$::=$			
	$\|$	$\neg\, term$		
	$\|$	$term_1 = term_2$		
	$\|$	$term_1 \leftrightarrow term_2$	M	
	$\|$	$term_1 \rightarrow term_2$		
	$\|$	$\bigwedge(\,\overline{term_i}^{\,i}\,)$		
	$\|$	$\bigvee(\,\overline{term_i}^{\,i}\,)$		
	$\|$	$array\_prop$		
	$\|$	$term_1\ binop_{bool}\ term_2$	M	
	$\|$	$\texttt{if}\ term_1\ \texttt{then}\ term_2\ \texttt{else}\ term_3$		
$arith\_op$	$::=$			
	$\|$	$term_1 + term_2$		
	$\|$	$term_1 - term_2$		
	$\|$	$term_1 \times term_2$		
	$\|$	$term_1 / term_2$		
	$\|$	$term_1\ \texttt{rem\_t}\ term_2$		
	$\|$	$term_1\ \texttt{rem\_f}\ term_2$		
	$\|$	$term_1\ \hat{}\ term_2$		
	$\|$	$term_1\ binop_{arith}\ term_2$	M	
cmp_op	$::=$			
	$\|$	$term_1 < term_2$		less than

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

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
	|	$\texttt{assert\_undef}\,(pval,\ UB\_name)$		

	$\vert$	$\texttt{bool\_to\_integer}\,(\mathit{pval})$		
	$\vert$	$\texttt{conv\_int}\,(\tau, \mathit{pval})$		
	$\vert$	$\texttt{wrapI}\,(\tau, \mathit{pval})$		
tpexpr	$::=$			top-level pure expressions
	$\vert$	tpval		top-level pure values
	$\vert$	$\texttt{case}\,\mathit{pval}\,\texttt{of}\,\overline{\vert\ \mathit{tpexpr\_case\_branch}_i}^{i}\,\texttt{end}$		pattern matching
	$\vert$	$\texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}\,\texttt{in}\,\mathit{tpexpr}$	$\textsf{bind}$ binders($\mathit{ident\_or\_pattern}$) $\textsf{in}$ tpexpr	pure let
	$\vert$	$\texttt{let}\,\mathit{ident\_or\_pattern}{:}(y_1{:}\beta_1.\ \mathit{term}_1) = \mathit{tpexpr}_1\,\texttt{in}\,\mathit{tpexpr}_2$	$\textsf{bind}$ binders($\mathit{ident\_or\_pattern}$) $\textsf{in}$ tpexpr_2 $\textsf{bind}$ y_1 $\textsf{in}$ term_1	annoted pure let
	$\vert$	$\texttt{if}\,\mathit{pval}\,\texttt{then}\,\mathit{tpexpr}_1\,\texttt{else}\,\mathit{tpexpr}_2$		pure if
	$\vert$	$\sigma(\mathit{tpexpr})$	M	simul-sub σ in tpexpr
$\mathit{tpexpr\_case\_branch}$	$::=$			pure top-level case expression
	$\vert$	$\mathit{pattern} \Rightarrow \mathit{tpexpr}$	$\textsf{bind}$ binders($\mathit{pattern}$) $\textsf{in}$ tpexpr	top-level case expression br
$\mathit{m\_kill\_kind}$	$::=$			
	$\vert$	$\texttt{dynamic}$		
	$\vert$	$\texttt{static}\,\tau$		
bool, _	$::=$			OCaml booleans
	$\vert$	$\texttt{true}$		
	$\vert$	$\texttt{false}$		
$\mathit{points\_to}$, pt	$::=$			points-to separation logic pred
	$\vert$	$\mathit{term}_1 \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2$		
$\mathit{qpoints\_to}$, qpt	$::=$			quantified (integer-indexed) p
	$\vert$	$\circledast x.\ \mathit{iguard}; \mathit{term}_1 + x \times \mathrm{size\_of}(\tau) \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2$		
$\mathit{res\_term}$	$::=$			resource terms

	|	`emp`		empty heap
	|	$points\_to$		single-cell heap
	|	$qpoints\_to$		contiguous-cell heap
	|	$ident$		variable
	|	$\langle res\_term_1, res\_term_2 \rangle$		seperating-conjunction pair
	|	$\texttt{pack}\,(pval, res\_term)$		packing for existentials
	|	$\texttt{fold}\,(res\_term)$		fold into recursive res. pred.
	|	$\texttt{explode}\,(res\_term{:}pt)$		transform points-to-array into quantified points-to
	|	$\texttt{implode}\,(res\_term{:}qpt, int)$		transform quantified points-to into points-to-array
	|	$\texttt{break}\,(res\_term{:}qpt, int)$		split a qpt into a qpt and a pt
	|	$\texttt{glue}\,(res\_term_1{:}qpt, res\_term_2{:}pt)$		join a qpt and a pt into a qpt
	|	$\sigma(res\_term)$	M	substitution for resource terms
mem_action	::=			memory actions
	|	$\texttt{create}\,(pval, \tau)$		
	|	$\texttt{create\_readonly}\,(pval_1, \tau, pval_2)$		
	|	$\texttt{alloc}\,(pval_1, pval_2)$		
	|	$\texttt{kill}\,(m\_kill\_kind, pval, pt)$		
	|	$\texttt{store}\,(bool, \tau, pval_1, pval_2, mem\_order, pt)$		true means store is locking
	|	$\texttt{load}\,(\tau, pval, mem\_order, pt)$		
	|	$\texttt{rmw}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$		
	|	$\texttt{fence}\,(mem\_order)$		
	|	$\texttt{cmp\_exch\_strong}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$		
	|	$\texttt{cmp\_exch\_weak}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$		
	|	$\texttt{linux\_fence}\,(linux\_mem\_order)$		
	|	$\texttt{linux\_load}\,(\tau, pval, linux\_mem\_order)$		
	|	$\texttt{linux\_store}\,(\tau, pval_1, pval_2, linux\_mem\_order)$		
	|	$\texttt{linux\_rmw}\,(\tau, pval_1, pval_2, linux\_mem\_order)$		
$polarity$	::=			polarities for memory actions
	|			(pos) sequenced by `let weak` and `let strong`

	$\vert$	`neg`		only sequenced by `let strong`
pol_mem_action	::=			memory actions with polarity
	$\vert$	*polarity mem_action*		
mem_op	::=			operations involving the memory state
	$\vert$	$pval_1 \ binop_{rel} \ pval_2$		pointer relational binary operations
	$\vert$	$pval_1 -_{\tau} pval_2$		pointer subtraction
	$\vert$	`intFromPtr` $(\tau_1, \tau_2, pval)$		cast of pointer value to integer value
	$\vert$	`ptrFromInt` $(\tau_1, \tau_2, pval)$		cast of integer value to pointer value
	$\vert$	`ptrValidForDeref` $(\tau, pval, pt)$		dereferencing validity predicate
	$\vert$	`ptrWellAligned` $(\tau, pval)$		
	$\vert$	`ptrArrayShift` $(pval_1, \tau, pval_2)$		
	$\vert$	`memcpy` $(pval_1, pval_2, pval_3)$		
	$\vert$	`memcmp` $(pval_1, pval_2, pval_3)$		
	$\vert$	`realloc` $(pval_1, pval_2, pval_3)$		
	$\vert$	`va_start` $(pval_1, pval_2)$		
	$\vert$	`va_copy` $(pval)$		
	$\vert$	`va_arg` $(pval, \tau)$		
	$\vert$	`va_end` $(pval)$		
spine_elem	::=			spine element
	$\vert$	*pval*		pure or logical value
	$\vert$	*res_term*		resource value
	$\vert$	$\sigma(spine\_elem)$	M	substitution for spine elements / return values
spine	::=			spine
	$\vert$	$\overline{spine\_elem_i}^{\,i}$		
tval	::=			(effectful) top-level values
	$\vert$	`done` *spine*		end of top-level expression

	$\mid$	**undef** UB_name		undefined behaviour
$res\_pattern$	$::=$			resource terms
	$\mid$	**emp**	binders $= \{\}$	empty heap
	$\mid$	$ident$	binders $= ident$	variable
	$\mid$	**fold** $(res\_pattern)$	binders $= \{\}$	unfold (recursive) predicate
	$\mid$	$\langle res\_pattern_1, res\_pattern_2 \rangle$	binders $=$ binders$(res_pattern_1) \cup$ binders$(res_pattern_2)$	seperating-conjunction pair
	$\mid$	**pack** $(ident, res\_pattern)$	binders $= ident \cup$ binders$(res_pattern)$	packing for existentials
$ret\_pattern$	$::=$			return pattern
	$\mid$	**comp** $ident\_or\_pattern$	binders $=$ binders$(ident_or_pattern)$	computational variable
	$\mid$	**log** $ident$	binders $= ident$	logical variable
	$\mid$	**res** $res\_pattern$	binders $=$ binders$(res_pattern)$	resource variable
$init$,	$::=$			initialisation status
	$\mid$	$\checkmark$		initialised
	$\mid$	$\times$		uninitalised
res	$::=$			resources
	$\mid$	**emp**		empty heap
	$\mid$	$points\_to$		points-to heap pred.
	$\mid$	$qpoints\_to$		quantified (integer-indexed) points-to heap pred.
	$\mid$	$res_1 * res_2$		seperating conjunction
	$\mid$	$\exists\, ident{:}\beta.\ res$		existential
	$\mid$	$term \wedge res$		logical conjuction
	$\mid$	**if** $term$ **then** res_1 **else** res_2		ordered disjuction
	$\mid$	$\alpha(\overline{pval_i}^{\,i})$		predicate
	$\mid$	$\sigma(res)$	M	simul-sub σ in res
ret, _	$::=$			return types
	$\mid$	$\Sigma\, ident{:}\beta.\ ret$		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

	$\mid$ $\mathtt{let}\,\mathtt{weak}\,\overline{ret\_pattern_i}^{\,i} = is\_expr\,\mathtt{in}\,texpr$	bind binders$(\overline{ret_pattern_i}^{\,i})$ in $texpr$	weak sequencing
	$\mid$ $\mathtt{let}\,\mathtt{strong}\,\overline{ret\_pattern_i}^{\,i} = is\_expr\,\mathtt{in}\,texpr$	bind binders$(\overline{ret_pattern_i}^{\,i})$ in $texpr$	strong sequencing
$texpr$	$::=$		top-level (effectful) expressions
	$\mid$ seq_texpr		sequential (effectful) expressions
	$\mid$ is_texpr		indet seq (effectful) expressions
	$\mid$ $\sigma(texpr)$	M	simul-sub σ in $texpr$
arg	$::=$		argument/function types
	$\mid$ $\Pi\, ident{:}\beta.\ arg$		
	$\mid$ $\forall\, ident{:}\beta.\ arg$		
	$\mid$ $res \multimap arg$		
	$\mid$ $term \supset arg$		
	$\mid$ ret		
	$\mid$ $\sigma(arg)$	M	simul-sub σ in arg
$pure\_arg$	$::=$		pure argument/function types
	$\mid$ $\Pi\, ident{:}\beta.\ pure\_arg$		
	$\mid$ $term \supset pure\_arg$		
	$\mid$ $pure\_ret$		
$pure\_ret$	$::=$		pure return types
	$\mid$ $\Sigma\, ident{:}\beta.\ pure\_ret$		
	$\mid$ $term \wedge pure\_ret$		
	$\mid$ $\mathtt{I}$		
$\mathcal{C}$	$::=$		computational var env
	$\mid$ $\cdot$		
	$\mid$ $\mathcal{C}, ident{:}\beta$		
	$\mid$ $\overline{\mathcal{C}_i}^{\,i}$		

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

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

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

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

$$
\begin{array}{lll}
 & | & \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret}
\end{array}
$$

$$
\begin{array}{lll}
\mathit{pure\_opsem\_jtype} & ::= & \\
 & | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{pexpr}' \rangle \\
 & | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\, \mathit{term}) \rangle \\
 & | & \langle \mathit{tpexpr} \rangle \longrightarrow \langle \mathit{tpexpr}' \rangle
\end{array}
$$

$$
\begin{array}{lll}
\mathit{opsem\_jtype} & ::= & \\
 & | & \langle h; \mathit{seq\_expr} \rangle \longrightarrow \langle h'; \mathit{texpr}{:}\mathit{ret} \rangle \\
 & | & \langle h; \mathit{seq\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
 & | & \langle h; \mathit{mem\_op} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
 & | & \langle h; \mathit{mem\_action} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
 & | & \langle h; \mathit{is\_expr} \rangle \longrightarrow \langle h'; \mathit{is\_expr}' \rangle \\
 & | & \langle h; \mathit{is\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
 & | & \langle h; \mathit{texpr} \rangle \longrightarrow \langle h'; \mathit{texpr}' \rangle
\end{array}
$$

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

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

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

$$
\frac{\begin{array}{c}\mathcal{C}; \mathcal{L} \vdash \ \mathtt{guarded\_e}\,(\mathit{term}_1) \\ \mathcal{C}; \mathcal{L} \vdash \ \mathtt{guarded\_e}\,(\mathit{term}_2)\end{array}}{\mathcal{C}; \mathcal{L} \vdash \ \mathtt{guarded\_e}\,(\mathit{term}_1 + \mathit{term}_2)} \quad \textsc{Wf\_Guarded\_Eexpr\_Plus}
$$

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

$$\frac{\mathcal{C};\mathcal{L} \vdash \texttt{guarded\_e}\,(\mathit{term})}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term})} \quad \textsc{Wf\_Guarded\_Expr\_Eexpr}$$

$$\frac{\mathit{ident} \in \overline{\mathit{ident}_i}^{\,i}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{ident})} \quad \textsc{Wf\_Guarded\_Expr\_UVar}$$

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term}) \\ \mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term}')\end{array}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term} \leqslant \mathit{term}')} \quad \textsc{Wf\_Guarded\_Leq}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term}) \\ \mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term}')\end{array}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term} = \mathit{term}')} \quad \textsc{Wf\_Guarded\_Eq}$$

$$\frac{\overline{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term}_j)}^{\,j}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\bigvee(\overline{\mathit{term}_j}^{\,j}))} \quad \textsc{Wf\_Guarded\_Or}$$

$$\frac{\overline{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term}_j)}^{\,j}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\bigwedge(\overline{\mathit{term}_j}^{\,j}))} \quad \textsc{Wf\_Guarded\_And}$$

$$\frac{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{term})}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\neg\,\mathit{term})} \quad \textsc{Wf\_Guarded\_Neg}$$

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\cdot; \cdot; \cdot; \cdot \sqsubseteq \cdot; \cdot; \cdot; \cdot} \quad \text{WEAK\_EMPTY}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}, x{:}\beta; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}', x{:}\beta; \mathcal{L}'; \Phi'; \mathcal{R}'} \quad \text{WEAK\_CONS\_COMP}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}, x{:}\beta; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}', x{:}\beta; \Phi'; \mathcal{R}'} \quad \text{WEAK\_CONS\_LOG}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi, term; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi', term; \mathcal{R}'} \quad \text{WEAK\_CONS\_PHI}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}, x{:}res \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}', x{:}res} \quad \text{WEAK\_CONS\_RES}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}', x{:}\beta; \mathcal{L}'; \Phi'; \mathcal{R}'} \quad \text{WEAK\_SKIP\_COMP}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}', x{:}\beta; \Phi'; \mathcal{R}'} \quad \text{WEAK\_SKIP\_LOG}$$

$$\frac{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi', term; \mathcal{R}'} \quad \text{WEAK\_SKIP\_PHI}$$

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

$$\frac{}{\mathcal{C}; \mathcal{L}; \Phi; \cdot \vdash (\cdot){:}(\cdot; \cdot; \cdot; \cdot)} \quad \text{TY\_SUBS\_EMPTY}$$

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow term_1 \leftrightarrow term_2)\\ \Phi \vdash res_{11} \;\equiv\; res_{21}\\ \Phi \vdash res_{21} \;\equiv\; res_{22}\end{array}}{\Phi \vdash \texttt{if}\, term_1\, \texttt{then}\, res_{11}\, \texttt{else}\, res_{12} \;\equiv\; \texttt{if}\, term_2\, \texttt{then}\, res_{21}\, \texttt{else}\, res_{22}} \quad \textsc{Ty\_Res\_Eq\_OrdDisj}$$

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

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

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

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

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

$$\frac{\begin{array}{l}\Phi \vdash res_1' = \texttt{strip\_ifs}\,(res_1)\\ \Phi \vdash res_2' = \texttt{strip\_ifs}\,(res_2)\\ \Phi \vdash res_1' \;\equiv\; res_2'\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot, r{:}res_1 \vdash r \Leftarrow res_2} \quad \textsc{Ty\_Res\_Var}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\overline{\mathit{pattern}_i{:}\beta \rightsquigarrow \mathcal{C}_i\,\mathtt{with}\,\mathit{term}_i}^{i}}{\mathtt{Array}(\,\overline{\mathit{pattern}_i}^{\,i}\,){:}\mathtt{array}\,\beta \rightsquigarrow \overline{\mathcal{C}_i}^{\,i}\,\mathtt{with}\,[|\ \overline{\mathit{term}_i}^{\,i}\,|]}\ \textsc{Ty\_Pat\_Comp\_Array}$$

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

$\boxed{\mathit{ident\_or\_pattern}{:}\beta \rightsquigarrow \mathcal{C}\,\mathtt{with}\,\mathit{term}}$

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

$$\frac{\mathit{pattern}{:}\beta \rightsquigarrow \mathcal{C}\,\mathtt{with}\,\mathit{term}}{\mathit{pattern}{:}\beta \rightsquigarrow \mathcal{C}\,\mathtt{with}\,\mathit{term}}\ \textsc{Ty\_Pat\_Sym\_Or\_Pattern\_Pattern}$$

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

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

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

$$\frac{}{\Phi \vdash \mathit{qpt} = \mathtt{strip\_ifs}\,(\mathit{qpt})}\ \textsc{Ty\_Pat\_Res\_StripIfs\_QpointsTo}$$

$$\frac{}{\Phi \vdash \mathit{res}_1 * \mathit{res}_2 = \mathtt{strip\_ifs}\,(\mathit{res}_1 * \mathit{res}_2)}\ \textsc{Ty\_Pat\_Res\_StripIfs\_SepConj}$$

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

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

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow term) \\ \Phi \vdash res_1' = \texttt{strip\_ifs}\,(res_1')\end{array}}{\Phi \vdash res_1' = \texttt{strip\_ifs}\,(\texttt{if}\,term\,\texttt{then}\,res_1\,\texttt{else}\,res_2)} \quad \textsc{Ty\_Pat\_Res\_StripIfs\_True}$$

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow \neg\, term) \\ \Phi \vdash res_2' = \texttt{strip\_ifs}\,(res_2)\end{array}}{\Phi \vdash res_2' = \texttt{strip\_ifs}\,(\texttt{if}\,term\,\texttt{then}\,res_1\,\texttt{else}\,res_2)} \quad \textsc{Ty\_Pat\_Res\_StripIfs\_False}$$

$$\frac{}{\Phi \vdash \texttt{if}\,term\,\texttt{then}\,res_1\,\texttt{else}\,res_2 = \texttt{strip\_ifs}\,(\texttt{if}\,term\,\texttt{then}\,res_1\,\texttt{else}\,res_2)} \quad \textsc{Ty\_Pat\_Res\_StripIfs\_UnderDet}$$

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

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

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

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

$$\frac{\begin{array}{c}\Phi \vdash res\_pattern_1{:}res_1 \rightsquigarrow \mathcal{L}_1; \Phi_1; \mathcal{R}_1 \\ \Phi \vdash res\_pattern_2{:}res_2 \rightsquigarrow \mathcal{L}_2; \Phi_2; \mathcal{R}_2\end{array}}{\Phi \vdash res_1 * res_2\ \mathsf{as}\ \langle res\_pattern_1, res\_pattern_2\rangle \rightsquigarrow \mathcal{L}_1, \mathcal{L}_2; \Phi_1, \Phi_2; \mathcal{R}_1, \mathcal{R}_2} \quad \textsc{Ty\_Pat\_Res\_Match\_SepConj}$$

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

$$\frac{\Phi \vdash \mathit{res\_pattern}{:}x/y, \cdot(\mathit{res}) \rightsquigarrow \mathcal{L}'; \Phi'; \mathcal{R}'}{\Phi \vdash \exists\, y{:}\beta.\ \mathit{res}\ \mathtt{as}\ \mathtt{pack}\,(x, \mathit{res\_pattern}) \rightsquigarrow \mathcal{L}', x{:}\beta; \Phi'; \mathcal{R}'} \quad \text{TY\_PAT\_RES\_MATCH\_PACK}$$

$$\frac{\begin{array}{l}\alpha\ \equiv\ \overline{x_i{:}\beta_i}^{\,i}\ \mapsto \mathit{res}\ \in\ \mathtt{Globals}\\ \Phi \vdash \mathit{res\_pattern}{:}\overline{\mathit{pval}_i/x_i, \cdot}^{\,i}(\mathit{res}) \rightsquigarrow \mathcal{L}'; \Phi'; \mathcal{R}'\end{array}}{\Phi \vdash \alpha(\,\overline{\mathit{pval}_i}^{\,i}\,)\ \mathtt{as}\ \mathtt{fold}\,(\mathit{res\_pattern}) \rightsquigarrow \mathcal{L}'; \Phi'; \mathcal{R}'} \quad \text{TY\_PAT\_RES\_MATCH\_FOLD}$$

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

$$\frac{\begin{array}{l}\Phi \vdash \mathit{res}' = \mathtt{strip\_ifs}\,(\mathit{res})\\ \Phi \vdash \mathit{res}'\ \mathtt{as}\ \mathit{res\_pattern} \rightsquigarrow \mathcal{L}'; \Phi'; \mathcal{R}'\end{array}}{\Phi \vdash \mathit{res\_pattern}{:}\mathit{res} \rightsquigarrow \mathcal{L}'; \Phi'; \mathcal{R}'} \quad \text{TY\_PAT\_RES\_STRIP\_IFS}$$

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

$$\frac{}{\Phi \vdash\ {:}\mathtt{I} \rightsquigarrow \cdot; \cdot; \cdot; \cdot} \quad \text{TY\_PAT\_RET\_EMPTY}$$

$$\frac{\begin{array}{l}\mathit{ident\_or\_pattern}{:}\beta \rightsquigarrow \mathcal{C}_1\ \mathtt{with}\ \mathit{term}_1\\ \Phi \vdash\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{term}_1/y, \cdot(\mathit{ret}) \rightsquigarrow \mathcal{C}_2; \mathcal{L}_2; \Phi_2; \mathcal{R}_2\end{array}}{\Phi \vdash \mathtt{comp}\ \mathit{ident\_or\_pattern},\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\Sigma\, y{:}\beta.\ \mathit{ret} \rightsquigarrow \mathcal{C}_1, \mathcal{C}_2; \mathcal{L}_2; \Phi_2; \mathcal{R}_2} \quad \text{TY\_PAT\_RET\_COMP}$$

$$\frac{\Phi \vdash\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} \rightsquigarrow \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\Phi \vdash \mathtt{log}\ y,\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\exists\, y{:}\beta.\ \mathit{ret} \rightsquigarrow \mathcal{C}'; \mathcal{L}', y{:}\beta; \Phi'; \mathcal{R}'} \quad \text{TY\_PAT\_RET\_LOG}$$

$$\frac{\begin{array}{l}\Phi \vdash \mathit{res\_pattern}{:}\mathit{res} \rightsquigarrow \mathcal{L}_1; \Phi_1; \mathcal{R}_1\\ \Phi \vdash\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} \rightsquigarrow \mathcal{C}_2; \mathcal{L}_2; \Phi_2; \mathcal{R}_2\end{array}}{\Phi \vdash \mathtt{res}\ \mathit{res\_pattern},\ \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{res} \otimes \mathit{ret} \rightsquigarrow \mathcal{C}_2; \mathcal{L}_1, \mathcal{L}_2; \Phi_1, \Phi_2; \mathcal{R}_1, \mathcal{R}_2} \quad \text{TY\_PAT\_RET\_RES}$$

$$\frac{\Phi \vdash \overline{\mathit{ret\_pattern}_i}^{\,i} : \mathit{ret} \rightsquigarrow \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'}{\Phi \vdash \overline{\mathit{ret\_pattern}_i}^{\,i} : \mathit{term} \wedge \mathit{ret} \rightsquigarrow \mathcal{C}'; \mathcal{L}'; \Phi', \mathit{term}; \mathcal{R}'} \quad \text{TY\_PAT\_RET\_PHI}$$

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

$$\frac{\texttt{smt}\,(\Phi \Rightarrow \texttt{false})}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{undef}\ \mathit{UB\_name} \Leftarrow y{:}\beta.\, \mathit{term}} \quad \text{TY\_TPVAL\_UNDEF}$$

$$\frac{\begin{array}{c}\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta \\ \texttt{smt}\,(\Phi \Rightarrow \mathit{pval}/y, \cdot(\mathit{term}))\end{array}}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{done}\,\mathit{pval} \Leftarrow y{:}\beta.\, \mathit{term}} \quad \text{TY\_TPVAL\_DONE}$$

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

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

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

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

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

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

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \mathtt{integer}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \mathtt{create}\,(\mathit{pval},\tau) \Rightarrow \Sigma\, y_p{:}\mathtt{loc}.\ \mathtt{representable}\,(\tau *, y_p) \wedge \mathtt{alignedI}\,(\mathit{pval}, y_p) \wedge \exists\, y{:}\beta_\tau.\ y_p(1) \overset{\times}{\mapsto}_\tau y \otimes \mathtt{I}}\quad \text{TY\_ACTION\_CREATE}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_0 \Rightarrow \mathtt{loc}\\ \mathtt{smt}\,(\Phi \Rightarrow \mathit{pval}_0 = \mathit{pval}_1)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{pval}_1(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2 \Leftarrow \mathit{pval}_1(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{load}\,(\tau, \mathit{pval}_0, \_, \mathit{pval}_1(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2) \Rightarrow \Sigma\, y{:}\beta_\tau.\ y = \mathit{pval}_2 \wedge \mathit{pval}_1(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}_2 \otimes \mathtt{I}}\quad \text{TY\_ACTION\_LOAD}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_0 \Rightarrow \mathtt{loc}\\ \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \beta_\tau\\ \mathtt{smt}\,(\Phi \Rightarrow \mathtt{representable}\,(\tau, \mathit{pval}_1))\\ \mathtt{smt}\,(\Phi \Rightarrow \mathit{pval}_2 = \mathit{pval}_0)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{pval}_2(1) \mapsto_\tau \_ \Leftarrow \mathit{pval}_2(1) \mapsto_\tau \_\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{store}\,(\_, \tau, \mathit{pval}_0, \mathit{pval}_1, \_, \mathit{pval}_2(1) \mapsto_\tau \_) \Rightarrow \Sigma\, \_{:}\mathtt{unit}.\ \mathit{pval}_2(1) \overset{\checkmark}{\mapsto}_\tau \mathit{pval}_1 \otimes \mathtt{I}}\quad \text{TY\_ACTION\_STORE}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_0 \Rightarrow \mathtt{loc}\\ \mathtt{smt}\,(\Phi \Rightarrow \mathit{pval}_0 = \mathit{pval}_1)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{pval}_1(1) \mapsto_\tau \_ \Leftarrow \mathit{pval}_1(1) \mapsto_\tau \_\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{kill}\,(\mathtt{static}\,\tau, \mathit{pval}_0, \mathit{pval}_1(1) \mapsto_\tau \_) \Rightarrow \Sigma\, \_{:}\mathtt{unit}.\ \mathtt{I}}\quad \text{TY\_ACTION\_KILL\_STATIC}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\langle \mathit{res\_pattern}_1, \mathit{res\_pattern}_2 \rangle = \texttt{break}\,(\mathit{res\_term}{:}\mathit{qpt}, i) \leadsto \sigma_1, \sigma_2} \quad \textsc{Subs\_Decons\_Res\_Break\_Doesnt\_Work}$$

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

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

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

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

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

$$\frac{\overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \rightsquigarrow \psi}{\texttt{log}\, \mathit{ident} = \mathit{pval},\ \overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \rightsquigarrow \mathit{pval}/\mathit{ident}, \psi}\ \textsc{Subs\_Decons\_Ret\_Log}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\langle pexpr\rangle \longrightarrow \langle tpexpr_1{:}(y{:}\beta.\ term)\rangle}{\langle \texttt{let}\ ident\_or\_pattern = pexpr\ \texttt{in}\ tpexpr_2\rangle \longrightarrow \langle \texttt{let}\ ident\_or\_pattern{:}(y{:}\beta.\ term) = tpexpr_1\ \texttt{in}\ tpexpr_2\rangle}\ \textsc{Op\_TPE\_TPE\_Let\_LetT}$$

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

$$\frac{\langle tpexpr_1\rangle \longrightarrow \langle tpexpr_1'\rangle}{\langle \texttt{let}\ ident\_or\_pattern{:}(y{:}\beta.\ term) = tpexpr_1\ \texttt{in}\ tpexpr_2\rangle \longrightarrow \langle \texttt{let}\ ident\_or\_pattern{:}(y{:}\beta.\ term) = tpexpr_1'\ \texttt{in}\ tpexpr_2\rangle}\ \textsc{Op\_TPE\_TPE\_LetT\_LetT}$$

$$\frac{}{\langle \texttt{if True then}\ tpexpr_1\ \texttt{else}\ tpexpr_2\rangle \longrightarrow \langle tpexpr_1\rangle}\ \textsc{Op\_TPE\_TPE\_If\_True}$$

$$\frac{}{\langle \texttt{if False then}\ tpexpr_1\ \texttt{else}\ tpexpr_2\rangle \longrightarrow \langle tpexpr_2\rangle}\ \textsc{Op\_TPE\_TPE\_If\_False}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\langle h + \{\mathit{mem\_ptr}(\_) \mapsto_\tau \_\}; \texttt{kill}\,(\texttt{static}\,\tau, \mathit{mem\_ptr}, \mathit{mem\_ptr}(\_) \mapsto_\tau \_)\rangle \longrightarrow \langle h; \texttt{done}\,\texttt{Unit}\rangle} \quad \textsc{Op\_Action\_Tval\_Kill\_Static}$$

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

$$\frac{\langle h; \mathit{mem\_op}\rangle \longrightarrow \langle h; \mathit{tval}\rangle}{\langle h; \texttt{memop}\,(\mathit{mem\_op})\rangle \longrightarrow \langle h; \mathit{tval}\rangle} \quad \textsc{Op\_IsE\_IsE\_Memop}$$

$$\frac{\langle h; \mathit{mem\_action}\rangle \longrightarrow \langle h'; \mathit{tval}\rangle}{\langle h;\ \mathit{mem\_action}\rangle \longrightarrow \langle h'; \mathit{tval}\rangle} \quad \textsc{Op\_IsE\_IsE\_Action}$$

$$\frac{\langle h; \mathit{mem\_action}\rangle \longrightarrow \langle h'; \mathit{tval}\rangle}{\langle h; \texttt{neg}\,\mathit{mem\_action}\rangle \longrightarrow \langle h'; \mathit{tval}\rangle} \quad \textsc{Op\_IsE\_IsE\_Neg\_Action}$$

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

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

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

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

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

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

```
Definition rules:        233 good    0 bad
Definition rule clauses: 538 good    0 bad
```