$ident$, 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
	|	$\tau *$		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
	|	`!=`		inequality, similiarly defined

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

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

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

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

	$\vert$	$term_1 = term_2$		
	$\vert$	$term_1 \to term_2$		
	$\vert$	$\bigwedge(\,\overline{term_i}^{\,i}\,)$		
	$\vert$	$\bigvee(\,\overline{term_i}^{\,i}\,)$		
	$\vert$	$term_1\ binop_{bool}\ term_2$	M	
	$\vert$	$\mathtt{if}\ term_1\ \mathtt{then}\ term_2\ \mathtt{else}\ term_3$		
$arith\_op$	$::=$			
	$\vert$	$term_1 + term_2$		
	$\vert$	$term_1 - term_2$		
	$\vert$	$term_1 \times term_2$		
	$\vert$	$term_1 / term_2$		
	$\vert$	$term_1\ \mathtt{rem\_t}\ term_2$		
	$\vert$	$term_1\ \mathtt{rem\_f}\ term_2$		
	$\vert$	$term_1\ \hat{}\ term_2$		
	$\vert$	$term_1\ binop_{arith}\ term_2$	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	
$list\_op$	$::=$			
	$\vert$	$\mathtt{nil}$		
	$\vert$	$term_1 :: term_2$		
	$\vert$	$\mathtt{tl}\ term$		
	$\vert$	$term^{(int)}$		
$tuple\_op$	$::=$			
	$\vert$	$(\,\overline{term_i}^{\,i}\,)$		
	$\vert$	$term^{(int)}$		

$$
\begin{array}{lcl}
\mathit{pointer\_op} & ::= & \\
& | & \mathit{mem\_ptr} \\
& | & \mathit{term}_1 +_{\mathrm{ptr}} \mathit{term}_2 \\
& | & \texttt{cast\_int\_to\_ptr}\, \mathit{term} \\
& | & \texttt{cast\_ptr\_to\_int}\, \mathit{term} \\
\\
\mathit{array\_op} & ::= & \\
& | & [|\ \overline{\mathit{term}_i}^{\,i}\ |] \\
& | & \mathit{term}_1[\mathit{term}_2] \\
\\
\mathit{param\_op} & ::= & \\
& | & \mathit{ident}{:}\beta.\, \mathit{term} \\
& | & \mathit{term}(\mathit{term}_1, \,..\, , \mathit{term}_n) \\
\\
\mathit{struct\_op} & ::= & \\
& | & \mathit{term}.\mathit{member} \\
\\
\mathit{ct\_pred} & ::= & \\
& | & \texttt{representable}\,(\tau, \mathit{term}) \\
& | & \texttt{aligned}\,(\tau, \mathit{term}) \\
& | & \texttt{alignedI}\,(\mathit{term}_1, \mathit{term}_2) \\
\\
\mathit{term},\ \_ & ::= & \\
& | & \mathit{lit} \\
& | & \mathit{arith\_op} \\
& | & \mathit{bool\_op} \\
& | & \mathit{cmp\_op} \\
& | & \mathit{tuple\_op} \\
& | & \mathit{struct\_op} \\
& | & \mathit{pointer\_op} \\
& | & \mathit{list\_op}
\end{array}
$$

	$\vert$	$array\_op$		
	$\vert$	ct_pred		
	$\vert$	$param\_op$		
	$\vert$	$(term)$	S	parentheses
	$\vert$	$\sigma(term)$	M	simul-sub σ in $term$
	$\vert$	$pval$	M	
$pexpr$	::=			pure expressions
	$\vert$	$pval$		pure values
	$\vert$	$ctor\_expr(\overline{pval_i}^{i})$		data constructor applicatio
	$\vert$	$\texttt{array\_shift}\,(pval_1, \tau, pval_2)$		pointer array shift
	$\vert$	$\texttt{member\_shift}\,(pval, ident, member)$		pointer struct/union memb
	$\vert$	$\texttt{not}\,(pval)$		boolean not
	$\vert$	$pval_1\ binop\ pval_2$		binary operations
	$\vert$	$\texttt{memberof}\,(ident, member, pval)$		C struct/union member ac
	$\vert$	$name(\overline{pval_i}^{i})$		pure function call
	$\vert$	$\texttt{assert\_undef}\,(pval,\ UB\_name)$		
	$\vert$	$\texttt{bool\_to\_integer}\,(pval)$		
	$\vert$	$\texttt{conv\_int}\,(\tau, pval)$		
	$\vert$	$\texttt{wrapI}\,(\tau, pval)$		
$tpexpr$	::=			top-level pure expressions
	$\vert$	$tpval$		top-level pure values
	$\vert$	$\texttt{case}\,pval\,\texttt{of}\,\overline{\vert\ tpexpr\_case\_branch_i}^{i}\,\texttt{end}$		pattern matching
	$\vert$	$\texttt{let}\,ident\_or\_pattern = pexpr\,\texttt{in}\,tpexpr$	bind binders($ident\_or\_pattern$) in $tpexpr$	pure let
	$\vert$	$\texttt{let}\,ident\_or\_pattern{:}(y_1{:}\beta_1.\ term_1) = tpexpr_1\,\texttt{in}\,tpexpr_2$	bind binders($ident\_or\_pattern$) in $tpexpr_2$ bind y_1 in $term_1$	annoted pure let
	$\vert$	$\texttt{if}\,pval\,\texttt{then}\,tpexpr_1\,\texttt{else}\,tpexpr_2$		pure if
	$\vert$	$\sigma(tpexpr)$	M	simul-sub σ in $tpexpr$
$tpexpr\_case\_branch$	::=			pure top-level case expression

	$\vert$	$pattern \Rightarrow tpexpr$	bind binders($pattern$) in $tpexpr$	top-level case expression branch
m_kill_kind	::=			
	$\vert$	`dynamic`		
	$\vert$	`static` τ		
$bool$, _	::=			OCaml booleans
	$\vert$	`true`		
	$\vert$	`false`		
int, _	::=			OCaml fixed-width integer
	$\vert$	i		literal integer
res_term	::=			resource terms
	$\vert$	`emp`		empty heap
	$\vert$	$points\_to$		single-cell heap
	$\vert$	$ident$		variable
	$\vert$	$\langle res\_term_1, res\_term_2\rangle$		seperating-conjunction pair
	$\vert$	`pack` $(pval, res\_term)$		packing for existentials
	$\vert$	`fold` (res_term)		fold into recursive res. pred.
	$\vert$	$\sigma(res\_term)$	M	substitution for resource terms
mem_action	::=			memory actions
	$\vert$	`create` $(pval, \tau)$		
	$\vert$	`create_readonly` $(pval_1, \tau, pval_2)$		
	$\vert$	`alloc` $(pval_1, pval_2)$		
	$\vert$	`kill` $(m\_kill\_kind, pval, pt)$		
	$\vert$	`store` $(bool, \tau, pval_1, pval_2, mem\_order, pt)$		true means store is locking
	$\vert$	`load` $(\tau, pval, mem\_order, pt)$		
	$\vert$	`rmw` $(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$		
	$\vert$	`fence` (mem_order)		

	$\mid$	$\texttt{cmp\_exch\_strong}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$	
	$\mid$	$\texttt{cmp\_exch\_weak}\,(\tau, pval_1, pval_2, pval_3, mem\_order_1, mem\_order_2)$	
	$\mid$	$\texttt{linux\_fence}\,(linux\_mem\_order)$	
	$\mid$	$\texttt{linux\_load}\,(\tau, pval, linux\_mem\_order)$	
	$\mid$	$\texttt{linux\_store}\,(\tau, pval_1, pval_2, linux\_mem\_order)$	
	$\mid$	$\texttt{linux\_rmw}\,(\tau, pval_1, pval_2, linux\_mem\_order)$	
polarity	::=		polarities for memory actions
	$\mid$		(pos) sequenced by `let weak` and `let strong`
	$\mid$	`neg`	only sequenced by `let strong`
pol_mem_action	::=		memory actions with polarity
	$\mid$	$polarity\ mem\_action$	
mem_op	::=		operations involving the memory state
	$\mid$	$pval_1\ \ binop_{rel}\ \ pval_2$	pointer relational binary operations
	$\mid$	$pval_1\ -_{\tau}\ pval_2$	pointer subtraction
	$\mid$	$\texttt{intFromPtr}\,(\tau_1, \tau_2, pval)$	cast of pointer value to integer value
	$\mid$	$\texttt{ptrFromInt}\,(\tau_1, \tau_2, pval)$	cast of integer value to pointer value
	$\mid$	$\texttt{ptrValidForDeref}\,(\tau, pval, pt)$	dereferencing validity predicate
	$\mid$	$\texttt{ptrWellAligned}\,(\tau, pval)$	
	$\mid$	$\texttt{ptrArrayShift}\,(pval_1, \tau, pval_2)$	
	$\mid$	$\texttt{memcpy}\,(pval_1, pval_2, pval_3)$	
	$\mid$	$\texttt{memcmp}\,(pval_1, pval_2, pval_3)$	
	$\mid$	$\texttt{realloc}\,(pval_1, pval_2, pval_3)$	
	$\mid$	$\texttt{va\_start}\,(pval_1, pval_2)$	
	$\mid$	$\texttt{va\_copy}\,(pval)$	
	$\mid$	$\texttt{va\_arg}\,(pval, \tau)$	
	$\mid$	$\texttt{va\_end}\,(pval)$	
spine_elem	::=		spine element

	$\mid$	$pval$		pure or logical value
	$\mid$	res_term		resource value
	$\mid$	$\sigma(spine\_elem)$	M	substitution for spine elements / return values
$spine$	$::=$			spine
	$\mid$	$\overline{spine\_elem_i}^{\,i}$		
$tval$	$::=$			(effectful) top-level values
	$\mid$	$\mathtt{done}\ spine$		end of top-level expression
	$\mid$	$\mathtt{undef}\ UB\_name$		undefined behaviour
$res\_pattern$	$::=$			resource terms
	$\mid$	$\mathtt{emp}$	binders $= \{\}$	empty heap
	$\mid$	$ident$	binders $= ident$	variable
	$\mid$	$\mathtt{fold}\,(res\_pattern)$	binders $= \{\}$	unfold (recursive) predicate
	$\mid$	$\langle res\_pattern_1, res\_pattern_2\rangle$	binders $=$ binders$(res_pattern_1) \cup$ binders$(res_pattern_2)$	seperating-conjunction pair
	$\mid$	$\mathtt{pack}\,(ident, res\_pattern)$	binders $= ident \cup$ binders$(res_pattern)$	packing for existentials
$ret\_pattern$	$::=$			return pattern
	$\mid$	$\mathtt{comp}\,ident\_or\_pattern$	binders $=$ binders$(ident_or_pattern)$	computational variable
	$\mid$	$\mathtt{log}\,ident$	binders $= ident$	logical variable
	$\mid$	$\mathtt{res}\,res\_pattern$	binders $=$ binders$(res_pattern)$	resource variable
$init,$	$::=$			initialisation status
	$\mid$	$\checkmark$		initialised
	$\mid$	$\times$		uninitalised
$points\_to,\ pt$	$::=$			points-to separation logic predicate
	$\mid$	$term_1 \overset{init}{\mapsto}_{\tau} term_2$		
res	$::=$			resources

	|	$\texttt{emp}$		empty heap
	|	$\mathit{points\_to}$		points-top heap pred.
	|	$\mathit{res}_1 * \mathit{res}_2$		seperating conjunction
	|	$\exists\, \mathit{ident}{:}\beta.\ \mathit{res}$		existential
	|	$\mathit{term} \wedge \mathit{res}$		logical conjuction
	|	$\texttt{if}\, \mathit{term}\, \texttt{then}\, \mathit{res}_1\, \texttt{else}\, \mathit{res}_2$		ordered disjuction
	|	$\alpha(\,\overline{\mathit{pval}_i}^{\,i}\,)$		predicate
	|	$\sigma(\mathit{res})$	M	simul-sub σ in res
ret, _	::=			return types
	|	$\Sigma\, \mathit{ident}{:}\beta.\ \mathit{ret}$		return a computational value
	|	$\exists\, \mathit{ident}{:}\beta.\ \mathit{ret}$		return a logical value
	|	$\mathit{res} \otimes \mathit{ret}$		return a resource value
	|	$\mathit{term} \wedge \mathit{ret}$		return a predicate (post-condition)
	|	$\texttt{I}$		end return list
	|	$\sigma(\mathit{ret})$	M	simul-sub σ in ret
$\mathit{seq\_expr}$	::=			sequential (effectful) expressions
	|	$\texttt{ccall}\,(\tau, \mathit{ident}, \mathit{spine})$		C function call
	|	$\texttt{pcall}\,(\mathit{name}, \mathit{spine})$		procedure call
$\mathit{seq\_texpr}$	::=			sequential top-level (effectful) expressions
	|	tval		(effectful) top-level values
	|	$\texttt{run}\, \mathit{ident}\, \overline{\mathit{pval}_i}^{\,i}$		run from label
	|	$\texttt{let}\, \mathit{ident\_or\_pattern} = \mathit{pexpr}\, \texttt{in}\, \mathit{texpr}$	bind binders($\mathit{ident\_or\_pattern}$) in texpr	pure let
	|	$\texttt{let}\, \mathit{ident\_or\_pattern}{:}(y_1{:}\beta_1.\ \mathit{term}_1) = \mathit{tpexpr}\, \texttt{in}\, \mathit{texpr}$	bind binders($\mathit{ident\_or\_pattern}$) in texpr bind y_1 in term_1	annotated pure let
	|	$\texttt{let}\, \overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{seq\_expr}\, \texttt{in}\, \mathit{texpr}$	bind binders($\overline{\mathit{ret\_pattern}_i}^{\,i}$) in texpr	bind return patterns
	|	$\texttt{let}\, \overline{\mathit{ret\_pattern}_i}^{\,i}{:}\mathit{ret} = \mathit{texpr}_1\, \texttt{in}\, \mathit{texpr}_2$	bind binders($\overline{\mathit{ret\_pattern}_i}^{\,i}$) in texpr_2	annotated bind return patterns
	|	$\texttt{case}\, \mathit{pval}\, \texttt{of}\, \overline{\mid \mathit{texpr\_case\_branch}_i}^{\,i}\, \texttt{end}$		pattern matching
	|	$\texttt{if}\, \mathit{pval}\, \texttt{then}\, \mathit{texpr}_1\, \texttt{else}\, \mathit{texpr}_2$		conditional

	$\mid$	$\texttt{bound}\,[\mathit{int}](\mathit{is\_texpr})$		limit scope of indet seq behaviour, absent at ru
$\mathit{texpr\_case\_branch}$	::=			top-level case expression branch
	$\mid$	$\mathit{pattern} \Rightarrow \mathit{texpr}$	bind binders($\mathit{pattern}$) in texpr	top-level case expression branch
$\mathit{is\_expr}$	::=			indet seq (effectful) expressions
	$\mid$	tval		(effectful) top-level values
	$\mid$	$\texttt{memop}\,(\mathit{mem\_op})$		pointer op involving memory
	$\mid$	$\mathit{pol\_mem\_action}$		memory action
$\mathit{is\_texpr}$	::=			indet seq top-level (effectful) expressions
	$\mid$	$\texttt{let}\,\texttt{weak}\,\overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{is\_expr}\,\texttt{in}\,\mathit{texpr}$	bind binders($\overline{\mathit{ret\_pattern}_i}^{\,i}$) in texpr	weak sequencing
	$\mid$	$\texttt{let}\,\texttt{strong}\,\overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{is\_expr}\,\texttt{in}\,\mathit{texpr}$	bind binders($\overline{\mathit{ret\_pattern}_i}^{\,i}$) in texpr	strong sequencing
texpr	::=			top-level (effectful) expressions
	$\mid$	$\mathit{seq\_texpr}$		sequential (effectful) expressions
	$\mid$	$\mathit{is\_texpr}$		indet seq (effectful) expressions
	$\mid$	$\sigma(\mathit{texpr})$	M	simul-sub σ in texpr
arg	::=			argument/function types
	$\mid$	$\Pi\,\mathit{ident}{:}\beta.\ \mathit{arg}$		
	$\mid$	$\forall\,\mathit{ident}{:}\beta.\ \mathit{arg}$		
	$\mid$	$\mathit{res} \multimap \mathit{arg}$		
	$\mid$	$\mathit{term} \supset \mathit{arg}$		
	$\mid$	ret		
	$\mid$	$\sigma(\mathit{arg})$	M	simul-sub σ in arg
$\mathit{pure\_arg}$	::=			pure argument/function types
	$\mid$	$\Pi\,\mathit{ident}{:}\beta.\ \mathit{pure\_arg}$		
	$\mid$	$\mathit{term} \supset \mathit{pure\_arg}$		
	$\mid$	$\mathit{pure\_ret}$		

$pure\_ret$	$::=$			pure return types
	$\mid$	$\Sigma\, ident{:}\beta.\ pure\_ret$		
	$\mid$	$term \wedge pure\_ret$		
	$\mid$	$\mathtt{I}$		
$\mathcal{C}$	$::=$			computational var env
	$\mid$	$\cdot$		
	$\mid$	$\mathcal{C}, ident{:}\beta$		
	$\mid$	$\overline{\mathcal{C}_i}^{\,i}$		
$\mathcal{L}$	$::=$			logical var env
	$\mid$	$\cdot$		
	$\mid$	$\overline{\mathcal{L}_i}^{\,i}$		
	$\mid$	$\mathcal{L}, ident{:}\beta$		
Φ	$::=$			constraints env
	$\mid$	$\cdot$		
	$\mid$	$\Phi, term$		
	$\mid$	$\overline{\Phi_i}^{\,i}$		
$\mathcal{R}$	$::=$			resources env
	$\mid$	$\cdot$		
	$\mid$	$\mathcal{R}, ident{:}res$		
	$\mid$	$\overline{\mathcal{R}_i}^{\,i}$		
$\sigma,\ \psi$	$::=$			substitutions
	$\mid$	$\cdot$		
	$\mid$	$spine\_elem/ident, \sigma$		
	$\mid$	$term/ident, \sigma$		
	$\mid$	$\overline{\sigma_i}^{\,i}$		
	$\mid$	$\sigma(\psi)$	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
opsem	::=		
	|	$\forall\, i < j.\ \texttt{not}\,(\mathit{pattern}_i = \mathit{pval} \rightsquigarrow \sigma_i)$	
	|	$\texttt{fresh}\,(\mathit{mem\_ptr})$	
	|	term	
	|	$\mathit{pval}{:}\beta$	
formula	::=		
	|	$\mathit{judgement}$	
	|	typing	
	|	opsem	
	|	$\mathit{term} \,\equiv\, \mathit{term}'$	
	|	$\mathit{name}{:}\mathit{pure\_arg} \,\equiv\, \overline{x_i}^{\,i} \,\mapsto\, \mathit{tpexpr} \,\in\, \texttt{Globals}$	
	|	$\mathit{name}{:}\mathit{arg} \,\equiv\, \overline{x_i}^{\,i} \,\mapsto\, \mathit{texpr} \,\in\, \texttt{Globals}$	
heap, *h*, *f*	::=		heaps
	|	$\cdot$	
	|	$h + \{\mathit{points\_to}\}$	
	|	$h + f$	[O]
lemma_jtype	::=		
	|	$\overline{x_i}^{\,i} :: \mathit{arg} \rightsquigarrow \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \mid \mathit{ret}$	
	|	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$	
	|	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$	

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

$\mathit{tpexpr\_jtype}$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpexpr} \Leftarrow \mathit{ident}{:}\beta.\ \mathit{term}$
$\mathit{action\_jtype}$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_action} \Rightarrow \mathit{ret}$
$\mathit{memop\_jtype}$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_op} \Rightarrow \mathit{ret}$
$\mathit{tval\_jtype}$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{tval} \Leftarrow \mathit{ret}$
$\mathit{seq\_expr\_jtype}$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_expr} \Rightarrow \mathit{ret}$
$\mathit{is\_expr\_jtype}$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_expr} \Rightarrow \mathit{ret}$
$\mathit{texpr\_jtype}$	$::=$	
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret}$
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}$
	$\vert$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{texpr} \Leftarrow \mathit{ret}$
$\mathit{subs\_jtype}$	$::=$	
	$\vert$	$\mathit{pattern} = \mathit{pval} \leadsto \sigma$
	$\vert$	$\mathit{ident\_or\_pattern} = \mathit{pval} \leadsto \sigma$
	$\vert$	$\mathit{res\_pattern} = \mathit{res\_term} \leadsto \sigma$
	$\vert$	$\overline{\mathit{ret\_pattern}_i = \mathit{spine\_elem}_i}^{i} \leadsto \sigma$
	$\vert$	$\overline{x_i = \mathit{spine\_elem}_i}^{i} :: \mathit{arg} \gg \sigma; \mathit{ret}$
$\mathit{pure\_opsem\_jtype}$	$::=$	

$$
\begin{array}{lll}
 & | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{pexpr}' \rangle \\
 & | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\, \mathit{term}) \rangle \\
 & | & \langle \mathit{tpexpr} \rangle \longrightarrow \langle \mathit{tpexpr}' \rangle \\
 & & \\
\mathit{opsem\_jtype} & ::= & \\
 & | & \langle h; \mathit{seq\_expr} \rangle \longrightarrow \langle h'; \mathit{texpr}{:}\mathit{ret} \rangle \\
 & | & \langle h; \mathit{seq\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
 & | & \langle h; \mathit{mem\_op} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
 & | & \langle h; \mathit{mem\_action} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
 & | & \langle h; \mathit{is\_expr} \rangle \longrightarrow \langle h'; \mathit{is\_expr}' \rangle \\
 & | & \langle h; \mathit{is\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
 & | & \langle h; \mathit{texpr} \rangle \longrightarrow \langle h'; \mathit{texpr}' \rangle
\end{array}
$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{loaded\_value}_i \Rightarrow \beta}^{i}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{array}\,(\,\overline{\mathit{loaded\_value}_i}^{i}\,) \Rightarrow \mathtt{obj}\,\mathtt{array}\,\beta} \quad \text{TY\_PVAL\_OBJ\_ARR}$$

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

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

$$\frac{x{:}\beta \in \mathcal{C}}{\mathcal{C};\mathcal{L};\Phi \vdash x \Rightarrow \beta} \quad \text{TY\_PVAL\_VAR\_COMP}$$

$$\frac{x{:}\beta \in \mathcal{L}}{\mathcal{C};\mathcal{L};\Phi \vdash x \Rightarrow \beta} \quad \text{TY\_PVAL\_VAR\_LOG}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \mathtt{obj}\,\beta}{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \beta} \quad \text{TY\_PVAL\_OBJ}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi \vdash \mathit{object\_value} \Rightarrow \mathtt{obj}\,\beta}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{specified}\,\mathit{object\_value} \Rightarrow \beta} \quad \text{TY\_PVAL\_LOADED}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{Unit} \Rightarrow \mathtt{unit}} \quad \text{TY\_PVAL\_UNIT}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{True} \Rightarrow \mathtt{bool}} \quad \text{TY\_PVAL\_TRUE}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{False} \Rightarrow \mathtt{bool}} \quad \text{TY\_PVAL\_FALSE}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash value_i \Rightarrow \beta}^{i}}{\mathcal{C};\mathcal{L};\Phi \vdash \beta[\,\overline{value_i}^{i}\,] \Rightarrow \mathtt{list}\,\beta} \quad \text{TY\_PVAL\_LIST}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash value_i \Rightarrow \beta_i}^{i}}{\mathcal{C};\mathcal{L};\Phi \vdash (\,\overline{value_i}^{i}\,) \Rightarrow \overline{\beta_i}^{i}} \quad \text{TY\_PVAL\_TUPLE}$$

$$\frac{\mathtt{smt}\,(\Phi \Rightarrow \mathtt{false})}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{error}\,(string, pval) \Rightarrow \beta} \quad \text{TY\_PVAL\_ERROR}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{Nil}\,\beta(\,) \Rightarrow \mathtt{list}\,\beta} \quad \text{TY\_PVAL\_CTOR\_NIL}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi \vdash pval_1 \Rightarrow \beta \\ \mathcal{C};\mathcal{L};\Phi \vdash pval_2 \Rightarrow \mathtt{list}\,\beta\end{array}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{Cons}(pval_1, pval_2) \Rightarrow \mathtt{list}\,\beta} \quad \text{TY\_PVAL\_CTOR\_CONS}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash pval_i \Rightarrow \beta_i}^{i}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{Tuple}(\,\overline{pval_i}^{i}\,) \Rightarrow \overline{\beta_i}^{i}} \quad \text{TY\_PVAL\_CTOR\_TUPLE}$$

$$\frac{\overline{\mathcal{C};\mathcal{L};\Phi \vdash pval_i \Rightarrow \beta}^{i}}{\mathcal{C};\mathcal{L};\Phi \vdash \mathtt{Array}(\,\overline{pval_i}^{i}\,) \Rightarrow \mathtt{array}\,\beta} \quad \text{TY\_PVAL\_CTOR\_ARRAY}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\Phi \vdash \mathit{res\_pattern}{:}\mathit{res} \rightsquigarrow \mathcal{L}_1; \Phi_1; \mathcal{R}_1 \qquad \Phi \vdash \overline{\mathit{ret\_pattern}_i}^{\,i} {:}\mathit{ret} \rightsquigarrow \mathcal{C}_2; \mathcal{L}_2; \Phi_2; \mathcal{R}_2}{\Phi \vdash \texttt{res}\,\mathit{res\_pattern},\ \overline{\mathit{ret\_pattern}_i}^{\,i} {:}\mathit{res} \otimes \mathit{ret} \rightsquigarrow \mathcal{C}_2; \mathcal{L}_1, \mathcal{L}_2; \Phi_1, \Phi_2; \mathcal{R}_1, \mathcal{R}_2} \quad \textsc{Ty\_Pat\_Ret\_Res}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi \vdash pval \Rightarrow \texttt{bool}\\ \mathcal{C};\mathcal{L};\Phi, pval = \texttt{true};\mathcal{R} \vdash texpr_1 \Leftarrow ret\\ \mathcal{C};\mathcal{L};\Phi, pval = \texttt{false};\mathcal{R} \vdash texpr_2 \Leftarrow ret\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{if}\, pval\, \texttt{then}\, texpr_1\, \texttt{else}\, texpr_2 \Leftarrow ret} \quad \text{TY\_SEQ\_TE\_IF}$$

$$\frac{\begin{array}{l}ident{:}arg \equiv \overline{x_i}^{\,i} \mapsto texpr \in \texttt{Globals}\\ \mathcal{C};\mathcal{L};\Phi;\cdot \vdash \overline{x_i = pval_i}^{\,i} :: arg \gg \sigma;\texttt{false} \wedge \texttt{I}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\cdot \vdash \texttt{run}\, ident\, \overline{pval_i}^{\,i} \Leftarrow \texttt{false} \wedge \texttt{I}} \quad \text{TY\_SEQ\_TE\_RUN}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash is\_texpr \Leftarrow ret}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \texttt{bound}\,[int](is\_texpr) \Leftarrow ret} \quad \text{TY\_SEQ\_TE\_BOUND}$$

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

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}' \vdash is\_expr \Rightarrow ret_1\\ \Phi \vdash \overline{ret\_pattern_i}^{\,i} {:} ret_1 \leadsto \mathcal{C}_1;\mathcal{L}_1;\Phi_1;\mathcal{R}_1\\ \mathcal{C},\mathcal{C}_1;\mathcal{L},\mathcal{L}_1;\Phi,\Phi_1;\mathcal{R},\mathcal{R}_1 \vdash texpr \Leftarrow ret_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}',\mathcal{R} \vdash \texttt{let strong}\, \overline{ret\_pattern_i}^{\,i} = is\_expr\, \texttt{in}\, texpr \Leftarrow ret_2} \quad \text{TY\_IS\_TE\_LETS}$$

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

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash is\_texpr \Leftarrow ret}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash is\_texpr \Leftarrow ret} \quad \text{TY\_TE\_IS}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash seq\_texpr \Leftarrow ret}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash seq\_texpr \Leftarrow ret} \quad \text{TY\_TE\_SEQ}$$

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

$$\frac{}{\_{:}\_ = pval \leadsto \cdot} \quad \text{SUBS\_DECONS\_VALUE\_NO\_SYM\_ANNOT}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Definition rules:        213 good    0 bad
Definition rule clauses: 476 good    0 bad
```