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

$Sctypes\_t,\ \tau$	::=			C type
	|	$\texttt{array}\ int\ \tau$		array of length int of element type τ
	|	$\tau*$		pointer to type τ
$int,\ \_$	::=			OCaml fixed-width integer
	|	i		literal integer
	|	n		literal integer
tag	::=			OCaml type for struct/union tag
	|	$ident$		
$\beta,\ \_$	::=			base types
	|	$\texttt{unit}$		unit
	|	$\texttt{bool}$		boolean
	|	$\texttt{integer}$		integer
	|	$\texttt{real}$		rational numbers?
	|	$\texttt{loc}$		location
	|	$\texttt{array}\ \beta$		array
	|	$\texttt{list}\ \beta$		list
	|	$\overline{\beta_i}^{\,i}$		tuple
	|	$\texttt{struct}\ tag$		struct
	|	$\texttt{set}\ \beta$		set
	|	$\texttt{opt}\ (\beta)$		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
	|	`/\`	

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

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

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

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

	$\mid$	$term_1 \leqslant term_2$	less than or equal
	$\mid$	$term_1\ binop_{rel}\ term_2$	M
$list\_op$	$::=$		
	$\mid$	`nil`	
	$\mid$	$term_1 :: term_2$	
	$\mid$	`tl` $term$	
	$\mid$	$term^{(int)}$	
$tuple\_op$	$::=$		
	$\mid$	$(\ \overline{term_i}^{\,i}\ )$	
	$\mid$	$term^{(int)}$	
$pointer\_op$	$::=$		
	$\mid$	mem_ptr	
	$\mid$	$term_1 +_{\mathrm{ptr}} term_2$	
	$\mid$	`cast_int_to_ptr` $term$	
	$\mid$	`cast_ptr_to_int` $term$	
$array\_op$	$::=$		
	$\mid$	$[\mid\ \overline{term_i}^{\,i}\ \mid]$	
	$\mid$	$term_1[term_2]$	
	$\mid$	`const` $term$	
	$\mid$	$term_1[term_2] := term_3$	
$param\_op$	$::=$		
	$\mid$	$ident{:}\beta.\ term$	
	$\mid$	$term(term_1, \,..\, , term_n)$	
$struct\_op$	$::=$		
	$\mid$	$term.member$	

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

	$\mid$	$\texttt{bool\_to\_integer}\,(pval)$		
	$\mid$	$\texttt{conv\_int}\,(\tau, pval)$		
	$\mid$	$\texttt{wrapI}\,(\tau, pval)$		
$tpexpr$	$::=$			top-level pure expressions
	$\mid$	$tpval$		top-level pure values
	$\mid$	$\texttt{case}\, pval\, \texttt{of}\, \overline{\mid\, tpexpr\_case\_branch_i}^{i}\, \texttt{end}$		pattern matching
	$\mid$	$\texttt{let}\, ident\_or\_pattern = pexpr\, \texttt{in}\, tpexpr$	bind binders($ident\_or\_pattern$) in $tpexpr$	pure let
	$\mid$	$\texttt{let}\, ident\_or\_pattern{:}(y_1{:}\beta_1.\, term_1) = tpexpr_1\, \texttt{in}\, tpexpr_2$	bind binders($ident\_or\_pattern$) in $tpexpr_2$ bind y_1 in $term_1$	annoted pure let
	$\mid$	$\texttt{if}\, pval\, \texttt{then}\, tpexpr_1\, \texttt{else}\, tpexpr_2$		pure if
	$\mid$	$\sigma(tpexpr)$	M	simul-sub σ in $tpexpr$
$tpexpr\_case\_branch$	$::=$			pure top-level case expression
	$\mid$	$pattern \Rightarrow tpexpr$	bind binders($pattern$) in $tpexpr$	top-level case expression br
m_kill_kind	$::=$			
	$\mid$	$\texttt{dynamic}$		
	$\mid$	$\texttt{static}\,\tau$		
$bool$, $\_$	$::=$			OCaml booleans
	$\mid$	$\texttt{true}$		
	$\mid$	$\texttt{false}$		
$points\_to$, pt	$::=$			points-to separation logic prec
	$\mid$	$term_1 \overset{init}{\mapsto}_{\tau} term_2$		
$qpoints\_to$, qpt	$::=$			quantified (integer-indexed) p
	$\mid$	$\ast x.\, iguard; term_1 + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_{\tau} term_2$		
res_term	$::=$			resource terms

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

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

	| `undef` UB_name		undefined behaviour
$res\_pattern$ ::=			resource terms
	| `emp`	binders = $\{\}$	empty heap
	| $ident$	binders = $ident$	variable
	| `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
	| `pack` $(ident, res\_pattern)$	binders = $ident$ $\cup$ binders($res\_pattern$)	packing for existentials
$ret\_pattern$ ::=			return pattern
	| `comp` $ident\_or\_pattern$	binders = binders($ident\_or\_pattern$)	computational variable
	| `log` $ident$	binders = $ident$	logical variable
	| `res` $res\_pattern$	binders = binders($res\_pattern$)	resource variable
$init$, ::=			initialisation status
	| ✓		initialised
	| $\times$		uninitalised
res ::=			resources
	| `emp`		empty heap
	| $points\_to$		points-to heap pred.
	| $qpoints\_to$		quantified (integer-indexed) points-to heap pred.
	| $res_1 * res_2$		seperating conjunction
	| $\exists\, ident{:}\beta.\ res$		existential
	| $term \wedge res$		logical conjuction
	| `if` $term$ `then` res_1 `else` res_2		ordered disjuction
	| $\alpha(\overline{pval_i}^{\,i})$		predicate
	| $\sigma(res)$	M	simul-sub σ in res
ret, $\_$ ::=			return types
	| $\Sigma\, ident{:}\beta.\ ret$		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-conditior
	|	$\mathtt{I}$		end return list
	|	$\sigma(\mathit{ret})$	M	simul-sub σ in ret
$\mathit{seq\_expr}$	::=			sequential (effectful) expressions
	|	$\mathtt{ccall}\,(\tau, \mathit{ident}, \mathit{spine})$		C function call
	|	$\mathtt{pcall}\,(\mathit{name}, \mathit{spine})$		procedure call
$\mathit{seq\_texpr}$	::=			sequential top-level (effectful) expre
	|	tval		(effectful) top-level values
	|	$\mathtt{run}\, \mathit{ident}\, \overline{\mathit{pval}_i}^{\,i}$		run from label
	|	$\mathtt{let}\, \mathit{ident\_or\_pattern} = \mathit{pexpr}\, \mathtt{in}\, \mathit{texpr}$	bind binders($\mathit{ident\_or\_pattern}$) in texpr	pure let
	|	$\mathtt{let}\, \mathit{ident\_or\_pattern}{:}(y_1{:}\beta_1.\ \mathit{term}_1) = \mathit{tpexpr}\, \mathtt{in}\, \mathit{texpr}$	bind binders($\mathit{ident\_or\_pattern}$) in texpr bind y_1 in term_1	annotated pure let
	|	$\mathtt{let}\, \overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{seq\_expr}\, \mathtt{in}\, \mathit{texpr}$	bind binders($\overline{\mathit{ret\_pattern}_i}^{\,i}$) in texpr	bind return patterns
	|	$\mathtt{let}\, \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} = \mathit{texpr}_1\, \mathtt{in}\, \mathit{texpr}_2$	bind binders($\overline{\mathit{ret\_pattern}_i}^{\,i}$) in texpr_2	annotated bind return patterns
	|	$\mathtt{case}\, \mathit{pval}\, \mathtt{of}\, \overline{\mid \mathit{texpr\_case\_branch}_i}^{\,i}\, \mathtt{end}$		pattern matching
	|	$\mathtt{if}\, \mathit{pval}\, \mathtt{then}\, \mathit{texpr}_1\, \mathtt{else}\, \mathit{texpr}_2$		conditional
	|	$\mathtt{bound}\,[\mathit{int}](\mathit{is\_texpr})$		limit scope of indet seq behaviour
$\mathit{texpr\_case\_branch}$	::=			top-level case expression branch
	|	$\mathit{pattern} \Rightarrow \mathit{texpr}$	bind binders($\mathit{pattern}$) in texpr	top-level case expression branch
$\mathit{is\_expr}$	::=			indet seq (effectful) expressions
	|	tval		(effectful) top-level values
	|	$\mathtt{memop}\,(\mathit{mem\_op})$		pointer op involving memory
	|	$\mathit{pol\_mem\_action}$		memory action
$\mathit{is\_texpr}$	::=			indet seq top-level (effectful) expres

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

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

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

	$\vert$	$\overline{x_i}^{\,i} :: arg \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid ret$
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$
res_jtype	$::=$	
	$\vert$	$\Phi \vdash res \ \equiv \ res'$
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash res\_term \Leftarrow res$
	$\vert$	$h{:}\mathcal{R}$
$object\_value\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash object\_value \Rightarrow \texttt{obj}\,\beta$
$pval\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash pval \Rightarrow \beta$
$spine\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \ \overline{x_i = spine\_elem_i}^{\,i} \ :: arg \gg \sigma; ret$
$pexpr\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash pexpr \Rightarrow ident{:}\beta.\, term$
$comp\_pattern\_jtype$	$::=$	
	$\vert$	$pattern{:}\beta \rightsquigarrow \mathcal{C}\,\texttt{with}\,term$
	$\vert$	$ident\_or\_pattern{:}\beta \rightsquigarrow \mathcal{C}\,\texttt{with}\,term$
$res\_pattern\_jtype$	$::=$	
	$\vert$	$\Phi \vdash res' = \texttt{strip\_ifs}\,(res)$
	$\vert$	$\Phi \vdash res\,\texttt{as}\,res\_pattern \rightsquigarrow \mathcal{L}'; \Phi'; \mathcal{R}'$
	$\vert$	$\Phi \vdash res\_pattern{:}res \rightsquigarrow \mathcal{L}'; \Phi'; \mathcal{R}'$
$ret\_pattern\_jtype$	$::=$	

	$\vert$	$\Phi \vdash \overline{ret\_pattern_i}^{\,i} : ret \leadsto \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$
$tpval\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash tpval \Leftarrow ident{:}\beta.\, term$
$tpexpr\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash tpexpr \Leftarrow ident{:}\beta.\, term$
$action\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash mem\_action \Rightarrow ret$
$memop\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash mem\_op \Rightarrow ret$
$tval\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash tval \Leftarrow ret$
seq_expr_jtype	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash seq\_expr \Rightarrow ret$
is_expr_jtype	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash is\_expr \Rightarrow ret$
$texpr\_jtype$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash seq\_texpr \Leftarrow ret$
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash is\_texpr \Leftarrow ret$
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash texpr \Leftarrow ret$
$subs\_jtype$	$::=$	
	$\vert$	$pattern = pval \leadsto \sigma$
	$\vert$	$ident\_or\_pattern = pval \leadsto \sigma$

$$
\begin{array}{lll}
 & | & res\_pattern = res\_term \rightsquigarrow \sigma \\
 & | & \overline{ret\_pattern_i = spine\_elem_i}^{\,i} \rightsquigarrow \sigma \\
 & | & \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret
\end{array}
$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} qpt_1 \equiv \circledast x.\ iguard; term_1 + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau term_2 \\ qpt \equiv \circledast x.\ iguard \wedge (x\ \texttt{!=}\ i); term_1 + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau term_2 \\ pt \equiv term_1 + i \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau i/x, \cdot(term_2) \\ \Phi \vdash pt \equiv pt' \\ \Phi \vdash qpt \equiv qpt' \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash res\_term \Leftarrow qpt \end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{break}\,(res\_term{:}qpt_1, i) \Leftarrow qpt' * pt'} \quad \textsc{Ty\_Res\_Break}$$

$$\frac{\begin{array}{l} qpt_1 \equiv \circledast x.\ iguard_1; term_{11} + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau term_{12} \\ pt_2 \equiv term_{21} \overset{init}{\mapsto}_\tau term_{22} \\ i \equiv (term_{21} - term_{11})/\mathrm{size\_of}(\tau) \\ qpt \equiv \circledast x.\ (iguard_1 \vee x = i); term_{11} + x \times \mathrm{size\_of}(\tau) \overset{init}{\mapsto}_\tau \texttt{if}\ x = i\ \texttt{then}\ term_{22}\ \texttt{else}\ term_{12} \\ \Phi \vdash qpt \equiv qpt' \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1 \vdash res\_term_1 \Leftarrow qpt_1 \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_2 \vdash res\_term_2 \Leftarrow pt_2 \end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R}_2 \vdash \texttt{glue}\,(res\_term_1{:}qpt_1, res\_term_2{:}pt_2) \Leftarrow qpt'} \quad \textsc{Ty\_Res\_Glue}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\mathit{res\_pattern} = \mathtt{qpt} \leadsto \sigma}{\mathit{res\_pattern} = \mathtt{glue}\,(\mathit{res\_term}_1{:}\mathit{qpt}_1, \mathit{res\_term}_2{:}\mathit{pt}_2) \leadsto \sigma} \quad \textsc{Subs\_Decons\_Res\_Glue}$$

$$\frac{\mathit{res\_pattern} = \mathtt{qpt} \leadsto \sigma}{\mathit{res\_pattern} = \mathtt{explode}\,(\mathit{res\_term}{:}\mathit{pt}) \leadsto \sigma} \quad \textsc{Subs\_Decons\_Res\_Explode}$$

$$\frac{\mathit{res\_pattern} = \mathtt{qpt} \leadsto \sigma}{\mathit{res\_pattern} = \mathtt{implode}\,(\mathit{res\_term}{:}\mathit{qpt}, \mathit{int}) \leadsto \sigma} \quad \textsc{Subs\_Decons\_Res\_Implode}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Definition rules:        237 good    0 bad
Definition rule clauses: 547 good    0 bad
```