ident, x, y, y_p, y_f, $\_$, abbrev, r	subscripts: p for pointers, f for functions
n, i, j	index variables
$\mathit{impl\_const}$	implementation-defined constant
$\mathit{mem\_int}$	memory integer value
member	C struct/union member name
	Ott-hack, ignore (annotations)
nat	OCaml arbitrary-width natural number
$\mathit{mem\_ptr}$	abstract pointer value
$\mathit{mem\_val}$	abstract memory value
	Ott-hack, ignore (locations)
$\mathit{mem\_iv\_c}$	OCaml type for memory constraints on integer values
$\mathit{UB\_name}$	undefined behaviour
string	OCaml string
	Ott-hack, ignore (OCaml type variable TY)
	Ott-hack, ignore (OCaml Symbol.prefix)
$\mathit{mem\_order}$, $\_$	OCaml type for memory order
$\mathit{linux\_mem\_order}$	OCaml type for Linux memory order
$\mathit{logical\_val}$	logical values (to be specified)
	Ott-hack, ignore (OCaml type variable bt)

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

	|	`>`	greater than
	|	`<`	less than
	|	`>=`	greater than or equal to
	|	`<=`	less than or equal to
	|	`/\`	conjucntion
	|	`\/`	disjunction
$binop_{arith}$	::=		arithmentic binary operators
	|	`+`	
	|	`-`	
	|	`*`	
	|	`/`	
	|	`rem_t`	
	|	`rem_f`	
	|	`^`	
$binop_{rel}$	::=		relational binary operators
	|	`=`	
	|	`>`	
	|	`<`	
	|	`>=`	
	|	`<=`	
$binop_{bool}$	::=		boolean binary operators
	|	`/\`	
	|	`\/`	
object_value	::=		C object values (inhabitants of object types), which can be read/stored
	|	*mem_int*	integer value
	|	*mem_ptr*	pointer value

	|	$\texttt{array}\,(\,\overline{\mathit{loaded\_value}_i}^{\,i}\,)$	C array value
	|	$(\,\texttt{struct}\,\mathit{ident})\{\,\overline{.\,\mathit{member}_i{:}\tau_i = \mathit{mem\_val}_i}^{\,i}\,\}$	C struct value
	|	$(\,\texttt{union}\,\mathit{ident})\{\,.\,\mathit{member} = \mathit{mem\_val}\}$	C union value
$\mathit{loaded\_value}$	::=		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
	|	`Unit`	unit
	|	`True`	boolean true
	|	`False`	boolean false
	|	$\beta[\,\overline{\mathit{value}_i}^{\,i}\,]$	list
	|	$(\,\overline{\mathit{value}_i}^{\,i}\,)$	tuple
$\mathit{ctor\_val}$	::=		data constructors
	|	$\texttt{Nil}\,\beta$	empty list
	|	`Cons`	list cons
	|	`Tuple`	tuple
	|	`Array`	C array
	|	`Specified`	non-unspecified loaded value
$\mathit{ctor\_expr}$	::=		data constructors
	|	`Ivmax`	max integer value
	|	`Ivmin`	min integer value
	|	`Ivsizeof`	sizeof value
	|	`Ivalignof`	alignof value
	|	`IvCOMPL`	bitwise complement
	|	`IvAND`	bitwise AND

	|	$\mathtt{IvOR}$	bitwise OR
	|	$\mathtt{IvXOR}$	bitwise XOR
	|	$\mathtt{Fvfromint}$	cast integer to floating value
	|	$\mathtt{Ivfromfloat}$	cast floating to integer value
name	::=		
	|	ident	Core identifier
	|	$\mathit{impl\_const}$	implementation-defined constant
pval	::=		pure values
	|	ident	Core identifier
	|	$\mathit{impl\_const}$	implementation-defined constant
	|	value	Core values
	|	$\mathtt{constrained}\,(\,\overline{\mathit{mem\_iv\_c}_i, \mathit{pval}_i}^{\,i}\,)$	constrained value
	|	$\mathtt{error}\,(\mathit{string}, \mathit{pval})$	impl-defined static error
	|	$\mathit{ctor\_val}(\,\overline{\mathit{pval}_i}^{\,i}\,)$	data constructor application
	|	$(\,\mathtt{struct}\,\mathit{ident})\{\,\overline{.\,\mathit{member}_i = \mathit{pval}_i}^{\,i}\,\}$	C struct expression
	|	$(\,\mathtt{union}\,\mathit{ident})\{\,.\,\mathit{member} = \mathit{pval}\}$	C union expression
pexpr	::=		pure expressions
	|	pval	pure values
	|	$\mathit{ctor\_expr}(\,\overline{\mathit{pval}_i}^{\,i}\,)$	data constructor application
	|	$\mathtt{array\_shift}\,(\mathit{pval}_1, \tau, \mathit{pval}_2)$	pointer array shift
	|	$\mathtt{member\_shift}\,(\mathit{pval}, \mathit{ident}, \mathit{member})$	pointer struct/union member shift
	|	$\mathtt{not}\,(\mathit{pval})$	boolean not
	|	$\mathit{pval}_1\ \mathit{binop}\ \mathit{pval}_2$	binary operations
	|	$\mathtt{memberof}\,(\mathit{ident}, \mathit{member}, \mathit{pval})$	C struct/union member access
	|	$\mathit{name}(\,\overline{\mathit{pval}_i}^{\,i}\,)$	pure function call
	|	$\mathtt{assert\_undef}\,(\mathit{pval},\ \mathit{UB\_name})$	
	|	$\mathtt{bool\_to\_integer}\,(\mathit{pval})$	

	|	$\texttt{conv\_int}\,(\tau, \mathit{pval})$		
	|	$\texttt{wrapI}\,(\tau, \mathit{pval})$		
tpval	::=			top-level pure values
	|	$\texttt{undef}\ \mathit{UB\_name}$		undefined behaviour
	|	$\texttt{done}\,\mathit{pval}$		pure done
$\mathit{ident\_opt\_\beta}$	::=			type annotated optional identifier
	|	$\_{:}\beta$		
	|	$\mathit{ident}{:}\beta$		
$\mathit{pattern}$	::=			
	|	$\mathit{ident\_opt\_\beta}$		
	|	$\mathit{ctor\_val}(\,\overline{\mathit{pattern}_i}^{\,i}\,)$		
$\mathit{ident\_or\_pattern}$	::=			
	|	ident		
	|	$\mathit{pattern}$		
tpexpr	::=			top-level pure expressions
	|	tpval		top-level pure values
	|	$\texttt{case}\,\mathit{pval}\,\texttt{of}\,\overline{\mid \mathit{pattern}_i \Rightarrow \mathit{tpexpr}_i}^{\,i}\,\texttt{end}$		pattern matching
	|	$\texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}\,\texttt{in}\,\mathit{tpexpr}$		pure let
	|	$\texttt{if}\,\mathit{pval}\,\texttt{then}\,\mathit{tpexpr}_1\,\texttt{else}\,\mathit{tpexpr}_2$		pure if
	|	$[\mathcal{C}/\mathcal{C}']\mathit{tpexpr}$	M	simul-sub all vars in $\mathcal{C}$ for all vars in $\mathcal{C}'$ in tpexpr
$\mathit{m\_kill\_kind}$	::=			
	|	$\texttt{dynamic}$		
	|	$\texttt{static}\,\tau$		

bool, _	::=		OCaml booleans
	|	`true`	
	|	`false`	
int, _	::=		OCaml fixed-width integer
	|	i	literal integer
$\mathit{mem\_action}$	::=		memory actions
	|	`create` (pval, τ)	
	|	`create_readonly` $(\mathit{pval}_1, \tau, \mathit{pval}_2)$	
	|	`alloc` $(\mathit{pval}_1, \mathit{pval}_2)$	
	|	`kill` $(\mathit{m\_kill\_kind}, \mathit{pval})$	
	|	`store` $(\mathit{bool}, \tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{mem\_order})$	true means store is locking
	|	`load` $(\tau, \mathit{pval}, \mathit{mem\_order})$	
	|	`rmw` $(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3, \mathit{mem\_order}_1, \mathit{mem\_order}_2)$	
	|	`fence` $(\mathit{mem\_order})$	
	|	`cmp_exch_strong` $(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3, \mathit{mem\_order}_1, \mathit{mem\_order}_2)$	
	|	`cmp_exch_weak` $(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3, \mathit{mem\_order}_1, \mathit{mem\_order}_2)$	
	|	`linux_fence` $(\mathit{linux\_mem\_order})$	
	|	`linux_load` $(\tau, \mathit{pval}, \mathit{linux\_mem\_order})$	
	|	`linux_store` $(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{linux\_mem\_order})$	
	|	`linux_rmw` $(\tau, \mathit{pval}_1, \mathit{pval}_2, \mathit{linux\_mem\_order})$	
$\mathit{polarity}$	::=		polarities for memory actions
	|	`Pos`	sequenced by `let weak` and `let strong`
	|	`Neg`	only sequenced by `let strong`
$\mathit{pol\_mem\_action}$	::=		memory actions with polarity
	|	$\mathit{polarity}\ \mathit{mem\_action}$	

$\mathit{mem\_op}$	::=		operations involving the memory state
	|	$\mathit{pval}_1 \equiv \mathit{pval}_2$	pointer equality comparison
	|	$\mathit{pval}_1 \neq \mathit{pval}_2$	pointer inequality comparison
	|	$\mathit{pval}_1 < \mathit{pval}_2$	pointer less-than comparison
	|	$\mathit{pval}_1 > \mathit{pval}_2$	pointer greater-than comparison
	|	$\mathit{pval}_1 \leq \mathit{pval}_2$	pointer less-than comparison
	|	$\mathit{pval}_1 \geq \mathit{pval}_2$	pointer greater-than comparison
	|	$\mathit{pval}_1 -_{\tau} \mathit{pval}_2$	pointer subtraction
	|	$\texttt{intFromPtr}\,(\tau_1, \tau_2, \mathit{pval})$	cast of pointer value to integer value
	|	$\texttt{ptrFromInt}\,(\tau_1, \tau_2, \mathit{pval})$	cast of integer value to pointer value
	|	$\texttt{ptrValidForDeref}\,(\tau, \mathit{pval})$	dereferencing validity predicate
	|	$\texttt{ptrWellAligned}\,(\tau, \mathit{pval})$	
	|	$\texttt{ptrArrayShift}\,(\mathit{pval}_1, \tau, \mathit{pval}_2)$	
	|	$\texttt{memcpy}\,(\mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3)$	
	|	$\texttt{memcmp}\,(\mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3)$	
	|	$\texttt{realloc}\,(\mathit{pval}_1, \mathit{pval}_2, \mathit{pval}_3)$	
	|	$\texttt{va\_start}\,(\mathit{pval}_1, \mathit{pval}_2)$	
	|	$\texttt{va\_copy}\,(\mathit{pval})$	
	|	$\texttt{va\_arg}\,(\mathit{pval}, \tau)$	
	|	$\texttt{va\_end}\,(\mathit{pval})$	
$\mathit{res\_term}$	::=		resource terms
	|	$\texttt{emp}$	empty heap
	|	$\texttt{pt}$	single-cell heap
	|	ident	variable
	|	$\langle \mathit{res\_term}_1, \mathit{res\_term}_2 \rangle$	seperating-conjunction pair
	|	$\texttt{pack}\,(\mathit{pval}, \mathit{res\_term}_2)$	packing for existentials
$\mathit{spine\_elem}$	::=		spine element
	|	pval	pure value

	$\vert$	$\mathit{logical\_val}$		logical variable
	$\vert$	$\mathit{res\_term}$		resource valuel
tval	$::=$			(effectful) top-level values
	$\vert$	$\texttt{done}\,\overline{\mathit{spine\_elem}_i}^{\,i}$		end of top-level expression
	$\vert$	$\texttt{undef}\ \mathit{UB\_name}$		undefined behaviour
$\mathit{res\_pattern}$	$::=$			resource terms
	$\vert$	$\texttt{emp}$		empty heap
	$\vert$	$\texttt{pt}$		single-cell heap
	$\vert$	ident		variable
	$\vert$	$\langle \mathit{res\_pattern}_1, \mathit{res\_pattern}_2 \rangle$		seperating-conjunction pair
	$\vert$	$\texttt{pack}\,(\mathit{ident}, \mathit{res\_pattern})$		packing for existentials
$\mathit{ret\_pattern}$	$::=$			return pattern
	$\vert$	$\texttt{comp}\,\mathit{ident\_or\_pattern}$		computational variable
	$\vert$	$\texttt{log}\,\mathit{ident}$		logical variable
	$\vert$	$\texttt{res}\,\mathit{res\_pattern}$		resource variable
$\mathit{bool\_op}$	$::=$			
	$\vert$	$\neg\,\mathit{term}$		
	$\vert$	$\mathit{term}_1 = \mathit{term}_2$		
	$\vert$	$\bigwedge(\,\overline{\mathit{term}_i}^{\,i}\,)$		
	$\vert$	$\bigvee(\,\overline{\mathit{term}_i}^{\,i}\,)$		
	$\vert$	$\mathit{term}_1\ \mathit{binop}_{\mathit{bool}}\ \mathit{term}_2$	M	
	$\vert$	$\texttt{if}\,\mathit{term}_1\,\texttt{then}\,\mathit{term}_2\,\texttt{else}\,\mathit{term}_3$		
$\mathit{arith\_op}$	$::=$			
	$\vert$	$\mathit{term}_1 + \mathit{term}_2$		
	$\vert$	$\mathit{term}_1 - \mathit{term}_2$		

	|	$term_1 \times term_2$		
	|	$term_1 / term_2$		
	|	$term_1$ `rem_t` $term_2$		
	|	$term_1$ `rem_f` $term_2$		
	|	$term_1$ `^` $term_2$		
	|	$term_1\ binop_{arith}\ term_2$	M	
cmp_op	::=			
	|	$term_1 < term_2$		less than
	|	$term_1 \leq term_2$		less than or equal
	|	$term_1\ binop_{rel}\ term_2$	M	
$list\_op$	::=			
	|	`nil`		
	|	`tl` $term$		
	|	$term^{(int)}$		
$tuple\_op$	::=			
	|	$(\,\overline{term_i}^{\,i}\,)$		
	|	$term^{(int)}$		
$pointer\_op$	::=			
	|	mem_ptr		
	|	$term_1 +_{\mathrm{ptr}} term_2$		
$option\_op$	::=			
	|	`none` β		
	|	`some` $term$		
$array\_op$	::=			

	|	$\mathit{term}_1[\mathit{term}_2]$		
param_op	::=			
	|	$\mathit{ident}{:}\beta.\ \mathit{term}$		
	|	$\mathit{term}(\mathit{term}_1,\ ..,\mathit{term}_n)$		
struct_op	::=			
	|	$\mathit{term}.\mathit{member}$		
ct_pred	::=			
	|	`representable` (τ, term)		
	|	`alignedI` $(\mathit{term}_1, \mathit{term}_2)$		
term, _	::=			
	|	*lit*		
	|	*arith_op*		
	|	*bool_op*		
	|	*cmp_op*		
	|	*tuple_op*		
	|	*struct_op*		
	|	*pointer_op*		
	|	*list_op*		
	|	*array_op*		
	|	*ct_pred*		
	|	*option_op*		
	|	*param_op*		
	|	(term)	S	parentheses
	|	$[\mathit{term}_1/\mathit{ident}]\mathit{term}_2$	M	substitute term_1 for *ident* in term_2
	|	*pval*	M	only the ones which can be embeded into the SMT value grammar, so no array literals

init,	$::=$			initialisation status
	$\mid$	$\checkmark$		initialised
	$\mid$	$\times$		uninitalised
$\mathit{points\_to}$	$::=$			arbitrary predicate
	$\mid$	$\mathit{term}_1\, \mathbb{Q}\overset{\mathit{init}}{\mapsto}_{\tau}\, \mathit{term}_2$		
$\mathit{resource}$	$::=$			resources
	$\mid$	$\texttt{emp}$		empty heap
	$\mid$	$\mathit{points\_to}$		points-top heap pred.
	$\mid$	$\mathit{resource}_1 \star \mathit{resource}_2$		seperating conjunction
	$\mid$	$\exists\, \mathit{ident}{:}\beta.\ \mathit{resource}$		existential
	$\mid$	$\mathit{term} \wedge \mathit{resource}$		logical conjuction
	$\mid$	$\langle \mathit{resource} \rangle$	S	parentheses
	$\mid$	$[\mathit{pval}/\mathit{ident}]\mathit{resource}$	M	substitute *pval* for *ident* in *resource*
ret, _	$::=$			return types
	$\mid$	$\Sigma\, \mathit{ident}{:}\beta.\ \mathit{ret}$		
	$\mid$	$\exists\, \mathit{ident}{:}\beta.\ \mathit{ret}$		
	$\mid$	$\mathit{resource} \star \mathit{ret}$		
	$\mid$	$\mathit{term} \wedge \mathit{ret}$		
	$\mid$	$\texttt{I}$		
	$\mid$	$[\mathit{spine\_elem}/\mathit{ident}]\mathit{ret}$	M	
$\mathit{seq\_expr}$	$::=$			sequential (effectful) expressions
	$\mid$	pexpr		pure expressions
	$\mid$	$\texttt{ccall}\,(\tau, \mathit{pval},\ \overline{\mathit{spine\_elem}_i}^{\,i}\,)$		C function call
	$\mid$	$\texttt{pcall}\,(\mathit{name},\ \overline{\mathit{spine\_elem}_i}^{\,i}\,)$		procedure call
$\mathit{seq\_texpr}$	$::=$			sequential top-level (effectful) expressions

	$\mid$	tval		(effectful) top-level values
	$\mid$	$\texttt{run}\,\mathit{ident}\,\mathit{pval}_1, \,..\,, \mathit{pval}_n$		run from label
	$\mid$	$\texttt{nd}\,(\mathit{pval}_1, \,..\,, \mathit{pval}_n)$		nondeterministic choice
	$\mid$	$\texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i} = \mathit{seq\_expr}\,\texttt{in}\,\mathit{texpr}$		bind return patterns
	$\mid$	$\texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i} :\mathit{ret} = \mathit{texpr}_1\,\texttt{in}\,\mathit{texpr}_2$		annotated bind return patterns
	$\mid$	$\texttt{case}\,\mathit{pval}\,\texttt{of}\,\overline{\mid \mathit{pattern}_i \Rightarrow \mathit{texpr}_i}^{\,i}\,\texttt{end}$		pattern matching
	$\mid$	$\texttt{if}\,\mathit{pval}\,\texttt{then}\,\mathit{texpr}_1\,\texttt{else}\,\mathit{texpr}_2$		conditional
	$\mid$	$\texttt{bound}\,[\mathit{int}](\mathit{is\_texpr})$		limit scope of indet seq behaviour, absent at runtime
$\mathit{is\_expr}$	::=			indet seq (effectful) expressions
	$\mid$	$\texttt{memop}\,(\mathit{mem\_op})$		pointer op involving memory
	$\mid$	$\mathit{pol\_mem\_action}$		memory action
	$\mid$	$\texttt{unseq}\,(\mathit{texpr}_1, \,..\,, \mathit{texpr}_n)$		unsequenced expressions
$\mathit{is\_texpr}$	::=			indet seq top-level (effectful) expressions
	$\mid$	$\texttt{let}\,\texttt{weak}\,\mathit{pattern} = \mathit{is\_expr}\,\texttt{in}\,\texttt{mu\_texpr\_aux}$		weak sequencing
	$\mid$	$\texttt{let}\,\texttt{strong}\,\mathit{ident\_or\_pattern} = \mathit{is\_expr}\,\texttt{in}\,\texttt{mu\_texpr\_aux}$		strong sequencing
texpr	::=			top-level (effectful) expressions
	$\mid$	$\mathit{seq\_texpr}$		sequential (effectful) expressions
	$\mid$	$\mathit{is\_texpr}$		indet seq (effectful) expressions
	$\mid$	$[\mathcal{C}/\mathcal{C}']\mathit{texpr}$	M	simul-sub all vars in $\mathcal{C}$ for all vars in $\mathcal{C}'$ in texpr
$\mathit{terminals}$	::=			
	$\mid$	λ		
	$\mid$	$\longrightarrow$		
	$\mid$	$\rightarrow$		
	$\mid$	$\leadsto$		
	$\mid$	$\Rightarrow$		
	$\mid$	$\Leftarrow$		

$\mid \quad \vdash$
$\mid \quad \in$
$\mid \quad \Pi$
$\mid \quad \forall$
$\mid \quad \multimap$
$\mid \quad \supset$
$\mid \quad \Sigma$
$\mid \quad \exists$
$\mid \quad \star$
$\mid \quad \times$
$\mid \quad \wedge$
$\mid \quad \bigwedge$
$\mid \quad \neg$
$\mid \quad =$
$\mid \quad \neq$
$\mid \quad \leq$
$\mid \quad \geq$
$\mid \quad \&$
$\mid \quad \cdot$
$\mid \quad |$
$\mid \quad +_{\mathrm{ptr}}$
$\mid \quad \mapsto$
$\mid \quad *$
$\mid \quad ::$
$\mid \quad \checkmark$
$\mid \quad :$
$\mid \quad \cdot$
$\mid \quad \cdot$
$\mid \quad \gg$

	$\mid$	$::$		
	$\mid$	$\hat{}$		
	$\mid$	$\bigvee$		
	$\mid$	$\equiv$		
	$\mid$	$\langle$		
	$\mid$	$\rangle$		
z	$::=$			OCaml arbitrary-width integer
	$\mid$	i	M	literal integer
	$\mid$	$\mathrm{to\_int}(\mathit{mem\_int})$	M	
	$\mid$	$\mathrm{size\_of}(\tau)$	M	size of a C type
	$\mid$	$\mathrm{offset\_of}_{\mathit{tag}}(\mathit{member})$	M	offset of a struct member
	$\mid$	`ptr_size`	M	size of a pointer
	$\mid$	$\mathrm{max\_int}_{\tau}$	M	maximum value of int of type τ
	$\mid$	$\mathrm{min\_int}_{\tau}$	M	minimum value of int of type τ
$\mathbb{Q}$	$::=$			OCaml type for rational numbers
	$\mid$	$\frac{\mathit{int}_1}{\mathit{int}_2}$		
lit	$::=$			
	$\mid$	ident		
	$\mid$	`unit`		
	$\mid$	bool		
	$\mid$	z		
	$\mid$	$\mathbb{Q}$		
arg	$::=$			argument/function types
	$\mid$	$\Pi\, \mathit{ident}{:}\beta.\ \mathit{arg}$		
	$\mid$	$\forall\, \mathit{ident}{:}\beta.\ \mathit{arg}$		
	$\mid$	$\mathit{resource} \multimap \mathit{arg}$		

	$\mid$	$\mathit{term} \supset \mathit{arg}$		
	$\mid$	ret		
	$\mid$	$[\mathit{spine\_elem}/\mathit{ident}]\mathit{arg}$	M	
$\mathit{pure\_arg}$	$::=$			pure argument/function types
	$\mid$	$\Pi\,\mathit{ident}{:}\beta.\ \mathit{pure\_arg}$		
	$\mid$	$\mathit{term} \supset \mathit{pure\_arg}$		
	$\mid$	$\mathit{pure\_ret}$		
$\mathit{pure\_ret}$	$::=$			pure return types
	$\mid$	$\Sigma\,\mathit{ident}{:}\beta.\ \mathit{pure\_ret}$		
	$\mid$	$\mathit{term} \wedge \mathit{pure\_ret}$		
	$\mid$	$\mathtt{I}$		
$\mathcal{C}$	$::=$			computational var env
	$\mid$	$\cdot$		
	$\mid$	$\mathcal{C}, \mathit{ident}{:}\beta$		
	$\mid$	$\overline{\mathcal{C}_i}^{\,i}$		
	$\mid$	$\mathrm{fresh}(\mathcal{C})$	M	identical context except with fresh variable names
$\mathcal{L}$	$::=$			logical var env
	$\mid$	$\cdot$		
	$\mid$	$\overline{\mathcal{L}_i}^{\,i}$		
	$\mid$	$\mathcal{L}, \mathit{ident}{:}\beta$		
	$\mid$	$[\mathcal{C}/\mathcal{C}']\mathcal{L}$	M	
Φ	$::=$			constraints env
	$\mid$	$\cdot$		
	$\mid$	Φ, term		
	$\mid$	$\overline{\Phi_i}^{\,i}$		

| $[\mathcal{C}/\mathcal{C}']\Phi$ M

$\mathcal{R}$ ::= resources env
| $\cdot$
| $\mathcal{R}, \mathit{ident}{:}\mathit{resource}$
| $\mathcal{R}_1, \mathcal{R}_2$
| $[\mathcal{C}/\mathcal{C}']\mathcal{R}$ M

formula ::=
| $\mathit{judgement}$
| $\mathit{abbrev} \equiv \mathit{term}$
| $\texttt{smt}\,(\Phi \Rightarrow \mathit{term})$
| $\mathit{ident}{:}\beta \ \in \mathcal{C}$
| $\mathit{ident}{:}\,\texttt{struct}\,\mathit{tag}\ \&\,\overline{\mathit{member}_i{:}\tau_i}^{\,i} \ \in \texttt{Globals}$
| $\overline{\mathcal{C}_i; \mathcal{L}_i; \Phi_i \vdash \mathit{mem\_val}_i \Rightarrow \texttt{mem}\,\beta_i}^{\,i}$ dependent on memory object model
| $\overline{\mathcal{C}_i; \mathcal{L}_i; \Phi_i \vdash \mathit{pval}_i \Rightarrow \beta_i}^{\,i}$
| $\mathit{name}{:}\mathit{arg} \ \in \texttt{Globals}$
| $\overline{\mathit{term}_i\ \texttt{as}\,\mathit{pattern}_i{:}\beta_i \rightsquigarrow \mathcal{C}_i; \Phi_i}^{\,i}$
| $\overline{\mathcal{C}_i; \mathcal{L}_i; \Phi_i \vdash \mathit{tpexpr}_i \Leftarrow y_i{:}\beta_i.\ \mathit{term}_i}^{\,i}$
| $\overline{\mathcal{C}_i; \mathcal{L}_i; \Phi_i; \mathcal{R}_i \vdash \mathit{texpr}_i \Leftarrow \mathit{ret}_i}^{\,i}$
| $\mathcal{L} \vdash \mathit{logical\_val}{:}\beta$
| $\mathit{pval}{:}\mathit{arg} \ \in \texttt{Globals}$

object_value_jtype ::=
| $\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{object\_value} \Rightarrow \texttt{obj}\,\beta$

pval_jtype ::=
| $\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta$

resource_jtype ::=

$$\begin{array}{lll}
 & | & \mathit{resource} \equiv \mathit{resource}' \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{resource}
\end{array}$$

$$\begin{array}{lll}
\mathit{spine\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \overline{\mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \mathit{ret}
\end{array}$$

$$\begin{array}{lll}
\mathit{pexpr\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pexpr} \Rightarrow \mathit{ident}{:}\beta.\, \mathit{term}
\end{array}$$

$$\begin{array}{lll}
\mathit{pattern\_jtype} & ::= & \\
 & | & \mathit{term}\,\mathsf{as}\,\mathit{pattern}{:}\beta \leadsto \mathcal{C}; \Phi \\
 & | & \mathit{term}\,\mathsf{as}\,\mathit{ident\_or\_pattern}{:}\beta \leadsto \mathcal{C}; \Phi \\
 & | & \overline{\mathit{ret\_pattern}_i}^{\,i} {:}\mathit{ret} \leadsto \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \\
 & | & \mathit{res\_pattern}{:}\mathit{resource} \leadsto \mathcal{L}; \Phi; \mathcal{R}
\end{array}$$

$$\begin{array}{lll}
\mathit{tpval\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpval} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term}
\end{array}$$

$$\begin{array}{lll}
\mathit{tpexpr\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpexpr} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term}
\end{array}$$

$$\begin{array}{lll}
\mathit{action\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_action} \Rightarrow \mathit{ret}
\end{array}$$

$$\begin{array}{lll}
\mathit{tval\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{tval} \Leftarrow \mathit{ret}
\end{array}$$

$$\begin{array}{lll}
\mathit{texpr\_jtype} & ::= & \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_expr} \Rightarrow \mathit{ret} \\
 & | & \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_expr} \Rightarrow \mathit{ret}
\end{array}$$

	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret}$
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}$
	$\vert$	$\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \mathit{texpr} \Leftarrow \mathit{ret}$
$\mathit{judgement}$	$::=$	
	$\vert$	$\mathit{object\_value\_jtype}$
	$\vert$	$\mathit{pval\_jtype}$
	$\vert$	$\mathit{resource\_jtype}$
	$\vert$	$\mathit{spine\_jtype}$
	$\vert$	$\mathit{pexpr\_jtype}$
	$\vert$	$\mathit{pattern\_jtype}$
	$\vert$	$\mathit{tpval\_jtype}$
	$\vert$	$\mathit{tpexpr\_jtype}$
	$\vert$	$\mathit{action\_jtype}$
	$\vert$	$\mathit{tval\_jtype}$
	$\vert$	$\mathit{texpr\_jtype}$
$\mathit{user\_syntax}$	$::=$	
	$\vert$	ident
	$\vert$	n
	$\vert$	$\mathit{impl\_const}$
	$\vert$	$\mathit{mem\_int}$
	$\vert$	member
	$\vert$	
	$\vert$	nat
	$\vert$	$\mathit{mem\_ptr}$
	$\vert$	$\mathit{mem\_val}$
	$\vert$	
	$\vert$	$\mathit{mem\_iv\_c}$
	$\vert$	$\mathit{UB\_name}$

$$\begin{array}{ll}
| & \mathit{string} \\
| & \\
| & \\
| & \mathit{mem\_order} \\
| & \mathit{linux\_mem\_order} \\
| & \mathit{logical\_val} \\
| & \\
| & \mathit{Sctypes\_t} \\
| & \mathit{tag} \\
| & \beta \\
| & \mathit{binop} \\
| & \mathit{binop}_{\mathit{arith}} \\
| & \mathit{binop}_{\mathit{rel}} \\
| & \mathit{binop}_{\mathit{bool}} \\
| & \mathit{ident} \\
| & \tau \\
| & \mathit{ident} \\
| & \mathit{object\_value} \\
| & \mathit{loaded\_value} \\
| & \beta \\
| & \mathit{value} \\
| & \mathit{ctor\_val} \\
| & \mathit{ctor\_expr} \\
| & \tau \\
| & \mathit{name} \\
| & \mathit{pval} \\
| & \mathit{pval} \\
| & \mathit{pexpr} \\
| & \mathit{pexpr}
\end{array}$$

$| \quad \mathit{tpval}$

$| \quad \mathit{tpval}$

$| \quad \mathit{ident\_opt\_\beta}$

$| \quad \mathit{pattern}$

$| \quad \mathit{pattern}$

$| \quad \mathit{ident\_or\_pattern}$

$| \quad \mathit{tpexpr}$

$| \quad \mathit{tpexpr}$

$| \quad \mathit{m\_kill\_kind}$

$| \quad \mathit{bool}$

$| \quad \mathit{int}$

$| \quad \mathit{mem\_action}$

$| \quad \mathit{mem\_action}$

$| \quad \mathit{polarity}$

$| \quad \mathit{pol\_mem\_action}$

$| \quad \mathit{mem\_op}$

$| \quad \mathit{res\_term}$

$| \quad \mathit{spine\_elem}$

$| \quad \mathit{tval}$

$| \quad \mathit{tval}$

$| \quad \mathit{res\_pattern}$

$| \quad \mathit{ret\_pattern}$

$| \quad \mathit{bool\_op}$

$| \quad \mathit{arith\_op}$

$| \quad \mathit{cmp\_op}$

$| \quad \mathit{list\_op}$

$| \quad \mathit{tuple\_op}$

$| \quad \mathit{pointer\_op}$

$| \quad \beta$

$|\quad \mathit{option\_op}$

$|\quad \mathit{array\_op}$

$|\quad \mathit{param\_op}$

$|\quad \mathit{struct\_op}$

$|\quad \mathit{ct\_pred}$

$|\quad \mathit{term}$

$|\quad \mathit{term}$

$|\quad \mathit{init}$

$|\quad \mathit{points\_to}$

$|\quad \mathit{resource}$

$|\quad \mathit{ret}$

$|\quad \mathit{seq\_expr}$

$|\quad \mathit{seq\_expr}$

$|\quad \mathit{seq\_texpr}$

$|\quad \mathit{seq\_texpr}$

$|\quad \mathit{is\_expr}$

$|\quad \mathit{is\_expr}$

$|\quad \mathit{is\_texpr}$

$|\quad \mathit{is\_texpr}$

$|\quad \mathit{texpr}$

$|\quad \mathit{terminals}$

$|\quad z$

$|\quad \mathbb{Q}$

$|\quad \mathit{lit}$

$|\quad \mathit{arg}$

$|\quad \mathit{pure\_arg}$

$|\quad \mathit{pure\_ret}$

$|\quad \mathcal{C}$

$|\quad \mathcal{L}$

$|\quad \Phi$

$|\quad \mathcal{R}$

$|\quad \mathit{formula}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\overline{\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval}_i \Rightarrow \beta_i}^{\,i}}{\mathcal{C}; \mathcal{L}; \Phi \vdash (\,\texttt{struct}\,\mathit{tag})\{\,\overline{.\,\mathit{member}_i = \mathit{pval}_i}^{\,i}\,\} \Rightarrow \texttt{struct}\,\mathit{tag}} \quad \textsc{Pval\_Struct}$$

$\boxed{\mathit{resource} \equiv \mathit{resource}'}$

$$\frac{}{\texttt{emp} \equiv \texttt{emp}} \quad \textsc{Resource\_Eq\_Emp}$$

$$\frac{\texttt{smt}\,(\cdot \Rightarrow (\mathit{term}_1 = \mathit{term}'_1) \wedge (\mathit{term}_2 = \mathit{term}'_2))}{\mathit{term}_1\ \mathbb{Q}\overset{\mathit{init}}{\mapsto}_{\tau}\ \mathit{term}_2 \equiv \mathit{term}'_1\ \mathbb{Q}\overset{\mathit{init}}{\mapsto}_{\tau}\ \mathit{term}'_2} \quad \textsc{Resource\_Eq\_PointsTo}$$

$$\frac{\begin{array}{c}\mathit{resource}_1 \equiv \mathit{resource}'_1\\ \mathit{resource}_2 \equiv \mathit{resource}'_2\end{array}}{\mathit{resource}_1 \star \mathit{resource}_2 \equiv \mathit{resource}'_1 \star \mathit{resource}'_2} \quad \textsc{Resource\_Eq\_SepConj}$$

$$\frac{\mathit{resource} \equiv \mathit{resource}'}{\exists\,\mathit{ident}{:}\beta.\ \mathit{resource} \equiv \exists\,\mathit{ident}{:}\beta.\ \mathit{resource}'} \quad \textsc{Resource\_Eq\_Exists}$$

$$\frac{\begin{array}{c}\texttt{smt}\,(\cdot, \mathit{term} \Rightarrow \mathit{term}')\\ \texttt{smt}\,(\cdot, \mathit{term}' \Rightarrow \mathit{term})\\ \mathit{resource} \equiv \mathit{resource}'\end{array}}{\mathit{term} \wedge \mathit{resource} \equiv \mathit{term}' \wedge \mathit{resource}'} \quad \textsc{Resource\_Eq\_Term}$$

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

$$\frac{}{\mathcal{C};\mathcal{L};\Phi;\cdot\vdash \mathtt{emp} \Leftarrow \mathtt{emp}}\quad\textsc{Resource\_Emp}$$

$$\frac{}{\mathcal{C};\mathcal{L};\Phi;\cdot\vdash \mathtt{pt} \Leftarrow \mathit{points\_to}}\quad\textsc{Resource\_PointsTo}$$

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

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

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval} \Rightarrow \beta\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{res\_term}_2 \Leftarrow [\mathit{pval}/y]\mathit{resource}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathtt{pack}\,(\mathit{pval}, \mathit{res\_term}_2) \Leftarrow \exists\, y{:}\beta.\ \mathit{resource}}\quad\textsc{Resource\_Pack}$$

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

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

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval} \Rightarrow \beta\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash\ \overline{\mathit{spine\_elem}_i}^{\,i}\ :: [\mathit{pval}/x]\mathit{arg} \gg \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{pval},\ \overline{\mathit{spine\_elem}_i}^{\,i}\ :: \Pi\, x{:}\beta.\ \mathit{arg} \gg \mathit{ret}}\quad\textsc{Spine\_Computational}$$

$$\frac{\begin{array}{l}\mathcal{L}\vdash \mathit{logical\_val}{:}\beta\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash\ \overline{\mathit{spine\_elem}_i}^{\,i}\ :: [\mathit{logical\_val}/x]\mathit{arg} \gg \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{logical\_val},\ \overline{\mathit{spine\_elem}_i}^{\,i}\ :: \forall\, x{:}\beta.\ \mathit{arg} \gg \mathit{ret}}\quad\textsc{Spine\_Logical}$$

$$\frac{\begin{array}{c}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1 \vdash \mathit{res\_term} \Leftarrow \mathit{resource}\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R}_2 \vdash \overline{\mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}_1,\mathcal{R}_2 \vdash \mathit{res\_term},\ \overline{\mathit{spine\_elem}_i}^{\,i} :: \mathit{resource} \multimap \mathit{arg} \gg \mathit{ret}}\ \textsc{Spine\_Resource}$$

$$\frac{\begin{array}{l}\mathtt{smt}\,(\Phi \Rightarrow \mathit{term})\\ \mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{\mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R} \vdash \overline{\mathit{spine\_elem}_i}^{\,i} :: \mathit{term} \supset \mathit{arg} \gg \mathit{ret}}\ \textsc{Spine\_Constraint}$$

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

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

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

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

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

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

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

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

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

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

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

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

$$\boxed{\mathit{term}\,\mathtt{as}\,\mathit{pattern}{:}\beta \leadsto \mathcal{C};\Phi}$$

$$\frac{}{\mathit{term}\,\mathtt{as}\,\_{:}\beta{:}\beta \leadsto \cdot;\cdot}\quad \textsc{Comp\_Pattern\_No\_Sym\_Annot}$$

$$\frac{}{\mathit{term}\,\mathsf{as}\,x{:}\beta{:}\beta \leadsto \cdot, x{:}\beta; \cdot, x = \mathit{term}}\quad \textsc{Comp\_Pattern\_Sym\_Annot}$$

$$\frac{}{\mathit{term}\,\mathsf{as}\,\mathtt{Nil}\,\beta(\,){:}\mathtt{list}\,\beta \leadsto \cdot; \cdot}\quad \textsc{Comp\_Pattern\_Nil}$$

$$\frac{\begin{array}{c}\mathit{term}^{(1)}\,\mathsf{as}\,\mathit{pattern}_1{:}\beta \leadsto \mathcal{C}_1; \Phi_1\\ \mathtt{tl}\,\mathit{term}\,\mathsf{as}\,\mathit{pattern}_2{:}\mathtt{list}\,\beta \leadsto \mathcal{C}_2; \Phi_1\end{array}}{\mathit{term}\,\mathsf{as}\,\mathtt{Cons}(\mathit{pattern}_1, \mathit{pattern}_2){:}\mathtt{list}\,\beta \leadsto \mathcal{C}_1, \mathcal{C}_2; \Phi_1, \Phi_2}\quad \textsc{Comp\_Pattern\_Cons}$$

$$\frac{\overline{\mathit{term}^{(i)}\,\mathsf{as}\,\mathit{pattern}_i{:}\beta_i \leadsto \mathcal{C}_i; \Phi_i}^{\,i}}{\mathit{term}\,\mathsf{as}\,\mathtt{Tuple}(\,\overline{\mathit{pattern}_i}^{\,i}\,){:}\overline{\beta_i}^{\,i} \leadsto \overline{\mathcal{C}_i}^{\,i}; \overline{\Phi_i}^{\,i}}\quad \textsc{Comp\_Pattern\_Tuple}$$

$$\frac{\overline{\mathit{term}[i]\,\mathsf{as}\,\mathit{pattern}_i{:}\beta \leadsto \mathcal{C}_i; \Phi_i}^{\,i}}{\mathit{term}\,\mathsf{as}\,\mathtt{Array}(\,\overline{\mathit{pattern}_i}^{\,i}\,){:}\mathtt{array}\,\beta \leadsto \overline{\mathcal{C}_i}^{\,i}; \overline{\Phi_i}^{\,i}}\quad \textsc{Comp\_Pattern\_Array}$$

$$\frac{\mathit{term}\,\mathsf{as}\,\mathit{pattern}{:}\beta \leadsto \mathcal{C}; \Phi}{\mathit{term}\,\mathsf{as}\,\mathtt{Specified}(\mathit{pattern}){:}\beta \leadsto \mathcal{C}; \Phi}\quad \textsc{Comp\_Pattern\_Specified}$$

$$\boxed{\mathit{term}\,\mathsf{as}\,\mathit{ident\_or\_pattern}{:}\beta \leadsto \mathcal{C}; \Phi}$$

$$\frac{}{\mathit{term}\,\mathsf{as}\,x{:}\beta \leadsto \cdot, x{:}\beta; \cdot, x = \mathit{term}}\quad \textsc{Sym\_Or\_Pattern\_Sym}$$

$$\frac{\mathit{term}\,\mathsf{as}\,\mathit{pattern}{:}\beta \leadsto \mathcal{C}; \Phi}{\mathit{term}\,\mathsf{as}\,\mathit{pattern}{:}\beta \leadsto \mathcal{C}; \Phi}\quad \textsc{Sym\_Or\_Pattern\_Pattern}$$

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

$$\frac{}{:\mathtt{I} \rightsquigarrow \cdot;\cdot;\cdot;\cdot}\quad\textsc{Ret\_Pattern\_Empty}$$

$$\frac{\begin{array}{c} y\ \mathtt{as}\ \mathit{ident\_or\_pattern}{:}\beta \rightsquigarrow \mathcal{C}_1;\Phi_1 \\ \overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\mathit{ret} \rightsquigarrow \mathcal{C}_2;\mathcal{L}_2;\Phi_2;\mathcal{R}_2 \end{array}}{\mathtt{comp}\ \mathit{ident\_or\_pattern},\ \overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\Sigma\, y{:}\beta.\ \mathit{ret} \rightsquigarrow \mathcal{C}_1,\mathcal{C}_2;\mathcal{L}_2;\Phi_2;\mathcal{R}_2}\quad\textsc{Ret\_Pattern\_Computational}$$

$$\frac{\overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\mathit{ret} \rightsquigarrow \mathcal{C};\mathcal{L};\Phi;\mathcal{R}}{\mathtt{log}\ y,\ \overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\exists\, y{:}\beta.\ \mathit{ret} \rightsquigarrow \mathcal{C};\mathcal{L}, y{:}\beta;\Phi;\mathcal{R}}\quad\textsc{Ret\_Pattern\_Logical}$$

$$\frac{\begin{array}{c} \mathit{res\_pattern}{:}\mathit{resource} \rightsquigarrow \mathcal{L}_1;\Phi_1;\mathcal{R}_1 \\ \overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\mathit{ret} \rightsquigarrow \mathcal{C}_2;\mathcal{L}_2;\Phi_2;\mathcal{R}_2 \end{array}}{\mathtt{res}\ \mathit{res\_pattern},\ \overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\mathit{resource} \star \mathit{ret} \rightsquigarrow \mathcal{C}_2;\mathcal{L}_1,\mathcal{L}_2;\Phi_1,\Phi_2;\mathcal{R}_1,\mathcal{R}_2}\quad\textsc{Ret\_Pattern\_Resource}$$

$$\frac{\overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\mathit{ret} \rightsquigarrow \mathcal{C};\mathcal{L};\Phi;\mathcal{R}}{\overline{\mathit{ret\_pattern}_i}^{\,i}\ {:}\mathit{term} \wedge \mathit{ret} \rightsquigarrow \mathcal{C};\mathcal{L};\Phi,\mathit{term};\mathcal{R}}\quad\textsc{Ret\_Pattern\_Constraint}$$

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

$$\frac{}{\mathtt{emp}{:}\mathtt{emp} \rightsquigarrow \cdot;\cdot;\cdot}\quad\textsc{Res\_Pattern\_Empty}$$

$$\frac{}{\mathtt{pt}{:}\mathit{points\_to} \rightsquigarrow \cdot;\cdot;\cdot, r{:}\mathit{points\_to}}\quad\textsc{Res\_Pattern\_PointsTo}$$

$$\frac{}{r{:}\mathit{resource} \rightsquigarrow \cdot;\cdot;\cdot, r{:}\mathit{resource}}\quad\textsc{Res\_Pattern\_Var}$$

$$\frac{\begin{array}{c} \mathit{res\_pattern}_1{:}\mathit{resource}_1 \leadsto \mathcal{L}_1; \Phi_1; \mathcal{R}_1 \\ \mathit{res\_pattern}_2{:}\mathit{resource}_2 \leadsto \mathcal{L}_2; \Phi_2; \mathcal{R}_2 \end{array}}{\langle \mathit{res\_pattern}_1, \mathit{res\_pattern}_2 \rangle{:}\mathit{resource}_1 \star \mathit{resource}_2 \leadsto \mathcal{L}_1, \mathcal{L}_2; \Phi_1, \Phi_2; \mathcal{R}_1, \mathcal{R}_2} \quad \textsc{Res\_Pattern\_SepConj}$$

$$\frac{\mathit{res\_pattern}{:}\mathit{resource} \leadsto \mathcal{L}; \Phi; \mathcal{R}}{\mathit{res\_pattern}{:}\mathit{term} \wedge \mathit{resource} \leadsto \mathcal{L}; \Phi, \mathit{term}; \mathcal{R}} \quad \textsc{Res\_Pattern\_Conj}$$

$$\frac{\mathit{res\_pattern}{:}[x/y]\mathit{resource} \leadsto \mathcal{L}; \Phi; \mathcal{R}}{\texttt{pack}\,(x, \mathit{res\_pattern}){:}\exists\, y{:}\beta.\ \mathit{resource} \leadsto \mathcal{L}, x{:}\beta; \Phi; \mathcal{R}} \quad \textsc{Res\_Pattern\_Pack}$$

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

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

$$\frac{\begin{array}{c} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta \\ \texttt{smt}\,(\Phi \Rightarrow \mathit{term}) \end{array}}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{done}\,\mathit{pval} \Leftarrow y{:}\beta.\ \mathit{term}} \quad \textsc{TPVal\_Done}$$

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

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

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pexpr} \Rightarrow y_1{:}\beta_1.\ \mathit{term}_1 \\ y_1\,\texttt{as}\,\mathit{ident\_or\_pattern}{:}\beta_1 \leadsto \mathcal{C}_1; \Phi_1 \\ \mathcal{C}, \mathrm{fresh}(\mathcal{C}_1); \mathcal{L}, y_1{:}\beta_1; \Phi, \mathit{term}_1, [\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\Phi_1 \vdash [\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\mathit{tpexpr} \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2 \end{array}}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{let}\,\mathit{ident\_or\_pattern} = \mathit{pexpr}\,\texttt{in}\,\mathit{tpexpr} \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2} \quad \textsc{TPExpr\_Let}$$

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta_1 \\ \overline{y_1 \, \texttt{as} \, \mathit{pattern}_i{:}\beta_1 \rightsquigarrow \mathcal{C}_i; \Phi_i}^{i} \\ \overline{\mathcal{C}, \mathrm{fresh}(\mathcal{C}_i); \mathcal{L}, y_1{:}\beta_1; \Phi, y_1 = \mathit{pval}, [\mathrm{fresh}(\mathcal{C}_i)/\mathcal{C}_i]\Phi_i \vdash [\mathrm{fresh}(\mathcal{C}_i)/\mathcal{C}_i]\mathit{tpexpr}_i \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2}^{i} \end{array}}{\mathcal{C}; \mathcal{L}; \Phi \vdash \texttt{case} \, \mathit{pval} \, \texttt{of} \, \overline{\mid \mathit{pattern}_i \Rightarrow \mathit{tpexpr}_i}^{i} \, \texttt{end} \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2} \quad \textsc{TPExpr\_Case}$$

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

$$\frac{\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \texttt{integer}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot \vdash \texttt{create} \, (\mathit{pval}, \tau) \Rightarrow \Sigma \, y_p{:}\texttt{loc}.\ \exists \, y{:}\beta_\tau.\ \texttt{representable} \, (\tau{*}, y_p) \wedge \texttt{alignedI} \, (\mathit{pval}, y_p) \wedge \langle y_p \ 1 \overset{\times}{\mapsto}_\tau \ y \rangle \star \texttt{I}} \quad \textsc{Action\_Create}$$

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval}_1 \Rightarrow \texttt{loc} \\ \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval}_2 \Rightarrow \beta_\tau \\ \texttt{smt} \, (\Phi \Rightarrow \texttt{representable} \, (\tau, \mathit{pval}_2)) \\ \texttt{smt} \, (\Phi \Rightarrow \mathit{pval}_0 = \mathit{pval}_1) \end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot, r{:}\langle \mathit{pval}_0 \ 1 \mapsto_\tau \ \_ \rangle \vdash \texttt{store} \, (\_, \tau, \mathit{pval}_1, \mathit{pval}_2, \_) \Rightarrow \Sigma \, \_{:}\texttt{unit}.\ \mathit{pval}_0 \ 1 \overset{\checkmark}{\mapsto}_\tau \ \mathit{pval}_2 \star \texttt{I}} \quad \textsc{Action\_Store}$$

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval}_1 \Rightarrow \texttt{loc} \\ \texttt{smt} \, (\Phi \Rightarrow \mathit{pval}_0 = \mathit{pval}_1) \end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot, r{:}\langle \mathit{pval}_0 \ 1 \mapsto_\tau \ \_ \rangle \vdash \texttt{kill} \, (\texttt{static} \, \tau, \mathit{pval}_1) \Rightarrow \Sigma \, \_{:}\texttt{unit}.\ \texttt{I}} \quad \textsc{Action\_Kill\_Static}$$

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

$$\frac{}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{done} \ \Leftarrow \texttt{I}} \quad \textsc{TVal\_I}$$

$$\frac{\begin{array}{l} \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{done} \, \overline{\mathit{spine\_elem}_i}^{i} \Leftarrow [\mathit{pval}/y]\mathit{ret} \end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{done} \, \mathit{pval}, \ \overline{\mathit{spine\_elem}_i}^{i} \Leftarrow \Sigma \, y{:}\beta.\ \mathit{ret}} \quad \textsc{TVal\_Computational}$$

$$\frac{\begin{array}{l}\mathcal{L} \vdash \mathit{logical\_val}{:}\beta \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{done}\ \overline{\mathit{spine\_elem}_i}^{\,i} \Leftarrow [\mathit{logical\_val}/y]\mathit{ret}\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{done}\ \mathit{logical\_val},\ \overline{\mathit{spine\_elem}_i}^{\,i} \Leftarrow \exists\, y{:}\beta.\ \mathit{ret}}\ \textsc{TVal\_Logical}$$

$$\frac{\begin{array}{l}\texttt{smt}\,(\Phi \Rightarrow \mathit{term}) \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{done}\ \overline{\mathit{spine\_elem}_i}^{\,i} \Leftarrow \mathit{ret}\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{done}\ \overline{\mathit{spine\_elem}_i}^{\,i} \Leftarrow \mathit{term} \wedge \mathit{ret}}\ \textsc{TVal\_Constraint}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1 \vdash \mathit{res\_term} \Leftarrow \mathit{resource} \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_2 \vdash \texttt{done}\ \overline{\mathit{spine\_elem}_i}^{\,i} \Leftarrow \mathit{ret}\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R}_2 \vdash \texttt{done}\ \mathit{res\_term},\ \overline{\mathit{spine\_elem}_i}^{\,i} \Leftarrow \mathit{resource} \star \mathit{ret}}\ \textsc{TVal\_Resource}$$

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

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

$$\frac{\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pexpr} \Rightarrow y{:}\beta.\ \mathit{term}}{\mathcal{C}; \mathcal{L}; \Phi; \cdot \vdash \mathit{pval} \Rightarrow \Sigma\, y{:}\beta.\ \mathit{term} \wedge \texttt{I}}\ \textsc{Seq\_Expr\_Pure}$$

$$\frac{\begin{array}{l}\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \texttt{loc} \\ \mathit{pval}{:}\mathit{arg}\ \in \texttt{Globals} \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash\ \overline{\mathit{spine\_elem}_i}^{\,i}\ :: \mathit{arg} \gg \mathit{ret}\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{ccall}\,(\tau, \mathit{pval},\ \overline{\mathit{spine\_elem}_i}^{\,i}\ ) \Rightarrow \mathit{ret}}\ \textsc{Seq\_Expr\_CCall}$$

$$\frac{\begin{array}{l}\mathit{name}{:}\mathit{arg}\ \in \texttt{Globals} \\ \mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash\ \overline{\mathit{spine\_elem}_i}^{\,i}\ :: \mathit{arg} \gg \mathit{ret}\end{array}}{\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \texttt{pcall}\,(\mathit{name},\ \overline{\mathit{spine\_elem}_i}^{\,i}\ ) \Rightarrow \mathit{ret}}\ \textsc{Seq\_Expr\_Proc}$$

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

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{mem\_action}\Rightarrow \mathit{ret}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \texttt{Pos}\,\mathit{mem\_action}\Rightarrow \mathit{ret}}\quad \textsc{IS\_Expr\_Action}$$

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{mem\_action}\Rightarrow \mathit{ret}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \texttt{Neg}\,\mathit{mem\_action}\Rightarrow \mathit{ret}}\quad \textsc{IS\_Expr\_Neg\_Action}$$

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

$$\frac{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{tval}\Leftarrow \mathit{ret}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \mathit{tval}\Leftarrow \mathit{ret}}\quad \textsc{Seq\_TExpr\_TVal}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}'\vdash \mathit{seq\_expr}\Rightarrow \mathit{ret}_1\\ \overline{\mathit{ret\_pattern}_i}^{\,i}:\mathit{ret}_1\rightsquigarrow \mathcal{C}_1;\mathcal{L}_1;\Phi_1;\mathcal{R}_1\\ \mathcal{C},\mathrm{fresh}(\mathcal{C}_1);\mathcal{L},\mathcal{L}_1;\Phi,[\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\Phi_1;\mathcal{R},[\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\mathcal{R}_1\vdash [\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\mathit{texpr}\Leftarrow \mathit{ret}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}',\mathcal{R}\vdash \texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i}=\mathit{seq\_expr}\,\texttt{in}\,\mathit{texpr}\Leftarrow \mathit{ret}}\quad \textsc{Seq\_TExpr\_Let}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi;\mathcal{R}'\vdash \mathit{texpr}_1\Leftarrow \mathit{ret}_1\\ \overline{\mathit{ret\_pattern}_i}^{\,i}:\mathit{ret}_1\rightsquigarrow \mathcal{C}_1;\mathcal{L}_1;\Phi_1;\mathcal{R}_1\\ \mathcal{C},\mathrm{fresh}(\mathcal{C}_1);\mathcal{L},\mathcal{L}_1;\Phi,[\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\Phi_1;\mathcal{R},[\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\mathcal{R}_1\vdash [\mathrm{fresh}(\mathcal{C}_1)/\mathcal{C}_1]\mathit{texpr}_2\Leftarrow \mathit{ret}_2\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}',\mathcal{R}\vdash \texttt{let}\,\overline{\mathit{ret\_pattern}_i}^{\,i}:\mathit{ret}_1=\mathit{texpr}_1\,\texttt{in}\,\mathit{texpr}\Leftarrow \mathit{ret}_2}\quad \textsc{Seq\_TExpr\_LetT}$$

$$\frac{\begin{array}{l}\mathcal{C};\mathcal{L};\Phi\vdash \mathit{pval}\Rightarrow \beta_1\\ \overline{y_1\,\texttt{as}\,\mathit{pattern}_i{:}\beta_1\rightsquigarrow \mathcal{C}_i;\Phi_i}^{\,i}\\ \overline{\mathcal{C},\mathrm{fresh}(\mathcal{C}_i);\mathcal{L},y_1{:}\beta_1;\Phi,y_1=\mathit{pval},[\mathrm{fresh}(\mathcal{C}_i)/\mathcal{C}_i]\Phi_i;\mathcal{R}\vdash [\mathrm{fresh}(\mathcal{C}_i)/\mathcal{C}_i]\mathit{texpr}_i\Leftarrow \mathit{ret}}^{\,i}\end{array}}{\mathcal{C};\mathcal{L};\Phi;\mathcal{R}\vdash \texttt{case}\,\mathit{pval}\,\texttt{of}\,\overline{\mid \mathit{pattern}_i\Rightarrow \mathit{texpr}_i}^{\,i}\,\texttt{end}\Leftarrow \mathit{ret}}\quad \textsc{Seq\_TExpr\_Case}$$

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

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

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

```
Definition rules:        89 good    0 bad
Definition rule clauses: 209 good    0 bad
```