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

$Sctypes\_t,\ \tau$	::=			C type
	|	$\tau*$		pointer to type τ
tag	::=			OCaml type for struct/union tag
	|	$ident$		
$\beta,\ \_$	::=			base types
	|	`unit`		unit
	|	`bool`		boolean
	|	`integer`		integer
	|	`real`		rational numbers?
	|	`loc`		location
	|	`array` β		array
	|	`list` β		list
	|	$\overline{\beta_i}^{\,i}$		tuple
	|	`struct` tag		struct
	|	`set` β		set
	|	`opt` (β)		option
	|	$\beta \rightarrow \beta'$		parameter types
	|	β_τ	M	of a C type
$binop$	::=			binary operators
	|	`+`		addition
	|	`-`		subtraction
	|	`*`		multiplication
	|	`/`		division
	|	`rem_t`		modulus
	|	`rem_f`		remainder
	|	`^`		exponentiation
	|	`=`		equality, defined both for integer and C types

	$\mid$	`!=`	inequality, similiarly defined
	$\mid$	`>`	greater than, similarly defined
	$\mid$	`<`	less than, similarly defined
	$\mid$	`>=`	greater than or equal to, similarly defined
	$\mid$	`<=`	less than or equal to, similarly defined
	$\mid$	`/\`	conjucntion
	$\mid$	`\/`	disjunction
$binop_{arith}$	::=		arithmentic binary operators
	$\mid$	`+`	
	$\mid$	`-`	
	$\mid$	`*`	
	$\mid$	`/`	
	$\mid$	`rem_t`	
	$\mid$	`rem_f`	
	$\mid$	`^`	
$binop_{rel}$	::=		relational binary operators
	$\mid$	`=`	
	$\mid$	`!=`	
	$\mid$	`>`	
	$\mid$	`<`	
	$\mid$	`>=`	
	$\mid$	`<=`	
$binop_{bool}$	::=		boolean binary operators
	$\mid$	`/\`	
	$\mid$	`\/`	
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
	$\vert$	$\mathtt{array}\,(\,\overline{loaded\_value_i}^{\,i}\,)$	C array value
	$\vert$	$(\,\mathtt{struct}\,ident)\{\,\overline{.\,member_i{:}\tau_i = mem\_val_i}^{\,i}\,\}$	C struct value
	$\vert$	$(\,\mathtt{union}\,ident)\{\,.\,member = mem\_val\}$	C union value
loaded_value	::=		potentially unspecified C object values
	$\vert$	$\mathtt{specified}\,object\_value$	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
	$\vert$	$\mathtt{Nil}\,\beta$	empty list
	$\vert$	`Cons`	list cons
	$\vert$	`Tuple`	tuple

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

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

$ident\_or\_pattern$	$::=$			
	$\vert$	$ident$	binders $=$ $ident$	
	$\vert$	$pattern$	binders $=$ binders($pattern$)	
$bool\_op$	$::=$			
	$\vert$	$\neg\, term$		
	$\vert$	$term_1 = term_2$		
	$\vert$	$term_1 \to term_2$		
	$\vert$	$\bigwedge(\,\overline{term_i}^{\,i}\,)$		
	$\vert$	$\bigvee(\,\overline{term_i}^{\,i}\,)$		
	$\vert$	$term_1\ binop_{bool}\ term_2$	M	
	$\vert$	`if` $term_1$ `then` $term_2$ `else` $term_3$		
$arith\_op$	$::=$			
	$\vert$	$term_1 + term_2$		
	$\vert$	$term_1 - term_2$		
	$\vert$	$term_1 \times term_2$		
	$\vert$	$term_1 / term_2$		
	$\vert$	$term_1$ `rem_t` $term_2$		
	$\vert$	$term_1$ `rem_f` $term_2$		
	$\vert$	$term_1$ `^` $term_2$		
	$\vert$	$term_1\ binop_{arith}\ term_2$	M	
cmp_op	$::=$			
	$\vert$	$term_1 < term_2$		less than
	$\vert$	$term_1 \leq term_2$		less than or equal
	$\vert$	$term_1\ binop_{rel}\ term_2$	M	
$list\_op$	$::=$			
	$\vert$	`nil`		

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

$term$, _	::=			
	|	lit		
	|	$arith\_op$		
	|	$bool\_op$		
	|	cmp_op		
	|	$tuple\_op$		
	|	$struct\_op$		
	|	$pointer\_op$		
	|	$list\_op$		
	|	$array\_op$		
	|	ct_pred		
	|	$param\_op$		
	|	$(term)$	S	parentheses
	|	$\sigma(term)$	M	simul-sub σ in $term$
	|	$pval$	M	

$pexpr$	::=		pure expressions
	|	$pval$	pure values
	|	$ctor\_expr(\overline{pval_i}^{\,i})$	data constructor application
	|	$\texttt{array\_shift}\,(pval_1, \tau, pval_2)$	pointer array shift
	|	$\texttt{member\_shift}\,(pval, ident, member)$	pointer struct/union member shift
	|	$\texttt{not}\,(pval)$	boolean not
	|	$pval_1\ binop\ pval_2$	binary operations
	|	$\texttt{memberof}\,(ident, member, pval)$	C struct/union member access
	|	$name(\overline{pval_i}^{\,i})$	pure function call
	|	$\texttt{assert\_undef}\,(pval,\ UB\_name)$	
	|	$\texttt{bool\_to\_integer}\,(pval)$	
	|	$\texttt{conv\_int}\,(\tau, pval)$	
	|	$\texttt{wrapI}\,(\tau, pval)$	

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

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

	|	$\texttt{ptrArrayShift}\,(pval_1, \tau, pval_2)$		
	|	$\texttt{memcpy}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{memcmp}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{realloc}\,(pval_1, pval_2, pval_3)$		
	|	$\texttt{va\_start}\,(pval_1, pval_2)$		
	|	$\texttt{va\_copy}\,(pval)$		
	|	$\texttt{va\_arg}\,(pval, \tau)$		
	|	$\texttt{va\_end}\,(pval)$		
$spine\_elem$	::=			spine element
	|	$pval$		pure or logical value
	|	res_term		resource value
	|	$\sigma(spine\_elem)$	M	substitution for spine elements / return values
$spine$	::=			spine
	|	$\overline{spine\_elem_i}^{\,i}$		
$tval$	::=			(effectful) top-level values
	|	$\texttt{done}\,spine$		end of top-level expression
	|	$\texttt{undef}\ UB\_name$		undefined behaviour
$res\_pattern$	::=			resource terms
	|	$\texttt{emp}$	binders $= \{\}$	empty heap
	|	pt	binders $= \{\}$	single-cell heap
	|	$ident$	binders $= ident$	variable
	|	$\langle res\_pattern_1, res\_pattern_2\rangle$	binders $=$ binders$(res_pattern_1) \cup$ binders$(res_pattern_2)$	seperating-conjunction pair
	|	$\texttt{pack}\,(ident, res\_pattern)$	binders $= ident \cup$ binders$(res_pattern)$	packing for existentials
$ret\_pattern$	::=			return pattern
	|	$\texttt{comp}\,ident\_or\_pattern$	binders $=$ binders$(ident_or_pattern)$	computational variable

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

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

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

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

	$\vert$	$typing$	
	$\vert$	$opsem$	
	$\vert$	$term \equiv term'$	
	$\vert$	$name{:}pure\_arg \equiv \overline{x_i}^{\,i} \mapsto tpexpr \in \texttt{Globals}$	
	$\vert$	$name{:}arg \equiv \overline{x_i}^{\,i} \mapsto texpr \in \texttt{Globals}$	
$heap,\ h$	$::=$		heaps
	$\vert$	$\cdot$	
	$\vert$	$h + \{points\_to\}$	
$object\_value\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash object\_value \Rightarrow \texttt{obj}\, \beta$	
$pval\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash pval \Rightarrow \beta$	
res_jtype	$::=$		
	$\vert$	$\Phi \vdash res \equiv res'$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash res\_term \Leftarrow res$	
$spine\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \overline{x_i = spine\_elem_i}^{\,i} :: arg \gg \sigma; ret$	
$pexpr\_jtype$	$::=$		
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash pexpr \Rightarrow ident{:}\beta.\, term$	
$comp\_pattern\_jtype$	$::=$		
	$\vert$	$pattern{:}\beta \leadsto \mathcal{C}\, \texttt{with}\, term$	
	$\vert$	$ident\_or\_pattern{:}\beta \leadsto \mathcal{C}\, \texttt{with}\, term$	

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

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

subs_jtype ::=
| $\mathit{pattern} = \mathit{pval} \rightsquigarrow \sigma$
| $\mathit{ident\_or\_pattern} = \mathit{pval} \rightsquigarrow \sigma$
| $\mathit{res\_pattern} = \mathit{res\_term} \rightsquigarrow \sigma$
| $\overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{\,i} \rightsquigarrow \sigma$
| $\overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret}$

pure_opsem_jtype ::=
| $\langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{pexpr}' \rangle$
| $\langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\, \mathit{term}) \rangle$
| $\langle \mathit{tpexpr} \rangle \longrightarrow \langle \mathit{tpexpr}' \rangle$

opsem_jtype ::=
| $\langle h; \mathit{seq\_expr} \rangle \longrightarrow \langle h'; \mathit{texpr}{:}\mathit{ret} \rangle$
| $\langle h; \mathit{seq\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle$
| $\langle h; \mathit{mem\_op} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle$
| $\langle h; \mathit{mem\_action} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle$
| $\langle h; \mathit{is\_expr} \rangle \longrightarrow \langle h'; \mathit{is\_expr}' \rangle$
| $\langle h; \mathit{is\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle$
| $\langle h; \mathit{texpr} \rangle \longrightarrow \langle h'; \mathit{texpr}' \rangle$

lemma_jtype ::=
| $\overline{x_i}^{\,i} :: \mathit{arg} \rightsquigarrow \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \mid \mathit{ret}$
| $\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \sqsubseteq \mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}'$
| $\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash (\sigma){:}(\mathcal{C}';\mathcal{L}';\Phi';\mathcal{R}')$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\Phi \vdash \texttt{emp} \ \equiv\ \texttt{emp}} \quad \text{TY\_RES\_EQ\_EMP}$$

$$\frac{\texttt{smt}\,(\Phi \Rightarrow (\mathit{term}_1 = \mathit{term}_1') \wedge (\mathit{term}_2 = \mathit{term}_2'))}{\Phi \vdash \mathit{term}_1 \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2 \ \equiv\ \mathit{term}_1' \overset{\mathit{init}}{\mapsto}_{\tau} \mathit{term}_2'} \quad \text{TY\_RES\_EQ\_POINTSTO}$$

$$\frac{\begin{array}{c}\Phi \vdash \mathit{res}_1 \ \equiv\ \mathit{res}_1'\\ \Phi \vdash \mathit{res}_2 \ \equiv\ \mathit{res}_2'\end{array}}{\Phi \vdash \mathit{res}_1 * \mathit{res}_2 \ \equiv\ \mathit{res}_1' * \mathit{res}_2'} \quad \text{TY\_RES\_EQ\_SEPCONJ}$$

$$\frac{\Phi \vdash \mathit{res} \ \equiv\ \mathit{res}'}{\Phi \vdash \exists\,\mathit{ident}{:}\beta.\ \mathit{res} \ \equiv\ \exists\,\mathit{ident}{:}\beta.\ \mathit{res}'} \quad \text{TY\_RES\_EQ\_EXISTS}$$

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow (\mathit{term} \to \mathit{term}') \wedge (\mathit{term}' \to \mathit{term}))\\ \Phi \vdash \mathit{res} \ \equiv\ \mathit{res}'\end{array}}{\Phi \vdash \mathit{term} \wedge \mathit{res} \ \equiv\ \mathit{term}' \wedge \mathit{res}'} \quad \text{TY\_RES\_EQ\_TERM}$$

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

$$\frac{}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{emp} \Leftarrow \texttt{emp}} \quad \text{TY\_RES\_EMP}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \texttt{bool} \\ \mathcal{C};\mathcal{L};\Phi, pval = \texttt{true} \vdash tpexpr_1 \Leftarrow y{:}\beta.\ term \\ \mathcal{C};\mathcal{L};\Phi, pval = \texttt{false} \vdash tpexpr_2 \Leftarrow y{:}\beta.\ term \end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{if}\, pval\, \texttt{then}\, tpexpr_1\, \texttt{else}\, tpexpr_2 \Leftarrow y{:}\beta.\ term}\quad \textsc{Ty\_TPE\_If}$$

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L};\Phi \vdash pexpr \Rightarrow y_1{:}\beta_1.\ term_1 \\ ident\_or\_pattern{:}\beta_1 \rightsquigarrow \mathcal{C}_1\, \texttt{with}\, term \\ \mathcal{C},\mathcal{C}_1;\mathcal{L};\Phi, term/y_1,\cdot(term_1) \vdash tpexpr \Leftarrow y_2{:}\beta_2.\ term_2 \end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \texttt{let}\, ident\_or\_pattern = pexpr\, \texttt{in}\, tpexpr \Leftarrow y_2{:}\beta_2.\ term_2}\quad \textsc{Ty\_TPE\_Let}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L};\Phi \vdash pexpr \;\Rightarrow\; y{:}\beta.\; term \\ ident\_or\_pattern{:}\beta \rightsquigarrow \mathcal{C}_1\;\texttt{with}\,term_1 \\ \mathcal{C},\mathcal{C}_1;\mathcal{L};\Phi, term_1/y, \cdot(term);\mathcal{R} \vdash texpr \Leftarrow ret \end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{let}\,ident\_or\_pattern = pexpr\,\texttt{in}\,texpr \Leftarrow ret} \quad \textsc{Ty\_Seq\_TE\_LetP}$$

$$\frac{\begin{array}{l} \mathcal{C};\mathcal{L};\Phi \vdash tpexpr \;\Leftarrow\; y{:}\beta.\; term \\ ident\_or\_pattern{:}\beta \rightsquigarrow \mathcal{C}_1\;\texttt{with}\,term_1 \\ \mathcal{C},\mathcal{C}_1;\mathcal{L};\Phi, term_1/y, \cdot(term);\mathcal{R} \vdash texpr \Leftarrow ret \end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{let}\,ident\_or\_pattern{:}(y{:}\beta.\; term) = tpexpr\,\texttt{in}\,texpr \Leftarrow ret} \quad \textsc{Ty\_Seq\_TE\_LetPT}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l}\mathtt{fresh}\,(mem\_ptr)\\ \mathtt{representable}\,(\tau*, mem\_ptr)\\ \mathtt{alignedI}\,(mem\_int, mem\_ptr)\\ pval{:}\beta_\tau\end{array}}{\langle h; \mathtt{create}\,(mem\_int, \tau)\rangle \longrightarrow \langle h + \{mem\_ptr \overset{\times}{\mapsto}_\tau pval\}; \mathtt{done}\, mem\_ptr, pval, mem\_ptr \overset{\times}{\mapsto}_\tau pval\rangle} \quad \text{OP\_ACTION\_TVAL\_CREATE}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Definition rules:        200 good    0 bad
Definition rule clauses: 446 good    0 bad
```