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

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

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

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

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

ident_or_pattern	::=			
	|	*ident*	binders = *ident*	
	|	*pattern*	binders = binders(*pattern*)	
bool_op	::=			
	|	$\neg\, term$		
	|	$term_1 = term_2$		
	|	$term_1 \rightarrow term_2$		
	|	$\bigwedge(\,\overline{term_i}^{\,i}\,)$		
	|	$\bigvee(\,\overline{term_i}^{\,i}\,)$		
	|	$term_1\ binop_{bool}\ term_2$	M	
	|	`if` $term_1$ `then` $term_2$ `else` $term_3$		
arith_op	::=			
	|	$term_1 + term_2$		
	|	$term_1 - term_2$		
	|	$term_1 \times term_2$		
	|	$term_1 / term_2$		
	|	$term_1$ `rem_t` $term_2$		
	|	$term_1$ `rem_f` $term_2$		
	|	$term_1$ `^` $term_2$		
	|	$term_1\ binop_{arith}\ term_2$	M	
cmp_op	::=			
	|	$term_1 < term_2$		less than
	|	$term_1 \leq term_2$		less than or equal
	|	$term_1\ binop_{rel}\ term_2$	M	
list_op	::=			
	|	`nil`		

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

$term$, $\_$	::=			
	|	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
	|	$\texttt{assert\_undef}\,(pval,\ UB\_name)$	
	|	$\texttt{bool\_to\_integer}\,(pval)$	
	|	$\texttt{conv\_int}\,(\tau, pval)$	
	|	$\texttt{wrapI}\,(\tau, pval)$	

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

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

	|	$\texttt{ptrArrayShift}\,(pval_1, \tau, pval_2)$		
	|	$\texttt{memcpy}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{memcmp}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{realloc}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{va\_start}\,(pval_1, pval_2)$		
	|	$\texttt{va\_copy}\,(pval)$		
	|	$\texttt{va\_arg}\,(pval, \tau)$		
	|	$\texttt{va\_end}\,(pval)$		
$spine\_elem$	::=			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
	|	pt	binders $= \{\}$	single-cell heap
	|	$ident$	binders $= ident$	variable
	|	$\langle res\_pattern_1, res\_pattern_2\rangle$	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
$ret\_pattern$	::=			return pattern
	|	$\texttt{comp}\, ident\_or\_pattern$	binders $=$ binders$(ident_or_pattern)$	computational variable

	$\|$	$\texttt{log}\, ident$	binders = $ident$	logical variable
	$\|$	$\texttt{res}\, res\_pattern$	binders = binders($res\_pattern$)	resource variable
$init$,	::=			initialisation status
	$\|$	$\checkmark$		initialised
	$\|$	$\times$		uninitalised
$points\_to$, pt	::=			points-to separation logic predicate
	$\|$	$term_1 \overset{init}{\mapsto}_{\tau} term_2$		
res	::=			resources
	$\|$	$\texttt{emp}$		empty heap
	$\|$	$points\_to$		points-top heap pred.
	$\|$	$res_1 * res_2$		seperating conjunction
	$\|$	$\exists\, ident{:}\beta.\ res$		existential
	$\|$	$term \wedge res$		logical conjuction
	$\|$	$\sigma(res)$	M	simul-sub σ in res
ret, $\_$	::=			return types
	$\|$	$\Sigma\, ident{:}\beta.\ ret$		return a computational value
	$\|$	$\exists\, ident{:}\beta.\ ret$		return a logical value
	$\|$	$res \otimes ret$		return a resource value
	$\|$	$term \wedge ret$		return a predicate (post-condition)
	$\|$	$\texttt{I}$		end return list
	$\|$	$\sigma(ret)$	M	simul-sub σ in ret
seq_expr	::=			sequential (effectful) expressions
	$\|$	$\texttt{ccall}\,(\tau, ident, spine)$		C function call
	$\|$	$\texttt{pcall}\,(name, spine)$		procedure call

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

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

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

$$
\begin{array}{lcll}
 & | & \mathit{typing} & \\
 & | & \mathit{opsem} & \\
 & | & \mathit{term} \;\equiv\; \mathit{term}' & \\
 & | & \mathit{name}{:}\mathit{pure\_arg} \;\equiv\; \overline{x_i}^{\,i} \mapsto \mathit{tpexpr} \;\in\; \texttt{Globals} & \\
 & | & \mathit{name}{:}\mathit{arg} \;\equiv\; \overline{x_i}^{\,i} \mapsto \mathit{texpr} \;\in\; \texttt{Globals} & \\
 & & & \\
\mathit{heap},\ h & ::= & & \text{heaps} \\
 & | & . & \\
 & | & h + \{\mathit{points\_to}\} & \\
 & & & \\
\mathit{lemma\_jtype} & ::= & & \\
 & | & \overline{x_i}^{\,i} :: \mathit{arg} \rightsquigarrow \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \mid \mathit{ret} & \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \sqsubseteq \mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}' & \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash (\sigma){:}(\mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}') & \\
 & & & \\
\mathit{res\_jtype} & ::= & & \\
 & | & \Phi \vdash \mathit{res} \;\equiv\; \mathit{res}' & \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{res} & \\
 & | & h{:}\mathcal{R} & \\
 & & & \\
\mathit{object\_value\_jtype} & ::= & & \\
 & | & \mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \texttt{obj}\,\beta & \\
 & & & \\
\mathit{pval\_jtype} & ::= & & \\
 & | & \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval} \Rightarrow \beta & \\
 & & & \\
\mathit{spine\_jtype} & ::= & & \\
 & | & \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma;\mathit{ret} & \\
 & & & \\
\mathit{pexpr\_jtype} & ::= & &
\end{array}
$$

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

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

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

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

$$\frac{\overline{x_i}^{\,i} :: arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{x,\ \overline{x_i}^{\,i} :: \Pi\, x{:}\beta.\ arg \rightsquigarrow \mathcal{C}, x{:}\beta; \mathcal{L}; \Phi; \mathcal{R} \mid ret} \quad \text{ARG\_ENV\_COMP}$$

$$\frac{\overline{x_i}^{\,i} :: arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{x,\ \overline{x_i}^{\,i} :: \forall\, x{:}\beta.\ arg \rightsquigarrow \mathcal{C}; \mathcal{L}, x{:}\beta; \Phi; \mathcal{R} \mid ret} \quad \text{ARG\_ENV\_LOG}$$

$$\frac{\overline{x_i}^{\,i} :: arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{\overline{x_i}^{\,i} :: term \supset arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi, term; \mathcal{R} \mid ret} \quad \text{ARG\_ENV\_PHI}$$

$$\frac{\overline{x_i}^{\,i} :: arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret}{x,\ \overline{x_i}^{\,i} :: res \multimap arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}, x{:}res \mid ret} \quad \text{ARG\_ENV\_RES}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \texttt{integer} \\ abbrev_1 \;\equiv\; \text{max\_int}_\tau - \text{min\_int}_\tau + 1 \\ abbrev_2 \;\equiv\; pval\,\texttt{rem\_f}\,abbrev_1 \end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{wrapI}\,(\tau, pval) \Rightarrow y{:}\beta.\ y = \texttt{if}\,abbrev_2 \leq \text{max\_int}_\tau\,\texttt{then}\,abbrev_2\,\texttt{else}\,abbrev_2 - abbrev_1} \quad \textsc{Ty\_PE\_WrapI}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_0 \Rightarrow \mathtt{loc} \\ \mathtt{smt}\,(\Phi \Rightarrow \mathit{pval}_1 = \mathit{pval}_0) \\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{pval}_1 \overset{\checkmark}{\mapsto}_\tau \_ \Leftarrow \mathit{pval}_1 \overset{\checkmark}{\mapsto}_\tau \_\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathtt{ptrValidForDeref}\,(\tau, \mathit{pval}_0, \mathit{pval}_1 \overset{\checkmark}{\mapsto}_\tau \_) \Rightarrow \Sigma\, y{:}\mathtt{bool}.\ y = \mathtt{aligned}\,(\tau, \mathit{pval}_1) \wedge \mathit{pval}_1 \overset{\checkmark}{\mapsto}_\tau \_ \otimes \mathtt{I}} \quad \textsc{Ty\_Memop\_PtrValidForDeref}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \mathtt{loc}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \mathtt{ptrWellAligned}\,(\tau, \mathit{pval}) \Rightarrow \Sigma\, y{:}\mathtt{bool}.\ y = \mathtt{aligned}\,(\tau, \mathit{pval}) \wedge \mathtt{I}} \quad \textsc{Ty\_Memop\_PtrWellAligned}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_1 \Rightarrow \mathtt{loc} \\ \mathcal{C};\mathcal{L};\Phi \vdash \mathit{pval}_2 \Rightarrow \mathtt{integer}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \mathtt{ptrArrayShift}\,(\mathit{pval}_1, \tau, \mathit{pval}_2) \Rightarrow \Sigma\, y{:}\mathtt{loc}.\ y = \mathit{pval}_1 +_{\mathrm{ptr}} (\mathit{pval}_2 \times \mathrm{size\_of}(\tau)) \wedge \mathtt{I}} \quad \textsc{Ty\_Memop\_PtrArrayShift}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Definition rules:        202 good    0 bad
Definition rule clauses: 450 good    0 bad
```