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, τ	::=			C type
	|	$\tau *$		pointer to type τ
tag	::=			OCaml type for struct/union tag
	|	*ident*		
β, _	::=			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
	|	$\beta \to \beta'$		parameter types
	|	β_τ	M	of a C type
binop	::=			binary operators
	|	`+`		addition
	|	`-`		subtraction
	|	`*`		multiplication
	|	`/`		division
	|	`rem_t`		modulus
	|	`rem_f`		remainder
	|	`^`		exponentiation
	|	`=`		equality, defined both for integer and C types
	|	`!=`		inequality, similiarly defined

	$\mid$	`>`	greater than, similarly defined
	$\mid$	`<`	less than, similarly defined
	$\mid$	`>=`	greater than or equal to, similarly defined
	$\mid$	`<=`	less than or equal to, similarly defined
	$\mid$	`/\`	conjucntion
	$\mid$	`\/`	disjunction
$binop_{arith}$	::=		arithmentic binary operators
	$\mid$	`+`	
	$\mid$	`-`	
	$\mid$	`*`	
	$\mid$	`/`	
	$\mid$	`rem_t`	
	$\mid$	`rem_f`	
	$\mid$	`^`	
$binop_{rel}$	::=		relational binary operators
	$\mid$	`=`	
	$\mid$	`!=`	
	$\mid$	`>`	
	$\mid$	`<`	
	$\mid$	`>=`	
	$\mid$	`<=`	
$binop_{bool}$	::=		boolean binary operators
	$\mid$	`/\`	
	$\mid$	`\/`	
mem_int	::=		memory integer value
	$\mid$	1 M	
	$\mid$	0 M	

object_value	::=		C object values (inhabitants of object types), which can be read/stored
	|	*mem_int*	integer value
	|	*mem_ptr*	pointer value
	|	$\texttt{array}\,(\,\overline{\mathit{loaded\_value}_i}^{\,i}\,)$	C array value
	|	$(\,\texttt{struct}\,\mathit{ident})\{\,\overline{.\,\mathit{member}_i{:}\tau_i = \mathit{mem\_val}_i}^{\,i}\,\}$	C struct value
	|	$(\,\texttt{union}\,\mathit{ident})\{\,.\,\mathit{member} = \mathit{mem\_val}\}$	C union value
loaded_value	::=		potentially unspecified C object values
	|	$\texttt{specified}\,\mathit{object\_value}$	specified loaded value
value	::=		Core values
	|	*object_value*	C object value
	|	*loaded_value*	loaded C object value
	|	`Unit`	unit
	|	`True`	boolean true
	|	`False`	boolean false
	|	$\beta[\,\overline{\mathit{value}_i}^{\,i}\,]$	list
	|	$(\,\overline{\mathit{value}_i}^{\,i}\,)$	tuple
bool_value	::=		Core booleans
	|	`True`	boolean true
	|	`False`	boolean false
ctor_val	::=		data constructors
	|	$\texttt{Nil}\,\beta$	empty list
	|	`Cons`	list cons
	|	`Tuple`	tuple
	|	`Array`	C array
	|	`Specified`	non-unspecified loaded value
ctor_expr	::=		data constructors

	$\mid$	`Ivmax`		max integer value
	$\mid$	`Ivmin`		min integer value
	$\mid$	`Ivsizeof`		sizeof value
	$\mid$	`Ivalignof`		alignof value
	$\mid$	`IvCOMPL`		bitwise complement
	$\mid$	`IvAND`		bitwise AND
	$\mid$	`IvOR`		bitwise OR
	$\mid$	`IvXOR`		bitwise XOR
	$\mid$	`Fvfromint`		cast integer to floating value
	$\mid$	`Ivfromfloat`		cast floating to integer value
$name$	$::=$			
	$\mid$	$ident$		Core identifier
	$\mid$	$impl\_const$		implementation-defined constant
$pval$	$::=$			pure values
	$\mid$	$ident$		Core identifier
	$\mid$	$impl\_const$		implementation-defined constant
	$\mid$	$value$		Core values
	$\mid$	$\texttt{constrained}\,(\overline{mem\_iv\_c_i, pval_i}^{\,i})$		constrained value
	$\mid$	$\texttt{error}\,(string, pval)$		impl-defined static error
	$\mid$	$ctor\_val(\overline{pval_i}^{\,i})$		data constructor application
	$\mid$	$(\texttt{struct}\,ident)\{\overline{.member_i = pval_i}^{\,i}\}$		C struct expression
	$\mid$	$(\texttt{union}\,ident)\{.member = pval\}$		C union expression
$tpval$	$::=$			top-level pure values
	$\mid$	$\texttt{undef}\ UB\_name$		undefined behaviour
	$\mid$	$\texttt{done}\,pval$		pure done
$ident\_opt\_\beta$	$::=$			type annotated optional identifier
	$\mid$	$\_{:}\beta$	binders $= \{\}$	

	$\vert$	$ident{:}\beta$	binders = $ident$	
$pattern$	::=			
	$\vert$	$ident\_opt\_\beta$	binders = binders($ident\_opt\_\beta$)	
	$\vert$	$ctor\_val(\,\overline{pattern_i}^{\,i}\,)$	binders = binders($\overline{pattern_i}^{\,i}$)	
z	::=			OCaml arbitrary-width integer
	$\vert$	i	M	literal integer
	$\vert$	mem_int	M	
	$\vert$	size_of(τ)	M	size of a C type
	$\vert$	offset_of$_{tag}$($member$)	M	offset of a struct member
	$\vert$	`ptr_size`	M	size of a pointer
	$\vert$	max_int$_\tau$	M	maximum value of int of type τ
	$\vert$	min_int$_\tau$	M	minimum value of int of type τ
$\mathbb{Q}$, q, $\_$	::=			OCaml type for rational numbers
	$\vert$	$\frac{int_1}{int_2}$		
lit	::=			
	$\vert$	$ident$		
	$\vert$	`unit`		
	$\vert$	$bool$		
	$\vert$	z		
	$\vert$	$\mathbb{Q}$		
$ident\_or\_pattern$	::=			
	$\vert$	$ident$	binders = $ident$	
	$\vert$	$pattern$	binders = binders($pattern$)	
$array\_prop\_form$	::=			array property formulas
	$\vert$	$\forall\,\overline{ident_i}^{\,i}\,.\,form_1 \rightarrow form_2$		

form	$::=$		
	$\mid$	$\bigwedge(\overline{\mathit{form}_i}^{\,i})$	
	$\mid$	$\bigvee(\overline{\mathit{form}_i}^{\,i})$	
	$\mid$	$\mathit{expr\_form} \leq \mathit{expr\_form}'$	
	$\mid$	$\mathit{expr\_form} = \mathit{expr\_form}'$	
$\mathit{expr\_form}$	$::=$		
	$\mid$	ident	
	$\mid$	z	
	$\mid$	$\mathit{expr\_form}_1 \times \mathit{expr\_form}_2$	
	$\mid$	$\mathit{expr\_form}_1 + \mathit{expr\_form}_2$	
	$\mid$	$\mathit{expr\_form}_1[\mathit{expr\_form}_2]$	
$\mathit{bool\_op}$	$::=$		
	$\mid$	$\neg\,\mathit{term}$	
	$\mid$	$\mathit{term}_1 = \mathit{term}_2$	
	$\mid$	$\mathit{term}_1 \rightarrow \mathit{term}_2$	
	$\mid$	$\bigwedge(\overline{\mathit{term}_i}^{\,i})$	
	$\mid$	$\bigvee(\overline{\mathit{term}_i}^{\,i})$	
	$\mid$	$\mathit{term}_1\ \mathit{binop}_{\mathit{bool}}\ \mathit{term}_2$	M
	$\mid$	$\texttt{if}\ \mathit{term}_1\ \texttt{then}\ \mathit{term}_2\ \texttt{else}\ \mathit{term}_3$	
$\mathit{arith\_op}$	$::=$		
	$\mid$	$\mathit{term}_1 + \mathit{term}_2$	
	$\mid$	$\mathit{term}_1 - \mathit{term}_2$	
	$\mid$	$\mathit{term}_1 \times \mathit{term}_2$	
	$\mid$	$\mathit{term}_1 / \mathit{term}_2$	
	$\mid$	$\mathit{term}_1\ \texttt{rem\_t}\ \mathit{term}_2$	
	$\mid$	$\mathit{term}_1\ \texttt{rem\_f}\ \mathit{term}_2$	
	$\mid$	$\mathit{term}_1\ \texttt{\^{}}\ \mathit{term}_2$	
	$\mid$	$\mathit{term}_1\ \mathit{binop}_{\mathit{arith}}\ \mathit{term}_2$	M

cmp_op ::=
| $term_1 < term_2$ less than
| $term_1 \leq term_2$ less than or equal
| $term_1 \ binop_{rel} \ term_2$ M

$list\_op$ ::=
| `nil`
| $term_1 :: term_2$
| `tl` $term$
| $term^{(int)}$

$tuple\_op$ ::=
| $(\ \overline{term_i}^{\,i}\ )$
| $term^{(int)}$

$pointer\_op$ ::=
| mem_ptr
| $term_1 +_{\mathrm{ptr}} term_2$
| `cast_int_to_ptr` $term$
| `cast_ptr_to_int` $term$

$array\_op$ ::=
| $[|\ \overline{term_i}^{\,i}\ |]$
| $term_1[term_2]$

$param\_op$ ::=
| $ident{:}\beta.\ term$
| $term(term_1, \ldots, term_n)$

$struct\_op$ ::=
| $term.member$

ct_pred	::=			
	|	$\texttt{representable}\,(\tau, term)$		
	|	$\texttt{aligned}\,(\tau, term)$		
	|	$\texttt{alignedI}\,(term_1, term_2)$		
$term,\ \_$	::=			
	|	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
	|	$\texttt{assert\_undef}\,(pval,\ UB\_name)$		

	|	`bool_to_integer` $(pval)$		
	|	`conv_int` $(\tau, pval)$		
	|	`wrapI` $(\tau, pval)$		
$tpexpr$	::=			top-level pure expressions
	|	$tpval$		top-level pure values
	|	`case` $pval$ `of` $\overline{\mid tpexpr\_case\_branch_i}^{i}$ `end`		pattern matching
	|	`let` $ident\_or\_pattern = pexpr$ `in` $tpexpr$	bind binders($ident\_or\_pattern$) in $tpexpr$	pure let
	|	`let` $ident\_or\_pattern{:}(y_1{:}\beta_1.\ term_1) = tpexpr_1$ `in` $tpexpr_2$	bind binders($ident\_or\_pattern$) in $tpexpr_2$ bind y_1 in $term_1$	annoted pure let
	|	`if` $pval$ `then` $tpexpr_1$ `else` $tpexpr_2$		pure if
	|	$\sigma(tpexpr)$	M	simul-sub σ in $tpexpr$
$tpexpr\_case\_branch$	::=			pure top-level case expression
	|	$pattern \Rightarrow tpexpr$	bind binders($pattern$) in $tpexpr$	top-level case expression br
m_kill_kind	::=			
	|	`dynamic`		
	|	`static` τ		
$bool$, _	::=			OCaml booleans
	|	`true`		
	|	`false`		
int, _	::=			OCaml fixed-width integer
	|	i		literal integer
res_term	::=			resource terms
	|	`emp`		empty heap
	|	$points\_to$		single-cell heap
	|	$ident$		variable

	$\vert$	$\langle res\_term_1, res\_term_2 \rangle$		seperating-conjunction pair
	$\vert$	$\texttt{pack}\,(pval, res\_term)$		packing for existentials
	$\vert$	$\texttt{fold}\,(res\_term)$		fold into recursive res. pred.
	$\vert$	$\sigma(res\_term)$	M	substitution for resource terms
mem_action	$::=$			memory actions
	$\vert$	$\texttt{create}\,(pval, \tau)$		
	$\vert$	$\texttt{create\_readonly}\,(pval_1, \tau, pval_2)$		
	$\vert$	$\texttt{alloc}\,(pval_1, pval_2)$		
	$\vert$	$\texttt{kill}\,(m\_kill\_kind, pval, pt)$		
	$\vert$	$\texttt{store}\,(bool, \tau, pval_1, pval_2, mem\_order, pt)$		true means store is locking
	$\vert$	$\texttt{load}\,(\tau, pval, mem\_order, pt)$		
	$\vert$	$\texttt{rmw}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$		
	$\vert$	$\texttt{fence}\,(mem\_order)$		
	$\vert$	$\texttt{cmp\_exch\_strong}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$		
	$\vert$	$\texttt{cmp\_exch\_weak}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$		
	$\vert$	$\texttt{linux\_fence}\,(linux\_mem\_order)$		
	$\vert$	$\texttt{linux\_load}\,(\tau, pval, linux\_mem\_order)$		
	$\vert$	$\texttt{linux\_store}\,(\tau, pval_1, pval_2, linux\_mem\_order)$		
	$\vert$	$\texttt{linux\_rmw}\,(\tau, pval_1, pval_2, linux\_mem\_order)$		
$polarity$	$::=$			polarities for memory actions
	$\vert$			(pos) sequenced by `let weak` and `let strong`
	$\vert$	$\texttt{neg}$		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

	|	$\texttt{intFromPtr}\,(\tau_1, \tau_2, pval)$		cast of pointer value to integer value
	|	$\texttt{ptrFromInt}\,(\tau_1, \tau_2, pval)$		cast of integer value to pointer value
	|	$\texttt{ptrValidForDeref}\,(\tau, pval, pt)$		dereferencing validity predicate
	|	$\texttt{ptrWellAligned}\,(\tau, pval)$		
	|	$\texttt{ptrArrayShift}\,(pval_1, \tau, pval_2)$		
	|	$\texttt{memcpy}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{memcmp}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{realloc}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{va\_start}\,(pval_1, pval_2)$		
	|	$\texttt{va\_copy}\,(pval)$		
	|	$\texttt{va\_arg}\,(pval, \tau)$		
	|	$\texttt{va\_end}\,(pval)$		
$spine\_elem$	::=			spine element
	|	$pval$		pure or logical value
	|	res_term		resource value
	|	$\sigma(spine\_elem)$	M	substitution for spine elements / return values
$spine$	::=			spine
	|	$\overline{spine\_elem_i}^{\,i}$		
$tval$	::=			(effectful) top-level values
	|	$\texttt{done}\ spine$		end of top-level expression
	|	$\texttt{undef}\ UB\_name$		undefined behaviour
$res\_pattern$	::=			resource terms
	|	$\texttt{emp}$	binders $= \{\}$	empty heap
	|	$ident$	binders $= ident$	variable
	|	$\texttt{fold}\,(res\_pattern)$	binders $= \{\}$	unfold (recursive) predicate
	|	$\langle res\_pattern_1, res\_pattern_2\rangle$	binders $=$ binders$(res_pattern_1) \cup$ binders$(res_pattern_2)$	seperating-conjunction pair
	|	$\texttt{pack}\,(ident, res\_pattern)$	binders $= ident \cup$ binders$(res_pattern)$	packing for existentials

$\mathit{ret\_pattern}$	::=			return pattern
	|	$\texttt{comp}\ \mathit{ident\_or\_pattern}$	binders = binders($\mathit{ident\_or\_pattern}$)	computational variable
	|	$\texttt{log}\ \mathit{ident}$	binders = ident	logical variable
	|	$\texttt{res}\ \mathit{res\_pattern}$	binders = binders($\mathit{res\_pattern}$)	resource variable
init,	::=			initialisation status
	|	$\checkmark$		initialised
	|	$\times$		uninitalised
$\mathit{points\_to}$, pt	::=			points-to separation logic predicate
	|	$\mathit{term}_1 \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2$		
res	::=			resources
	|	$\texttt{emp}$		empty heap
	|	$\mathit{points\_to}$		points-top heap pred.
	|	$\mathit{res}_1 * \mathit{res}_2$		seperating conjunction
	|	$\exists\, \mathit{ident}{:}\beta.\ \mathit{res}$		existential
	|	$\mathit{term} \wedge \mathit{res}$		logical conjuction
	|	$\texttt{if}\ \mathit{term}\ \texttt{then}\ \mathit{res}_1\ \texttt{else}\ \mathit{res}_2$		ordered disjuction
	|	$\alpha(\,\overline{\mathit{pval}_i}^{\,i}\,)$		predicate
	|	$\sigma(\mathit{res})$	M	simul-sub σ in res
ret, _	::=			return types
	|	$\Sigma\, \mathit{ident}{:}\beta.\ \mathit{ret}$		return a computational value
	|	$\exists\, \mathit{ident}{:}\beta.\ \mathit{ret}$		return a logical value
	|	$\mathit{res} \otimes \mathit{ret}$		return a resource value
	|	$\mathit{term} \wedge \mathit{ret}$		return a predicate (post-condition)
	|	$\texttt{I}$		end return list
	|	$\sigma(\mathit{ret})$	M	simul-sub σ in ret
$\mathit{seq\_expr}$	::=			sequential (effectful) expressions

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

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

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

formula	$::=$			
	$\mid$	$\mathit{judgement}$		
	$\mid$	typing		
	$\mid$	opsem		
	$\mid$	$\mathit{term} \equiv \mathit{term}'$		
	$\mid$	$\mathit{name}{:}\mathit{pure\_arg} \equiv \overline{x_i}^{\,i} \mapsto \mathit{tpexpr} \in \texttt{Globals}$		
	$\mid$	$\mathit{name}{:}\mathit{arg} \equiv \overline{x_i}^{\,i} \mapsto \mathit{texpr} \in \texttt{Globals}$		
heap, *h*, *f*	$::=$			heaps
	$\mid$	$\cdot$		
	$\mid$	$h + \{\mathit{points\_to}\}$		
	$\mid$	$h + f$	[O]	convenient for the soundness proof
wf_jtyp	$::=$			
	$\mid$	$\mathcal{C}; \mathcal{L} \vdash \texttt{guarded\_e}\,(\mathit{expr\_form})$		
	$\mid$	$\mathcal{C}; \mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{expr\_form})$		
	$\mid$	$\mathcal{C}; \mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{guarded}\,(\mathit{form})$		
	$\mid$	$\mathcal{C}; \mathcal{L} \vdash \texttt{vconstr\_e}\,(\mathit{expr\_form})$		
	$\mid$	$\mathcal{C}; \mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form})$		
	$\mid$	$\mathcal{C}; \mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{form})$		
	$\mid$	$\mathcal{C}; \mathcal{L} \vdash \texttt{well\_formed}\,(\mathit{array\_prop\_form})$		
lemma_jtype	$::=$			
	$\mid$	$\overline{x_i}^{\,i} :: \mathit{arg} \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid \mathit{ret}$		
	$\mid$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$		
	$\mid$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$		
res_jtype	$::=$			
	$\mid$	$\Phi \vdash \mathit{res} \equiv \mathit{res}'$		
	$\mid$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{res}$		
	$\mid$	$h{:}\mathcal{R}$		

object_value_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{object\_value} \Rightarrow \mathtt{obj}\, \beta$
pval_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta$
spine_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret}$
pexpr_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pexpr} \Rightarrow \mathit{ident}{:}\beta.\, \mathit{term}$
comp_pattern_jtype	$::=$	
	$\|$	$\mathit{pattern}{:}\beta \leadsto \mathcal{C}\, \mathtt{with}\, \mathit{term}$
	$\|$	$\mathit{ident\_or\_pattern}{:}\beta \leadsto \mathcal{C}\, \mathtt{with}\, \mathit{term}$
res_pattern_jtype	$::=$	
	$\|$	$\Phi \vdash \mathit{res}' = \mathtt{strip\_ifs}\,(\mathit{res})$
	$\|$	$\Phi \vdash \mathit{res}\, \mathtt{as}\, \mathit{res\_pattern} \leadsto \mathcal{L}'; \Phi'; \mathcal{R}'$
	$\|$	$\Phi \vdash \mathit{res\_pattern}{:}\mathit{res} \leadsto \mathcal{L}'; \Phi'; \mathcal{R}'$
ret_pattern_jtype	$::=$	
	$\|$	$\Phi \vdash \overline{\mathit{ret\_pattern}_i}^{\,i} {:}\mathit{ret} \leadsto \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$
tpval_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpval} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term}$
tpexpr_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpexpr} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term}$
action_jtype	$::=$	

	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{mem\_action} \Rightarrow \mathit{ret}$
memop_jtype	::=	
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{mem\_op} \Rightarrow \mathit{ret}$
tval_jtype	::=	
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{tval} \Leftarrow \mathit{ret}$
seq_expr_jtype	::=	
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{seq\_expr} \Rightarrow \mathit{ret}$
is_expr_jtype	::=	
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{is\_expr} \Rightarrow \mathit{ret}$
texpr_jtype	::=	
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret}$
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}$
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{texpr} \Leftarrow \mathit{ret}$
subs_jtype	::=	
	$\vert$	$\mathit{pattern} = \mathit{pval} \rightsquigarrow \sigma$
	$\vert$	$\mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma$
	$\vert$	$\mathit{res\_pattern} = \mathit{res\_term} \rightsquigarrow \sigma$
	$\vert$	$\overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{i} \rightsquigarrow \sigma$
	$\vert$	$\overline{x_i = \mathit{spine\_elem}_i}^{i} :: \mathit{arg} \gg \sigma; \mathit{ret}$
pure_opsem_jtype	::=	
	$\vert$	$\langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{pexpr}' \rangle$
	$\vert$	$\langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\, \mathit{term}) \rangle$
	$\vert$	$\langle \mathit{tpexpr} \rangle \longrightarrow \langle \mathit{tpexpr}' \rangle$

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(expr\_form)\\ \mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(expr\_form')\end{array}}{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(expr\_form \leq expr\_form')} \quad \textsc{Wf\_Guarded\_Leq}$$

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

$$\frac{\overline{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(form_j)}^{\,j}}{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(\bigvee(\overline{form_j}^{\,j}))} \quad \textsc{Wf\_Guarded\_Or}$$

$$\frac{\overline{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(form_j)}^{\,j}}{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(\bigwedge(\overline{form_j}^{\,j}))} \quad \textsc{Wf\_Guarded\_And}$$

$$\boxed{\mathcal{C};\mathcal{L} \vdash \texttt{vconstr\_e}\,(expr\_form)}$$

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

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L} \vdash \texttt{vconstr\_e}\,(expr\_form_1)\\ \mathcal{C};\mathcal{L} \vdash \texttt{vconstr\_e}\,(expr\_form_2)\end{array}}{\mathcal{C};\mathcal{L} \vdash \texttt{vconstr\_e}\,(expr\_form_1 + expr\_form_2)} \quad \textsc{Wf\_Vconstr\_Eexpr\_Plus}$$

$$\frac{\begin{array}{l} \mathit{ident} \in \mathcal{C};\mathcal{L} \\ \mathcal{C};\mathcal{L} \vdash \texttt{vconstr\_e}\,(\mathit{expr\_form}) \end{array}}{\mathcal{C};\mathcal{L} \vdash \texttt{vconstr\_e}\,(\mathit{ident}[\mathit{expr\_form}])} \quad \textsc{Wf\_Vconstr\_Eexpr\_Index}$$

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

$$\frac{\mathcal{C};\mathcal{L} \vdash \texttt{vconstr\_e}\,(\mathit{expr\_form})}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form})} \quad \textsc{Wf\_Vconstr\_Expr\_Eexpr}$$

$$\frac{\begin{array}{l} \mathit{ident} \in \mathcal{C};\mathcal{L} \\ \mathit{ident}' \in \overline{\mathit{ident}_i}^{\,i} \end{array}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{ident}[\mathit{ident}'])} \quad \textsc{Wf\_Vconstr\_Expr\_Index\_UVar}$$

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

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form}) \\ \mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form}') \end{array}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form} \leq \mathit{expr\_form}')} \quad \textsc{Wf\_Vconstr\_Leq}$$

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form}) \\ \mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form}') \end{array}}{\mathcal{C};\mathcal{L} \mid \overline{\mathit{ident}_i}^{\,i} \vdash \texttt{vconstr}\,(\mathit{expr\_form} = \mathit{expr\_form}')} \quad \textsc{Wf\_Vconstr\_Eq}$$

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

$$\frac{\overline{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{vconstr}\,(form_j)}^{\,j}}{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{vconstr}\,(\bigwedge(\overline{form_j}^{\,j}))}\ \textsc{Wf\_Vconstr\_And}$$

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

$$\frac{\mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{guarded}\,(form_1) \quad \mathcal{C};\mathcal{L} \mid \overline{ident_i}^{\,i} \vdash \texttt{vconstr}\,(form_2)}{\mathcal{C};\mathcal{L} \vdash \texttt{well\_formed}\,(\forall\,\overline{ident_i}^{\,i}\,.\,form_1 \to form_2)}\ \textsc{Wf\_Apf\_Base}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c}\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1 \vdash res\_term_1 \Leftarrow res_1 \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_2 \vdash res\_term_2 \Leftarrow res_2\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R}_2 \vdash \langle res\_term_1, res\_term_2 \rangle \Leftarrow res_1 * res_2} \ \textsc{Ty\_Res\_SepConj}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \texttt{list}\,\beta\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{Cons}(pval_1, pval_2) \Rightarrow \texttt{list}\,\beta} \quad \textsc{Ty\_Pval\_Ctor\_Cons}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\dfrac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_0 \Rightarrow \mathtt{loc}\\ \mathtt{smt}\,(\Phi \Rightarrow \mathit{pval}_0 = \mathit{pval}_1)\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{pval}_1 \mapsto_\tau \_ \Leftarrow \mathit{pval}_1 \mapsto_\tau \_\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{kill}\,(\mathtt{static}\,\tau, \mathit{pval}_0, \mathit{pval}_1 \mapsto_\tau \_) \Rightarrow \Sigma\, \_{:}\mathtt{unit}.\ \mathtt{I}} \quad \textsc{Ty\_Action\_Kill\_Static}$$

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

$$\dfrac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \mathtt{loc}\\ \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_2 \Rightarrow \mathtt{loc}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \mathit{pval}_1\ \mathit{binop}_{rel}\ \mathit{pval}_2 \Rightarrow \Sigma\, y{:}\mathtt{bool}.\ y = (\mathit{pval}_1\ \mathit{binop}_{rel}\ \mathit{pval}_2) \wedge \mathtt{I}} \quad \textsc{Ty\_Memop\_Rel\_Binop}$$

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

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

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

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

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

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

$$\dfrac{}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \mathtt{done}\ \Leftarrow \mathtt{I}} \quad \text{Ty\_TVal\_I}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c} pattern_1 = pval_1 \leadsto \sigma_1 \\ pattern_2 = pval_2 \leadsto \sigma_2 \end{array}}{\texttt{Cons}(pattern_1, pattern_2) = \texttt{Cons}(pval_1, pval_2) \leadsto \sigma_1, \sigma_2} \quad \textsc{Subs\_Decons\_Value\_Cons}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\mathit{mem\_int} \;\equiv\; \mathit{mem\_int}_1\ \mathit{binop}_{\mathit{arith}}\ \mathit{mem\_int}_2}{\langle \mathit{mem\_int}_1\ \ \mathit{binop}_{\mathit{arith}}\ \ \mathit{mem\_int}_2\rangle \longrightarrow \langle \mathit{mem\_int}\rangle} \quad \textsc{Op\_PE\_PE\_Arith\_Binop}$$

$$\frac{\mathit{bool\_value} \;\equiv\; \mathit{mem\_int}_1\ \mathit{binop}_{\mathit{rel}}\ \mathit{mem\_int}_2}{\langle \mathit{mem\_int}_1\ \ \mathit{binop}_{\mathit{rel}}\ \ \mathit{mem\_int}_2\rangle \longrightarrow \langle \mathit{bool\_value}\rangle} \quad \textsc{Op\_PE\_PE\_Rel\_Binop}$$

$$\frac{\mathit{bool\_value} \;\equiv\; \mathit{bool\_value}_1\ \mathit{binop}_{\mathit{bool}}\ \mathit{bool\_value}_2}{\langle \mathit{bool\_value}_1\ \ \mathit{binop}_{\mathit{bool}}\ \ \mathit{bool\_value}_2\rangle \longrightarrow \langle \mathit{bool\_value}\rangle} \quad \textsc{Op\_PE\_PE\_Bool\_Binop}$$

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

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

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

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

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

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

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

$$\frac{\begin{array}{c} pattern_j = pval \rightsquigarrow \sigma_j \\ \forall\, i < j.\ \texttt{not}\,(pattern_i = pval \rightsquigarrow \sigma_i) \end{array}}{\langle \texttt{case}\, pval\, \texttt{of}\, \overline{\mid\, pattern_i \Rightarrow tpexpr_i}^{\,i}\ \texttt{end}\rangle \longrightarrow \langle \sigma_j(tpexpr_j)\rangle} \quad \textsc{Op\_TPE\_TPE\_Case}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\langle h; \mathtt{if}\, \mathtt{True}\, \mathtt{then}\, \mathit{texpr}_1\, \mathtt{else}\, \mathit{texpr}_2\rangle \longrightarrow \langle h; \mathit{texpr}_1\rangle} \quad \textsc{Op\_STE\_TE\_If\_True}$$

$$\frac{}{\langle h; \mathtt{if}\, \mathtt{False}\, \mathtt{then}\, \mathit{texpr}_1\, \mathtt{else}\, \mathit{texpr}_2\rangle \longrightarrow \langle h; \mathit{texpr}_2\rangle} \quad \textsc{Op\_STE\_TE\_If\_False}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Definition rules:        233 good    0 bad
Definition rule clauses: 525 good    0 bad
```