n, i, j, k	Index variables for meta-lists
num	Numeric literals
nat	Internal literal numbers
hex	Bit vector literal, specified by C-style hex number
bin	Bit vector literal, specified by C-style binary number
string	String literals
regexp	Regular expresions, as a string literal
x, y, z	Variables
ix	Variables

l	$::=$		Source locations
	$\|$		
$x^l,\ y^l,\ z^l,\ name$	$::=$		Location-annotated names
	$\|$	$x\ l$	
	$\|$	$(ix)l$	Remove infix status
ix^l	$::=$		Location-annotated infix names
	$\|$	$ix\ l$	
	$\|$	$`x`l$	Add infix status
α	$::=$		Type variables
	$\|$	$'x$	
α^l	$::=$		Location-annotated type variables
	$\|$	$\alpha\ l$	
N	$::=$		numeric variables
	$\|$	$''x$	
N^l	$::=$		Location-annotated numeric variables
	$\|$	$N\ l$	
id	$::=$		Long identifers
	$\|$	$x^l_1. .. x^l_n.x^l\ l$	
tnv	$::=$		Union of type variables and Nexp type variables, without loca
	$\|$	α	
	$\|$	N	
$tnvar^l$	$::=$		Union of type variables and Nexp type variables, with location
	$\|$	α^l	
	$\|$	N^l	
$tnvs$	$::=$		Type variable lists
	$\|$	$tnv_1\ ..\ tnv_n$	
$tnvars^l$	$::=$		Type variable lists
	$\|$	$tnvar^l_1\ ..\ tnvar^l_n$	
$Nexp\_aux$	$::=$		Numerical expressions for specifying vector lengths and indexe
	$\|$	N	
	$\|$	num	
	$\|$	$Nexp_1 * Nexp_2$	
	$\|$	$Nexp_1 + Nexp_2$	
	$\|$	$(Nexp)$	

Nexp	::=		Location-annotated vector lengths
	|	$Nexp\_aux\ l$	
Nexp_constraint_aux	::=		Whether a vector is bounded or fixed size
	|	$Nexp = Nexp'$	
	|	$Nexp \geq Nexp'$	
Nexp_constraint	::=		Location-annotated Nexp range
	|	$Nexp\_constraint\_aux\ l$	
typ_aux	::=		Types
	|	_	Unspecified type
	|	α^l	Type variables
	|	$typ_1 \to typ_2$	Function types
	|	$typ_1 * \ldots. * typ_n$	Tuple types
	|	$Nexp$	As a typ to permit applications over Nexps, other
	|	$id\ typ_1 \mathrel{..} typ_n$	Type applications
	|	(typ)	
typ	::=		Location-annotated types
	|	$typ\_aux\ l$	
lit_aux	::=		Literal constants
	|	**true**	
	|	**false**	
	|	*num*	
	|	*hex*	hex and bin are constant bit vectors, entered as C
	|	*bin*	
	|	*string*	
	|	()	
	|	**bitzero**	bitzero and bitone are constant bits, if commonly
	|	**bitone**	
lit	::=		
	|	$lit\_aux\ l$	Location-annotated literal constants
$;^?$	::=		Optional semi-colons
	|		
	|	;	
pat_aux	::=		Patterns
	|	_	Wildcards
	|	$(pat\ \mathbf{as}\ x^l)$	Named patterns
	|	$(pat : typ)$	Typed patterns
	|	$id\ pat_1 \mathrel{..} pat_n$	Single variable and constructor patterns
	|	$\langle\vert fpat_1; \ldots; fpat_n\ ;^? \vert\rangle$	Record patterns
	|	$[\vert pat_1; \mathrel{..}; pat_n\ ;^? \vert]$	Vector patterns

	$\vert$	$[\vert pat_1 \,..\, pat_n \vert]$	Concatenated vector patterns
	$\vert$	$(pat_1 ,\ .... ,\ pat_n)$	Tuple patterns
	$\vert$	$[pat_1 ;\ ..\, ;\ pat_n\ ;^?]$	List patterns
	$\vert$	(pat)	
	$\vert$	$pat_1 :: pat_2$	Cons patterns
	$\vert$	$x^l + num$	constant addition patterns
	$\vert$	lit	Literal constant patterns
pat	$::=$		Location-annotated patterns
	$\vert$	$pat\_aux\ l$	
$fpat$	$::=$		Field patterns
	$\vert$	$id = pat\ l$	
$\vert^?$	$::=$		Optional bars
	$\vert$		
	$\vert$	$\vert$	
exp_aux	$::=$		Expressions
	$\vert$	id	Identifiers
	$\vert$	N	Nexp var, has type num
	$\vert$	$\mathbf{fun}\ psexp$	Curried functions
	$\vert$	$\mathbf{function}\ \vert^?\ pexp_1 \vert \ldots \vert pexp_n\ \mathbf{end}$	Functions with pattern matching
	$\vert$	$exp_1\ exp_2$	Function applications
	$\vert$	$exp_1\ ix^l\ exp_2$	Infix applications
	$\vert$	$\langle\vert fexps \vert\rangle$	Records
	$\vert$	$\langle\vert exp\ \mathbf{with}\ fexps \vert\rangle$	Functional update for records
	$\vert$	$exp.id$	Field projection for records
	$\vert$	$[\vert exp_1 ;\ ..\, ;\ exp_n\ ;^? \vert]$	Vector instantiation
	$\vert$	$exp.(Nexp)$	Vector access
	$\vert$	$exp.(Nexp_1..Nexp_2)$	Subvector extraction
	$\vert$	$\mathbf{match}\ exp\ \mathbf{with}\ \vert^?\ pexp_1 \vert \ldots \vert pexp_n\ l\ \mathbf{end}$	Pattern matching expressions
	$\vert$	$(exp : typ)$	Type-annotated expressions
	$\vert$	$\mathbf{let}\ letbind\ \mathbf{in}\ exp$	Let expressions
	$\vert$	$(exp_1 ,\ .... ,\ exp_n)$	Tuples
	$\vert$	$[exp_1 ;\ ..\, ;\ exp_n\ ;^?]$	Lists
	$\vert$	(exp)	
	$\vert$	$\mathbf{begin}\ exp\ \mathbf{end}$	Alternate syntax for (exp)
	$\vert$	$\mathbf{if}\ exp_1\ \mathbf{then}\ exp_2\ \mathbf{else}\ exp_3$	Conditionals
	$\vert$	$exp_1 :: exp_2$	Cons expressions
	$\vert$	lit	Literal constants
	$\vert$	$\{exp_1 \vert exp_2\}$	Set comprehensions
	$\vert$	$\{exp_1 \vert\ \mathbf{forall}\ qbind_1 \,..\, qbind_n \vert exp_2\}$	Set comprehensions with explicit bin
	$\vert$	$\{exp_1 ;\ ..\, ;\ exp_n\ ;^?\}$	Sets
	$\vert$	$q\ qbind_1 \ldots qbind_n.exp$	Logical quantifications
	$\vert$	$[exp_1 \vert\ \mathbf{forall}\ qbind_1 \,..\, qbind_n \vert exp_2]$	List comprehensions (all binders mus
	$\vert$	$\mathbf{do}\ id\ pat_1 < -exp_1 ;\ ..\ pat_n < -exp_n ;\ \mathbf{in}\ exp\ \mathbf{end}$	Do notation for monads

exp	::=		Location-annotated expressions
	|	$\mathit{exp\_aux}\ l$	
q	::=		Quantifiers
	|	**forall**	
	|	**exists**	
qbind	::=		Bindings for quantifiers
	|	x^l	
	|	$(\mathit{pat}\ \mathbf{IN}\ \mathit{exp})$	Restricted quantifications over sets
	|	$(\mathit{pat}\ \mathbf{MEM}\ \mathit{exp})$	Restricted quantifications over lists
fexp	::=		Field-expressions
	|	$\mathit{id} = \mathit{exp}\ l$	
fexps	::=		Field-expression lists
	|	$\mathit{fexp}_1;\ \ldots;\mathit{fexp}_n\ ;^?\ l$	
pexp	::=		Pattern matches
	|	$\mathit{pat} \rightarrow \mathit{exp}\ l$	
psexp	::=		Multi-pattern matches
	|	$\mathit{pat}_1 \ldots \mathit{pat}_n \rightarrow \mathit{exp}\ l$	
$\mathit{tannot}^?$	::=		Optional type annotations
	|		
	|	$: \mathit{typ}$	
$\mathit{funcl\_aux}$	::=		Function clauses
	|	$x^l\ \mathit{pat}_1 \ldots \mathit{pat}_n\ \mathit{tannot}^? = \mathit{exp}$	
$\mathit{letbind\_aux}$	::=		Let bindings
	|	$\mathit{pat}\ \mathit{tannot}^? = \mathit{exp}$	Value bindings
	|	$\mathit{funcl\_aux}$	Function bindings
$\mathit{letbind}$	::=		Location-annotated let bindings
	|	$\mathit{letbind\_aux}\ l$	
funcl	::=		Location-annotated function clauses
	|	$\mathit{funcl\_aux}\ l$	
$\mathit{id}^?$	::=		Optional name for inductively defined relatio
	|		
	|	$x^l :$	
$\mathit{rule\_aux}$	::=		Inductively defined relation clauses
	|	$\mathit{id}^?\ \mathbf{forall}\ x_1^l \mathinner{..} x_n^l.\mathit{exp} \Longrightarrow x^l\ \mathit{exp}_1 \mathinner{..} \mathit{exp}_i$	

$rule$	$::=$			Location-annotated inductively def
	$\vert$	$rule\_aux\ l$		
$typs$	$::=$			Type lists
	$\vert$	$typ_1 * \ldots * typ_n$		
$ctor\_def$	$::=$			Datatype definition clauses
	$\vert$	x^l **of** $typs$		
	$\vert$	x^l	S	Constant constructors
$texp$	$::=$			Type definition bodies
	$\vert$	typ		Type abbreviations
	$\vert$	$\langle\vert x_1^l : typ_1; \ldots ; x_n^l : typ_n\ ;^? \vert\rangle$		Record types
	$\vert$	$\vert^?\ ctor\_def_1 \vert \ldots \vert ctor\_def_n$		Variant types
$name^?$	$::=$			Optional name specification for va
	$\vert$			
	$\vert$	$[name = regexp]$		
td	$::=$			Type definitions
	$\vert$	$x^l\ tnvars^l\ name^? = texp$		
	$\vert$	$x^l\ tnvars^l\ name^?$		Definitions of opaque types
c	$::=$			Typeclass constraints
	$\vert$	$id\ tnvar^l$		
cs	$::=$			Typeclass and length constraint lis
	$\vert$			
	$\vert$	$c_1, .., c_i \Rightarrow$		Must have > 0 constraints
	$\vert$	$Nexp\_constraint_1, .., Nexp\_constraint_i \Rightarrow$		Must have > 0 constraints
	$\vert$	$c_1, .., c_i; Nexp\_constraint_1, .., Nexp\_constraint_n \Rightarrow$		Must have > 0 of both form of c
c_pre	$::=$			Type and instance scheme prefixes
	$\vert$			
	$\vert$	**forall** $tnvar_1^l .. tnvar_n^l . cs$		Must have > 0 type variables
$typschm$	$::=$			Type schemes
	$\vert$	$c\_pre\ typ$		
$instschm$	$::=$			Instance schemes
	$\vert$	$c\_pre(id\ typ)$		
$target$	$::=$			Backend target names
	$\vert$	**hol**		
	$\vert$	**isabelle**		
	$\vert$	**ocaml**		
	$\vert$	**coq**		

	$\|$	**tex**	
	$\|$	**html**	
τ	::=		Backend target name lists
	$\|$	$\{ target_1 ; \,..\, ; target_n \}$	
$\tau^?$	::=		Optional targets
	$\|$		
	$\|$	τ	
lemma_typ	::=		Types of Lemmata
	$\|$	**assert**	
	$\|$	**lemma**	
	$\|$	**theorem**	
lemma_decl	::=		Lemmata and Tests
	$\|$	$lemma\_typ\ \tau^?\ x^l : (exp)$	
	$\|$	$lemma\_typ\ \tau^? (exp)$	
val_def	::=		Value definitions
	$\|$	**let** $\tau^?$ *letbind*	Non-recursive value definitio
	$\|$	**let rec** $\tau^?$ $funcl_1$ **and** ... **and** $funcl_n$	Recursive function definition
	$\|$	**let inline** $\tau^?$ *letbind*	Function definitions to be inl
val_spec	::=		Value type specifications
	$\|$	**val** x^l : *typschm*	
def_aux	::=		Top-level definitions
	$\|$	**type** td_1 **and** ... **and** td_n	Type definitions
	$\|$	*val_def*	Value definitions
	$\|$	*lemma_decl*	Lemmata
	$\|$	**rename** $\tau^?\ id = x^l$	Rename constant or type
	$\|$	**module** $x^l =$ **struct** *defs* **end**	Module definitions
	$\|$	**module** $x^l = id$	Module renamings
	$\|$	**open** *id*	Opening modules
	$\|$	**indreln** $\tau^?$ $rule_1$ **and** ... **and** $rule_n$	Inductively defined relations
	$\|$	*val_spec*	Top-level type constraints
	$\|$	**class** $(x^l\ tnvar^l)$ **val** $x_1^l : typ_1\ l_1$ ... **val** $x_n^l : typ_n\ l_n$ **end**	Typeclass definitions
	$\|$	**instance** *instschm* $val\_def_1\ l_1$ ... $val\_def_n\ l_n$ **end**	Typeclass instantiations
def	::=		Location-annotated definitions
	$\|$	$def\_aux\ l$	
$;;^?$	::=		Optional double-semi-colon
	$\|$		
	$\|$	$;;$	

$defs$	::=			Definition sequences
	$\|$	$def_1\ ;;^?_1\ ..\ def_n\ ;;^?_n$		
p	::=			Unique paths
	$\|$	$x_1 . .. x_n . x$		
	$\|$	**__list**		
	$\|$	**__bool**		
	$\|$	**__num**		
	$\|$	**__set**		
	$\|$	**__string**		
	$\|$	**__unit**		
	$\|$	**__bit**		
	$\|$	**__vector**		
σ	::=			Type variable substitutions
	$\|$	$\{tnv_1 \mapsto t_1\ ..\ tnv_n \mapsto t_n\}$		
$t,\ u$	::=			Internal types
	$\|$	α		
	$\|$	$t_1 \to t_2$		
	$\|$	$t_1 * \ .... \ * t_n$		
	$\|$	$p\ t\_args$		
	$\|$	ne		
	$\|$	$\sigma(t)$	M	Multiple substitutions
	$\|$	$\sigma(tnv)$	M	Single variable substitution
	$\|$	$\mathbf{curry}\,(t\_multi, t)$	M	Curried, multiple argument functions
ne	::=			internal numeric expressions
	$\|$	N		
	$\|$	nat		
	$\|$	$ne_1 * ne_2$		
	$\|$	$ne_1 + ne_2$		
	$\|$	$(-ne)$		
	$\|$	$\mathbf{normalize}\,(ne)$	M	
	$\|$	$ne_1 + \ ...\ + ne_n$	M	
	$\|$	$\mathbf{bitlength}\,(bin)$	M	
	$\|$	$\mathbf{bitlength}\,(hex)$	M	
	$\|$	$\mathbf{length}\,(pat_1\ ...\ pat_n)$	M	
	$\|$	$\mathbf{length}\,(exp_1\ ...\ exp_n)$	M	
t_args	::=			Lists of types
	$\|$	$t_1\ ..\ t_n$		
	$\|$	$\sigma(t\_args)$	M	Multiple substitutions
t_multi	::=			Lists of types
	$\|$	$(t_1 *\ ..\ * t_n)$		
	$\|$	$\sigma(t\_multi)$	M	Multiple substitutions

nec	$::=$			Numeric expression constraints
	$\|$	$ne\langle nec$		
	$\|$	$ne = nec$		
	$\|$	$ne <= nec$		
	$\|$	ne		
$names$	$::=$			Sets of names
	$\|$	$\{x_1, \ .. \ , x_n\}$		
$\mathcal{C}$	$::=$			Typeclass constraint lists
	$\|$	$(p_1 \ tnv_1) \ .. \ (p_n \ tnv_n)$		
env_tag	$::=$			Tags for the (non-constructor) value descripti
	$\|$	**method**		Bound to a method
	$\|$	**val**		Specified with val
	$\|$	**let**		Defined with let or indreln
v_desc	$::=$			Value descriptions
	$\|$	$\langle\, \textbf{forall}\, tnvs.t\_multi \rightarrow p, (x \ \textbf{of} \ names)\rangle$		Constructors
	$\|$	$\langle\, \textbf{forall}\, tnvs.\mathcal{C} \Rightarrow t, env\_tag\rangle$		Values
f_desc	$::=$			
	$\|$	$\langle\, \textbf{forall}\, tnvs.p \rightarrow t, (x \ \textbf{of} \ names)\rangle$		Fields
xs	$::=$			
	$\|$	$x_1 \ .. \ x_n$		
$\Sigma^{\mathcal{C}}$	$::=$			Typeclass constraints
	$\|$	$\{(p_1 \ t_1), \ .. \ , (p_n \ t_n)\}$		
	$\|$	$\Sigma^{\mathcal{C}}{}_1 \cup \ .. \ \cup \Sigma^{\mathcal{C}}{}_n$	M	
$\Sigma^{\mathcal{N}}$	$::=$			Nexp constraint lists
	$\|$	$\{nec_1, \ .. \ , nec_n\}$		
	$\|$	$\Sigma^{\mathcal{N}}{}_1 \cup \ .. \ \cup \Sigma^{\mathcal{N}}{}_n$	M	
E	$::=$			Environments
	$\|$	$\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle$		
	$\|$	$E_1 \uplus E_2$	M	
	$\|$	ϵ	M	
E^{X}	$::=$			Value environments
	$\|$	$\{x_1 \mapsto v\_desc_1, \ .. \ , x_n \mapsto v\_desc_n\}$		
	$\|$	$E^{\mathrm{X}}_1 \uplus \ .. \ \uplus E^{\mathrm{X}}_n$	M	
E^{F}	$::=$			Field environments
	$\|$	$\{x_1 \mapsto f\_desc_1, \ .. \ , x_n \mapsto f\_desc_n\}$		
	$\|$	$E^{\mathrm{F}}_1 \uplus \ .. \ \uplus E^{\mathrm{F}}_n$	M	

E^{M}	::=			Module environments
	|	$\{x_1 \mapsto E_1, \,..\, , x_n \mapsto E_n\}$		
E^{P}	::=			Path environments
	|	$\{x_1 \mapsto p_1, \,..\, , x_n \mapsto p_n\}$		
	|	$E^{\mathrm{P}}_1 \uplus \,..\, \uplus E^{\mathrm{P}}_n$	M	
E^{L}	::=			Lexical bindings
	|	$\{x_1 \mapsto t_1, \,..\, , x_n \mapsto t_n\}$		
	|	$E^{\mathrm{L}}_1 \uplus \,..\, \uplus E^{\mathrm{L}}_n$	M	
tc_abbrev	::=			Type abbreviations
	|	$.t$		
	|			
tc_def	::=			Type and class constructor definitions
	|	$tnvs\ tc\_abbrev$		Type constructors
Δ	::=			Type constructor definitions
	|	$\{p_1 \mapsto tc\_def_1, \,..\, , p_n \mapsto tc\_def_n\}$		
	|	$\Delta_1 \uplus \Delta_2$	M	
δ	::=			Typeclass definitions
	|	$\{p_1 \mapsto xs_1, \,..\, , p_n \mapsto xs_n\}$		
	|	$\delta_1 \uplus \delta_2$	M	
$inst$	::=			A typeclass instance, t must not contain nested typ
	|	$\mathcal{C} \Rightarrow (p\ t)$		
I	::=			Global instances
	|	$\{inst_1, \,..\, , inst_n\}$		
	|	$I_1 \cup I_2$	M	
D	::=			Global type definition store
	|	$\langle \Delta, \delta, I \rangle$		
	|	$D_1 \uplus D_2$	M	
	|	ϵ	M	
$terminals$	::=			
	|	$\geq$		>=
	|	$\rightarrow$		->
	|	$\Longrightarrow$		==>
	|	$\langle\|$		<|
	|	$\|\rangle$		|>
	|	$\cap$		
	|	$\cup$		
	|	$\uplus$		

	$\mid$ $\notin$	
	$\mid$ $\subset$	
	$\mid$ $\neq$	
	$\mid$ $\emptyset$	
	$\mid$ $\langle$	
	$\mid$ $\rangle$	
	$\mid$ $\vdash$	
	$\mid$ $,$	
	$\mid$ $\mapsto$	
	$\mid$ $\triangleright$	
	$\mid$ $\leadsto$	
	$\mid$ $\Rightarrow$	
	$\mid$ $-$	
	$\mid$ ϵ	
formula	$::=$	
	$\mid$ *judgement*	
	$\mid$ $formula_1 \quad .. \quad formula_n$	
	$\mid$ $E^{\mathrm{M}}(x) \triangleright E$	Module lookup
	$\mid$ $E^{\mathrm{P}}(x) \triangleright p$	Path lookup
	$\mid$ $E^{\mathrm{F}}(x) \triangleright f\_desc$	Field lookup
	$\mid$ $E^{\mathrm{X}}(x) \triangleright v\_desc$	Value lookup
	$\mid$ $E^{\mathrm{L}}(x) \triangleright t$	Lexical binding lookup
	$\mid$ $\Delta(p) \triangleright tc\_def$	Type constructor lookup
	$\mid$ $\delta(p) \triangleright xs$	Type constructor lookup
	$\mid$ $\mathbf{dom}\,(E_1^{\mathrm{M}}) \cap \mathbf{dom}\,(E_2^{\mathrm{M}}) = \emptyset$	
	$\mid$ $\mathbf{dom}\,(E_1^{\mathrm{X}}) \cap \mathbf{dom}\,(E_2^{\mathrm{X}}) = \emptyset$	
	$\mid$ $\mathbf{dom}\,(E_1^{\mathrm{F}}) \cap \mathbf{dom}\,(E_2^{\mathrm{F}}) = \emptyset$	
	$\mid$ $\mathbf{dom}\,(E_1^{\mathrm{P}}) \cap \mathbf{dom}\,(E_2^{\mathrm{P}}) = \emptyset$	
	$\mid$ $\mathbf{disjoint\,doms}\,(E_1^{\mathrm{L}},\, ...\, , E_n^{\mathrm{L}})$	Pairwise disjoint domains
	$\mid$ $\mathbf{disjoint\,doms}\,(E_1^{\mathrm{X}},\, ...\, , E_n^{\mathrm{X}})$	Pairwise disjoint domains
	$\mid$ $\mathbf{compatible\,overlap}\,(x_1 \mapsto t_1,\, ..\, , x_n \mapsto t_n)$	$(x_i = x_j) \Longrightarrow (t_i = t_j)$
	$\mid$ $\mathbf{duplicates}\,(tnvs) = \emptyset$	
	$\mid$ $\mathbf{duplicates}\,(x_1,\, ..\, , x_n) = \emptyset$	
	$\mid$ $x \notin \mathbf{dom}\,(E^{\mathrm{L}})$	
	$\mid$ $x \notin \mathbf{dom}\,(E^{\mathrm{X}})$	
	$\mid$ $x \notin \mathbf{dom}\,(E^{\mathrm{F}})$	
	$\mid$ $p \notin \mathbf{dom}\,(\delta)$	
	$\mid$ $p \notin \mathbf{dom}\,(\Delta)$	
	$\mid$ $\mathbf{FV}\,(t) \subset tnvs$	Free type variables
	$\mid$ $\mathbf{FV}\,(t\_multi) \subset tnvs$	Free type variables
	$\mid$ $\mathbf{FV}\,(\mathcal{C}) \subset tnvs$	Free type variables
	$\mid$ $inst\,\mathbf{IN}\,I$	
	$\mid$ $(p\ t) \notin I$	
	$\mid$ $E_1^{\mathrm{L}} = E_2^{\mathrm{L}}$	
	$\mid$ $E_1^{\mathrm{X}} = E_2^{\mathrm{X}}$	

	|	$E_1^{\mathrm{F}} = E_2^{\mathrm{F}}$	
	|	$E_1 = E_2$	
	|	$\Delta_1 = \Delta_2$	
	|	$\delta_1 = \delta_2$	
	|	$I_1 = I_2$	
	|	$names_1 = names_2$	
	|	$t_1 = t_2$	
	|	$\sigma_1 = \sigma_2$	
	|	$p_1 = p_2$	
	|	$xs_1 = xs_2$	
	|	$tnvs_1 = tnvs_2$	
convert_tnvars	::=		
	|	$tnvars^l \leadsto tnvs$	
	|	$tnvar^l \leadsto tnv$	
look_m	::=		
	|	$E_1(x_1^l \,..\, x_n^l) \triangleright E_2$	Name path lookup
look_m_id	::=		
	|	$E_1(id) \triangleright E_2$	Module identifier lookup
look_tc	::=		
	|	$E(id) \triangleright p$	Path identifier lookup
check_t	::=		
	|	$\Delta \vdash t\,\mathbf{ok}$	Well-formed types
	|	$\Delta, tnv \vdash t\,\mathbf{ok}$	Well-formed type/Nexps m
teq	::=		
	|	$\Delta \vdash t_1 = t_2$	Type equality
convert_typ	::=		
	|	$\Delta, E \vdash typ \leadsto t$	Convert source types to in
	|	$\vdash Nexp \leadsto ne$	Convert and normalize nu
convert_typs	::=		
	|	$\Delta, E \vdash typs \leadsto t\_multi$	
check_lit	::=		
	|	$\vdash lit : t$	Typing literal constants
inst_field	::=		
	|	$\Delta, E \vdash \ \mathbf{field}\, id : p\ t\_args \to t \triangleright (x\ \mathbf{of}\ names)$	Field typing (also returns c
inst_ctor	::=		
	|	$\Delta, E \vdash \ \mathbf{ctor}\, id : t\_multi \to p\ t\_args \triangleright (x\ \mathbf{of}\ names)$	Data constructor typing (a

inst_val	::=		
	|	$\Delta, E \vdash \textbf{val}\, id : t \vartriangleright \Sigma^{\mathcal{C}}$	Typing top-level bindings, coll
not_ctor	::=		
	|	$E, E^{\text{L}} \vdash x\, \textbf{not ctor}$	v is not bound to a data const
not_shadowed	::=		
	|	$E^{\text{L}} \vdash id\, \textbf{not shadowed}$	id is not lexically shadowed
check_pat	::=		
	|	$\Delta, E, E_1^{\text{L}} \vdash pat : t \vartriangleright E_2^{\text{L}}$	Typing patterns, building thei
	|	$\Delta, E, E_1^{\text{L}} \vdash pat\_aux : t \vartriangleright E_2^{\text{L}}$	Typing patterns, building thei
id_field	::=		
	|	$E \vdash id\, \textbf{field}$	Check that the identifier is a p
id_value	::=		
	|	$E \vdash id\, \textbf{value}$	Check that the identifier is a p
check_exp	::=		
	|	$\Delta, E, E^{\text{L}} \vdash exp : t \vartriangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecting
	|	$\Delta, E, E^{\text{L}} \vdash exp\_aux : t \vartriangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecting
	|	$\Delta, E, E_1^{\text{L}} \vdash qbind_1 \,..\, qbind_n \vartriangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}$	Build the environment for qua
	|	$\Delta, E, E_1^{\text{L}} \vdash \textbf{list}\, qbind_1 \,..\, qbind_n \vartriangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}$	Build the environment for qua
	|	$\Delta, E, E^{\text{L}} \vdash funcl \vartriangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a fu
	|	$\Delta, E, E_1^{\text{L}} \vdash letbind \vartriangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a le
check_rule	::=		
	|	$\Delta, E, E^{\text{L}} \vdash rule \vartriangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for an i
check_texp_tc	::=		
	|	$xs, \Delta_1, E \vdash \textbf{tc}\, td \vartriangleright \Delta_2, E^{\text{P}}$	Extract the type constructor i
check_texps_tc	::=		
	|	$xs, \Delta_1, E \vdash \textbf{tc}\, td_1 \,..\, td_i \vartriangleright \Delta_2, E^{\text{P}}$	Extract the type constructor i
check_texp	::=		
	|	$\Delta, E \vdash tnvs\; p = texp \vartriangleright \langle E^{\text{F}}, E^{\text{X}} \rangle$	Check a type definition, with i
check_texps	::=		
	|	$xs, \Delta, E \vdash td_1 \,..\, td_n \vartriangleright \langle E^{\text{F}}, E^{\text{X}} \rangle$	
convert_class	::=		
	|	$\delta, E \vdash id \rightsquigarrow p$	Lookup a type class
solve_class_constraint	::=		
	|	$I \vdash (p\; t)\, \textbf{IN}\, \mathcal{C}$	Solve class constraint

solve_class_constraints	::=		
	|	$I \vdash \Sigma^{\mathcal{C}} \triangleright \mathcal{C}$	Solve class constraints
check_val_def	::=		
	|	$\Delta, I, E \vdash \mathit{val\_def} \triangleright E^{\mathrm{X}}$	Check a value definition
check_t_instance	::=		
	|	$\Delta, (\alpha_1, .., \alpha_n) \vdash t\,\mathbf{instance}$	Check that t be a typeclass instance
check_defs	::=		
	|	$\overline{z_j}^{\,j}, D_1, E_1 \vdash \mathit{def} \triangleright D_2, E_2$	Check a definition
	|	$\overline{z_j}^{\,j}, D_1, E_1 \vdash \mathit{defs} \triangleright D_2, E_2$	Check definitions, given module path, defi
judgement	::=		
	|	*convert_tnvars*	
	|	*look_m*	
	|	*look_m_id*	
	|	*look_tc*	
	|	*check_t*	
	|	*teq*	
	|	*convert_typ*	
	|	*convert_typs*	
	|	*check_lit*	
	|	*inst_field*	
	|	*inst_ctor*	
	|	*inst_val*	
	|	*not_ctor*	
	|	*not_shadowed*	
	|	*check_pat*	
	|	*id_field*	
	|	*id_value*	
	|	*check_exp*	
	|	*check_rule*	
	|	*check_texp_tc*	
	|	*check_texps_tc*	
	|	*check_texp*	
	|	*check_texps*	
	|	*convert_class*	
	|	*solve_class_constraint*	
	|	*solve_class_constraints*	
	|	*check_val_def*	
	|	*check_t_instance*	
	|	*check_defs*	
user_syntax	::=		
	|	*n*	
	|	*num*	

$$
\begin{array}{ll}
| & nat \\
| & hex \\
| & bin \\
| & string \\
| & regexp \\
| & x \\
| & ix \\
| & l \\
| & x^{l} \\
| & ix^{l} \\
| & \alpha \\
| & \alpha^{l} \\
| & N \\
| & N^{l} \\
| & id \\
| & tnv \\
| & tnvar^{l} \\
| & tnvs \\
| & tnvars^{l} \\
| & Nexp\_aux \\
| & Nexp \\
| & Nexp\_constraint\_aux \\
| & Nexp\_constraint \\
| & typ\_aux \\
| & typ \\
| & lit\_aux \\
| & lit \\
| & ;^{?} \\
| & pat\_aux \\
| & pat \\
| & fpat \\
| & |^{?} \\
| & exp\_aux \\
| & exp \\
| & q \\
| & qbind \\
| & fexp \\
| & fexps \\
| & pexp \\
| & psexp \\
| & tannot^{?} \\
| & funcl\_aux \\
| & letbind\_aux \\
| & letbind \\
| & funcl \\
| & id^{?}
\end{array}
$$

| $rule\_aux$
| $rule$
| $typs$
| $ctor\_def$
| $texp$
| $name^?$
| td
| c
| cs
| c_pre
| $typschm$
| $instschm$
| $target$
| τ
| $\tau^?$
| $lemma\_typ$
| $lemma\_decl$
| val_def
| val_spec
| def_aux
| def
| $;;^?$
| $defs$
| p
| σ
| t
| ne
| t_args
| t_multi
| nec
| $names$
| $\mathcal{C}$
| env_tag
| v_desc
| f_desc
| xs
| $\Sigma^{\mathcal{C}}$
| $\Sigma^{\mathcal{N}}$
| E
| E^{X}
| E^{F}
| E^{M}
| E^{P}
| E^{L}
| tc_abbrev
| tc_def

$|\ \Delta$
$|\ \delta$
$|\ \mathit{inst}$
$|\ I$
$|\ D$
$|\ \mathit{terminals}$
$|\ \mathit{formula}$

$\boxed{\mathit{tnvars}^l \leadsto \mathit{tnvs}}$

$$\frac{\mathit{tnvar}_1^l \leadsto \mathit{tnv}_1 \quad .. \quad \mathit{tnvar}_n^l \leadsto \mathit{tnv}_n}{\mathit{tnvar}_1^l \,..\, \mathit{tnvar}_n^l \leadsto \mathit{tnv}_1 \,..\, \mathit{tnv}_n} \quad \text{CONVERT\_TNVARS\_NONE}$$

$\boxed{\mathit{tnvar}^l \leadsto \mathit{tnv}}$

$$\frac{}{\alpha\, l \leadsto \alpha} \quad \text{CONVERT\_TNVAR\_A}$$

$$\frac{}{N\, l \leadsto N} \quad \text{CONVERT\_TNVAR\_N}$$

$\boxed{E_1(x_1^l \,..\, x_n^l) \triangleright E_2}$ Name path lookup

$$\frac{}{E(\,) \triangleright E} \quad \text{LOOK\_M\_NONE}$$

$$\frac{E^{\mathrm{M}}(x) \triangleright E_1 \qquad E_1(\,\overline{y_i^l}^{\,i}\,) \triangleright E_2}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle(x\, l\, \overline{y_i^l}^{\,i}\,) \triangleright E_2} \quad \text{LOOK\_M\_SOME}$$

$\boxed{E_1(\mathit{id}) \triangleright E_2}$ Module identifier lookup

$$\frac{E_1(\,\overline{y_i^l}^{\,i}\, x\, l_1) \triangleright E_2}{E_1(\overline{y_i^l.}^{\,i}\ x\, l_1\, l_2) \triangleright E_2} \quad \text{LOOK\_M\_ID\_ALL}$$

$\boxed{E(\mathit{id}) \triangleright p}$ Path identifier lookup

$$\frac{E(\,\overline{y_i^l}^{\,i}\,) \triangleright \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \qquad E^{\mathrm{P}}(x) \triangleright p}{E(\overline{y_i^l.}^{\,i}\ x\, l_1\, l_2) \triangleright p} \quad \text{LOOK\_TC\_ALL}$$

$\boxed{\Delta \vdash t\ \mathbf{ok}}$ Well-formed types

$$\frac{}{\Delta \vdash \alpha\ \mathbf{ok}} \quad \text{CHECK\_T\_VAR}$$

$$\frac{\Delta \vdash t_1\ \mathbf{ok} \qquad \Delta \vdash t_2\ \mathbf{ok}}{\Delta \vdash t_1 \to t_2\ \mathbf{ok}} \quad \text{CHECK\_T\_FN}$$

$$\frac{\Delta \vdash t_1\ \mathbf{ok} \quad .... \quad \Delta \vdash t_n\ \mathbf{ok}}{\Delta \vdash t_1 * \ .... \ * t_n\ \mathbf{ok}} \quad \text{CHECK\_T\_TUP}$$

$$\frac{\Delta(p) \triangleright \mathit{tnv}_1 \,..\, \mathit{tnv}_n\ \mathit{tc\_abbrev} \qquad \Delta, \mathit{tnv}_1 \vdash t_1\ \mathbf{ok} \quad .. \quad \Delta, \mathit{tnv}_n \vdash t_n\ \mathbf{ok}}{\Delta \vdash p\ t_1 \,..\, t_n\ \mathbf{ok}} \quad \text{CHECK\_T\_APP}$$

$\boxed{\Delta, tnv \vdash t\ \mathbf{ok}}$ Well-formed type/Nexps matching the application type variable

$$\frac{\Delta \vdash t\ \mathbf{ok}}{\Delta, \alpha \vdash t\ \mathbf{ok}} \quad \text{CHECK\_TLEN\_T}$$

$$\frac{}{\Delta, N \vdash ne\ \mathbf{ok}} \quad \text{CHECK\_TLEN\_LEN}$$

$\boxed{\Delta \vdash t_1 = t_2}$ Type equality

$$\frac{\Delta \vdash t\ \mathbf{ok}}{\Delta \vdash t = t} \quad \text{TEQ\_REFL}$$

$$\frac{\Delta \vdash t_2 = t_1}{\Delta \vdash t_1 = t_2} \quad \text{TEQ\_SYM}$$

$$\frac{\begin{array}{c}\Delta \vdash t_1 = t_2 \\ \Delta \vdash t_2 = t_3\end{array}}{\Delta \vdash t_1 = t_3} \quad \text{TEQ\_TRANS}$$

$$\frac{\begin{array}{c}\Delta \vdash t_1 = t_3 \\ \Delta \vdash t_2 = t_4\end{array}}{\Delta \vdash t_1 \to t_2 = t_3 \to t_4} \quad \text{TEQ\_ARROW}$$

$$\frac{\Delta \vdash t_1 = u_1 \quad .... \quad \Delta \vdash t_n = u_n}{\Delta \vdash t_1 * \ .... \ * t_n = u_1 * \ .... \ * u_n} \quad \text{TEQ\_TUP}$$

$$\frac{\begin{array}{l}\Delta(p) \rhd \alpha_1 \mathinner{..} \alpha_n \\ \Delta \vdash t_1 = u_1 \quad .. \quad \Delta \vdash t_n = u_n\end{array}}{\Delta \vdash p\ t_1 \mathinner{..} t_n = p\ u_1 \mathinner{..} u_n} \quad \text{TEQ\_APP}$$

$$\frac{\Delta(p) \rhd \alpha_1 \mathinner{..} \alpha_n\ .u}{\Delta \vdash p\ t_1 \mathinner{..} t_n = \{\alpha_1 \mapsto t_1 \mathinner{..} \alpha_n \mapsto t_n\}(u)} \quad \text{TEQ\_EXPAND}$$

$$\frac{ne = \mathbf{normalize}\,(ne')}{\Delta \vdash ne = ne'} \quad \text{TEQ\_NEXP}$$

$\boxed{\Delta, E \vdash typ \rightsquigarrow t}$ Convert source types to internal types

$$\frac{}{\Delta, E \vdash \alpha\ l'\ l \rightsquigarrow \alpha} \quad \text{CONVERT\_TYP\_VAR}$$

$$\frac{\begin{array}{c}\Delta, E \vdash typ_1 \rightsquigarrow t_1 \\ \Delta, E \vdash typ_2 \rightsquigarrow t_2\end{array}}{\Delta, E \vdash typ_1 \to typ_2\ l \rightsquigarrow t_1 \to t_2} \quad \text{CONVERT\_TYP\_FN}$$

$$\frac{\Delta, E \vdash typ_1 \rightsquigarrow t_1 \quad .... \quad \Delta, E \vdash typ_n \rightsquigarrow t_n}{\Delta, E \vdash typ_1 * \ .... \ * typ_n\ l \rightsquigarrow t_1 * \ .... \ * t_n} \quad \text{CONVERT\_TYP\_TUP}$$

$$\frac{\begin{array}{l}\Delta, E \vdash typ_1 \rightsquigarrow t_1 \quad .. \quad \Delta, E \vdash typ_n \rightsquigarrow t_n \\ E(id) \rhd p \\ \Delta(p) \rhd \alpha_1 \mathinner{..} \alpha_n\ tc\_abbrev\end{array}}{\Delta, E \vdash id\ typ_1 \mathinner{..} typ_n\ l \rightsquigarrow p\ t_1 \mathinner{..} t_n} \quad \text{CONVERT\_TYP\_APP}$$

$$\frac{\vdash Nexp \rightsquigarrow ne}{\Delta, E \vdash Nexp\ \rightsquigarrow ne} \quad \text{CONVERT\_TYP\_NEXP}$$

$$\frac{\Delta, E \vdash typ \rightsquigarrow t}{\Delta, E \vdash (typ)\ l \rightsquigarrow t} \quad \text{CONVERT\_TYP\_PAREN}$$

$$\frac{\Delta, E \vdash typ \leadsto t_1 \quad \Delta \vdash t_1 = t_2}{\Delta, E \vdash typ \leadsto t_2} \quad \text{CONVERT\_TYP\_EQ}$$

$\boxed{\vdash Nexp \leadsto ne}$ Convert and normalize numeric expressions

$$\frac{}{\vdash N\ l \leadsto N} \quad \text{CONVERT\_NEXP\_VAR}$$

$$\frac{}{\vdash num\ l \leadsto nat} \quad \text{CONVERT\_NEXP\_NUM}$$

$$\frac{\vdash Nexp_1 \leadsto ne_1 \quad \vdash Nexp_2 \leadsto ne_2}{\vdash Nexp_1 * Nexp_2\ l \leadsto ne_1 * ne_2} \quad \text{CONVERT\_NEXP\_MULT}$$

$$\frac{\vdash Nexp_1 \leadsto ne_1 \quad \vdash Nexp_2 \leadsto ne_2}{\vdash Nexp_1 + Nexp_2\ l \leadsto ne_1 + ne_2} \quad \text{CONVERT\_NEXP\_ADD}$$

$\boxed{\Delta, E \vdash typs \leadsto t\_multi}$

$$\frac{\Delta, E \vdash typ_1 \leadsto t_1 \quad .. \quad \Delta, E \vdash typ_n \leadsto t_n}{\Delta, E \vdash typ_1 * .. * typ_n \leadsto (t_1 * .. * t_n)} \quad \text{CONVERT\_TYPS\_ALL}$$

$\boxed{\vdash lit : t}$ Typing literal constants

$$\frac{}{\vdash \mathbf{true}\ l : \_\_\mathbf{bool}} \quad \text{CHECK\_LIT\_TRUE}$$

$$\frac{}{\vdash \mathbf{false}\ l : \_\_\mathbf{bool}} \quad \text{CHECK\_LIT\_FALSE}$$

$$\frac{}{\vdash num\ l : \_\_\mathbf{num}} \quad \text{CHECK\_LIT\_NUM}$$

$$\frac{nat = \mathbf{bitlength}\,(hex)}{\vdash hex\ l : \_\_\mathbf{vector}\ nat\ \_\_\mathbf{bit}} \quad \text{CHECK\_LIT\_HEX}$$

$$\frac{nat = \mathbf{bitlength}\,(bin)}{\vdash bin\ l : \_\_\mathbf{vector}\ nat\ \_\_\mathbf{bit}} \quad \text{CHECK\_LIT\_BIN}$$

$$\frac{}{\vdash string\ l : \_\_\mathbf{string}} \quad \text{CHECK\_LIT\_STRING}$$

$$\frac{}{\vdash ()\ l : \_\_\mathbf{unit}} \quad \text{CHECK\_LIT\_UNIT}$$

$$\frac{}{\vdash \mathbf{bitzero}\ l : \_\_\mathbf{bit}} \quad \text{CHECK\_LIT\_BITZERO}$$

$$\frac{}{\vdash \mathbf{bitone}\ l : \_\_\mathbf{bit}} \quad \text{CHECK\_LIT\_BITONE}$$

$\boxed{\Delta, E \vdash \mathbf{field}\ id : p\ t\_args \to t \triangleright (x\ \mathbf{of}\ names)}$ Field typing (also returns canonical field names)

$$\frac{\begin{array}{l} E(\overline{x_i^l}^{\,i}) \triangleright \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}} \rangle \\ E^{\mathrm{F}}(y) \triangleright \langle \mathbf{forall}\ tnv_1 .. tnv_n . p \to t, (z\ \mathbf{of}\ names) \rangle \\ \Delta \vdash t_1\ \mathbf{ok} \quad .. \quad \Delta \vdash t_n\ \mathbf{ok} \end{array}}{\Delta, E \vdash \mathbf{field}\ \overline{x_i^l.}^{\,i}\ y\ l_1\ l_2 : p\ t_1 .. t_n \to \{tnv_1 \mapsto t_1 .. tnv_n \mapsto t_n\}(t) \triangleright (z\ \mathbf{of}\ names)} \quad \text{INST\_FIELD\_ALL}$$

$\boxed{\Delta, E \vdash \mathbf{ctor}\ id : t\_multi \to p\ t\_args \triangleright (x\ \mathbf{of}\ names)}$ Data constructor typing (also returns canonical constru

$$\frac{\begin{array}{l} E(\,\overline{x_i^l}^{\,i}\,) \rhd \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}} \rangle \\ E^{\mathrm{X}}(y) \rhd \langle\, \mathbf{forall}\, tnv_1 \mathinner{..} tnv_n . t\_multi \to p, (z\, \mathbf{of}\, names) \rangle \\ \Delta \vdash t_1\, \mathbf{ok} \quad .. \quad \Delta \vdash t_n\, \mathbf{ok} \end{array}}{\Delta, E \vdash\ \mathbf{ctor}\, \overline{x_i^l .}^{\,i}\ y\, l_1\, l_2 : \{tnv_1 \mapsto t_1 \mathinner{..} tnv_n \mapsto t_n\}(t\_multi) \to p\, t_1 \mathinner{..} t_n \rhd (z\, \mathbf{of}\, names)} \quad \text{INST\_CTOR\_ALL}$$

$\boxed{\Delta, E \vdash\ \mathbf{val}\, id : t \rhd \Sigma^{\mathcal{C}}}$ Typing top-level bindings, collecting typeclass constraints

$$\frac{\begin{array}{l} E(\,\overline{x_i^l}^{\,i}\,) \rhd \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}} \rangle \\ E^{\mathrm{X}}(y) \rhd \langle\, \mathbf{forall}\, tnv_1 \mathinner{..} tnv_n . (p_1\, tnv'_1) \mathinner{..} (p_i\, tnv'_i) \Rightarrow t, env\_tag \rangle \\ \Delta \vdash t_1\, \mathbf{ok} \quad .. \quad \Delta \vdash t_n\, \mathbf{ok} \\ \sigma = \{tnv_1 \mapsto t_1 \mathinner{..} tnv_n \mapsto t_n\} \end{array}}{\Delta, E \vdash\ \mathbf{val}\, \overline{x_i^l .}^{\,i}\ y\, l_1\, l_2 : \sigma(t) \rhd \{(p_1\, \sigma(tnv'_1)), .., (p_i\, \sigma(tnv'_i))\}} \quad \text{INST\_VAL\_ALL}$$

$\boxed{E, E^{\mathrm{L}} \vdash x\, \mathbf{not\ ctor}}$ v is not bound to a data constructor

$$\frac{E^{\mathrm{L}}(x) \rhd t}{E, E^{\mathrm{L}} \vdash x\, \mathbf{not\ ctor}} \quad \text{NOT\_CTOR\_VAL}$$

$$\frac{x \notin\ \mathbf{dom}\,(E^{\mathrm{X}})}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}} \rangle, E^{\mathrm{L}} \vdash x\, \mathbf{not\ ctor}} \quad \text{NOT\_CTOR\_UNBOUND}$$

$$\frac{E^{\mathrm{X}}(x) \rhd \langle\, \mathbf{forall}\, tnv_1 \mathinner{..} tnv_n . (p_1\, tnv'_1) \mathinner{..} (p_i\, tnv'_i) \Rightarrow t, env\_tag \rangle}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}} \rangle, E^{\mathrm{L}} \vdash x\, \mathbf{not\ ctor}} \quad \text{NOT\_CTOR\_BOUND}$$

$\boxed{E^{\mathrm{L}} \vdash id\, \mathbf{not\ shadowed}}$ id is not lexically shadowed

$$\frac{x \notin\ \mathbf{dom}\,(E^{\mathrm{L}})}{E^{\mathrm{L}} \vdash\ x\, l_1\, l_2\, \mathbf{not\ shadowed}} \quad \text{NOT\_SHADOWED\_SING}$$

$$\frac{}{E^{\mathrm{L}} \vdash x_1^l . \mathinner{..} x_n^l . y^l . z^l\ l\, \mathbf{not\ shadowed}} \quad \text{NOT\_SHADOWED\_MULTI}$$

$\boxed{\Delta, E, E_1^{\mathrm{L}} \vdash pat : t \rhd E_2^{\mathrm{L}}}$ Typing patterns, building their binding environment

$$\frac{\Delta, E, E_1^{\mathrm{L}} \vdash pat\_aux : t \rhd E_2^{\mathrm{L}}}{\Delta, E, E_1^{\mathrm{L}} \vdash pat\_aux\ l : t \rhd E_2^{\mathrm{L}}} \quad \text{CHECK\_PAT\_ALL}$$

$\boxed{\Delta, E, E_1^{\mathrm{L}} \vdash pat\_aux : t \rhd E_2^{\mathrm{L}}}$ Typing patterns, building their binding environment

$$\frac{\Delta \vdash t\, \mathbf{ok}}{\Delta, E, E^{\mathrm{L}} \vdash \_ : t \rhd \{\,\}} \quad \text{CHECK\_PAT\_AUX\_WILD}$$

$$\frac{\begin{array}{l} \Delta, E, E_1^{\mathrm{L}} \vdash pat : t \rhd E_2^{\mathrm{L}} \\ x \notin\ \mathbf{dom}\,(E_2^{\mathrm{L}}) \end{array}}{\Delta, E, E_1^{\mathrm{L}} \vdash (pat\, \mathbf{as}\, x\ l) : t \rhd E_2^{\mathrm{L}} \uplus \{x \mapsto t\}} \quad \text{CHECK\_PAT\_AUX\_AS}$$

$$\frac{\begin{array}{l} \Delta, E, E_1^{\mathrm{L}} \vdash pat : t \rhd E_2^{\mathrm{L}} \\ \Delta, E \vdash typ \leadsto t \end{array}}{\Delta, E, E_1^{\mathrm{L}} \vdash (pat : typ) : t \rhd E_2^{\mathrm{L}}} \quad \text{CHECK\_PAT\_AUX\_TYP}$$

$$\frac{\begin{array}{l} \Delta, E \vdash\ \mathbf{ctor}\, id : (t_1 * \mathinner{..} * t_n) \to p\ t\_args \rhd (x\, \mathbf{of}\, names) \\ E^{\mathrm{L}} \vdash id\, \mathbf{not\ shadowed} \\ \Delta, E, E^{\mathrm{L}} \vdash pat_1 : t_1 \rhd E_1^{\mathrm{L}} \quad .. \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t_n \rhd E_n^{\mathrm{L}} \\ \mathbf{disjoint\ doms}\,(E_1^{\mathrm{L}}, .., E_n^{\mathrm{L}}) \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash id\ pat_1 \mathinner{..} pat_n : p\ t\_args \rhd E_1^{\mathrm{L}} \uplus \mathinner{..} \uplus E_n^{\mathrm{L}}} \quad \text{CHECK\_PAT\_AUX\_IDENT\_CONSTR}$$

$$\frac{\Delta \vdash t\ \mathbf{ok} \qquad E, E^{\mathrm{L}} \vdash x\ \mathbf{not\ ctor}}{\Delta, E, E^{\mathrm{L}} \vdash\ x\ l_1\ l_2\ :\ t \vartriangleright \{x \mapsto t\}} \quad \text{CHECK\_PAT\_AUX\_VAR}$$

$$\frac{\overline{\Delta, E \vdash\ \mathbf{field}\ id_i : p\ t\_args \to t_i \vartriangleright (x_i\ \mathbf{of}\ names)}^{i} \quad \overline{\Delta, E, E^{\mathrm{L}} \vdash pat_i : t_i \vartriangleright E^{\mathrm{L}}_i}^{i} \quad \mathbf{disjoint\ doms}\,(\,\overline{E^{\mathrm{L}}_i}^{i}\,) \quad \mathbf{duplicates}\,(\,\overline{x_i}^{i}\,) = \emptyset}{\Delta, E, E^{\mathrm{L}} \vdash \langle|\,\overline{id_i = pat_i\ l_i}^{i}\,;^{?}|\rangle : p\ t\_args \vartriangleright \uplus\,\overline{E^{\mathrm{L}}_i}^{i}} \quad \text{CHECK\_PAT\_AUX\_RECORD}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash pat_1 : t \vartriangleright E^{\mathrm{L}}_1 \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t \vartriangleright E^{\mathrm{L}}_n \quad \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_1, \ldots, E^{\mathrm{L}}_n) \quad \mathbf{length}\,(pat_1 \ldots pat_n) = nat}{\Delta, E, E^{\mathrm{L}} \vdash [|pat_1; \ldots; pat_n\,;^{?}|] : \_\_\mathbf{vector}\ nat\ t \vartriangleright E^{\mathrm{L}}_1 \uplus \ldots \uplus E^{\mathrm{L}}_n} \quad \text{CHECK\_PAT\_AUX\_VECTOR}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash pat_1 : \_\_\mathbf{vector}\ ne_1\ t \vartriangleright E^{\mathrm{L}}_1 \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : \_\_\mathbf{vector}\ ne_n\ t \vartriangleright E^{\mathrm{L}}_n \quad \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_1, \ldots, E^{\mathrm{L}}_n) \quad ne' = ne_1 + \ldots + ne_n}{\Delta, E, E^{\mathrm{L}} \vdash [|pat_1 \ldots pat_n|] : \_\_\mathbf{vector}\ ne'\ t \vartriangleright E^{\mathrm{L}}_1 \uplus \ldots \uplus E^{\mathrm{L}}_n} \quad \text{CHECK\_PAT\_AUX\_VECTORC}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash pat_1 : t_1 \vartriangleright E^{\mathrm{L}}_1 \quad \ldots. \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t_n \vartriangleright E^{\mathrm{L}}_n \quad \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_1, \ldots., E^{\mathrm{L}}_n)}{\Delta, E, E^{\mathrm{L}} \vdash (pat_1, \ldots., pat_n) : t_1 * \ldots. * t_n \vartriangleright E^{\mathrm{L}}_1 \uplus \ldots. \uplus E^{\mathrm{L}}_n} \quad \text{CHECK\_PAT\_AUX\_TUP}$$

$$\frac{\Delta \vdash t\ \mathbf{ok} \quad \Delta, E, E^{\mathrm{L}} \vdash pat_1 : t \vartriangleright E^{\mathrm{L}}_1 \quad .. \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t \vartriangleright E^{\mathrm{L}}_n \quad \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_1, .., E^{\mathrm{L}}_n)}{\Delta, E, E^{\mathrm{L}} \vdash [pat_1; ..; pat_n\,;^{?}] : \_\_\mathbf{list}\ t \vartriangleright E^{\mathrm{L}}_1 \uplus .. \uplus E^{\mathrm{L}}_n} \quad \text{CHECK\_PAT\_AUX\_LIST}$$

$$\frac{\Delta, E, E^{\mathrm{L}}_1 \vdash pat : t \vartriangleright E^{\mathrm{L}}_2}{\Delta, E, E^{\mathrm{L}}_1 \vdash (pat) : t \vartriangleright E^{\mathrm{L}}_2} \quad \text{CHECK\_PAT\_AUX\_PAREN}$$

$$\frac{\Delta, E, E^{\mathrm{L}}_1 \vdash pat_1 : t \vartriangleright E^{\mathrm{L}}_2 \quad \Delta, E, E^{\mathrm{L}}_1 \vdash pat_2 : \_\_\mathbf{list}\ t \vartriangleright E^{\mathrm{L}}_3 \quad \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_2, E^{\mathrm{L}}_3)}{\Delta, E, E^{\mathrm{L}}_1 \vdash pat_1 :: pat_2 : \_\_\mathbf{list}\ t \vartriangleright E^{\mathrm{L}}_2 \uplus E^{\mathrm{L}}_3} \quad \text{CHECK\_PAT\_AUX\_CONS}$$

$$\frac{\vdash lit : t}{\Delta, E, E^{\mathrm{L}} \vdash lit : t \vartriangleright \{\,\}} \quad \text{CHECK\_PAT\_AUX\_LIT}$$

$$\frac{E, E^{\mathrm{L}} \vdash x\ \mathbf{not\ ctor}}{\Delta, E, E^{\mathrm{L}} \vdash x\ l + num : \_\_\mathbf{num} \ \vartriangleright \{x \mapsto \_\_\mathbf{num}\,\}} \quad \text{CHECK\_PAT\_AUX\_NUM\_ADD}$$

$\boxed{E \vdash id\ \mathbf{field}}$ Check that the identifier is a permissible field identifier

$$\frac{E^{\mathrm{F}}(x) \vartriangleright f\_desc}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \vdash\ x\ l_1\ l_2\ \mathbf{field}} \quad \text{ID\_FIELD\_EMPTY}$$

$$\frac{E^{\mathrm{M}}(x) \vartriangleright E \quad x \notin \mathbf{dom}\,(E^{\mathrm{F}}) \quad E \vdash \overline{y^{l}_i.}^{i}\ z^{l}\ l_2\ \mathbf{field}}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \vdash x\ l_1.\ \overline{y^{l}_i.}^{i}\ z^{l}\ l_2\ \mathbf{field}} \quad \text{ID\_FIELD\_CONS}$$

$\boxed{E \vdash id\ \mathbf{value}}$ Check that the identifier is a permissible value identifier

$$\frac{E^{\mathrm{X}}(x) \triangleright v\_desc}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \vdash x\ l_1\ l_2\ \mathbf{value}} \quad \text{ID\_VALUE\_EMPTY}$$

$$\frac{\begin{array}{l} E^{\mathrm{M}}(x) \triangleright E \\ x \notin \mathbf{dom}\,(E^{\mathrm{X}}) \\ E \vdash \overline{y_i^l.}^{\,i}\ z^l\ l_2\ \mathbf{value} \end{array}}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \vdash x\ l_1.\ \overline{y_i^l.}^{\,i}\ z^l\ l_2\ \mathbf{value}} \quad \text{ID\_VALUE\_CONS}$$

$\boxed{\Delta, E, E^{\mathrm{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ Typing expressions, collecting typeclass and index constraints

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash exp\_aux : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}} \vdash exp\_aux\ l : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_EXP\_ALL}$$

$\boxed{\Delta, E, E^{\mathrm{L}} \vdash exp\_aux : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ Typing expressions, collecting typeclass and index constraints

$$\frac{E^{\mathrm{L}}(x) \triangleright t}{\Delta, E, E^{\mathrm{L}} \vdash x\ l_1\ l_2 : t \triangleright \{\,\}, \{\,\}} \quad \text{CHECK\_EXP\_AUX\_VAR}$$

$$\frac{}{\Delta, E, E^{\mathrm{L}} \vdash N : \_\mathbf{num} \ \triangleright \{\,\}, \{\,\}} \quad \text{CHECK\_EXP\_AUX\_NVAR}$$

$$\frac{\begin{array}{l} E^{\mathrm{L}} \vdash id\ \mathbf{not\ shadowed} \\ E \vdash id\ \mathbf{value} \\ \Delta, E \vdash \mathbf{ctor}\ id : t\_multi \rightarrow p\ t\_args \triangleright (x\ \mathbf{of}\ names) \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash id : \mathbf{curry}\,(t\_multi, p\ t\_args) \triangleright \{\,\}, \{\,\}} \quad \text{CHECK\_EXP\_AUX\_CTOR}$$

$$\frac{\begin{array}{l} E^{\mathrm{L}} \vdash id\ \mathbf{not\ shadowed} \\ E \vdash id\ \mathbf{value} \\ \Delta, E \vdash \mathbf{val}\ id : t \triangleright \Sigma^{\mathcal{C}} \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash id : t \triangleright \Sigma^{\mathcal{C}}, \{\,\}} \quad \text{CHECK\_EXP\_AUX\_VAL}$$

$$\frac{\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash pat_1 : t_1 \triangleright E_1^{\mathrm{L}} \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t_n \triangleright E_n^{\mathrm{L}} \\ \Delta, E, E^{\mathrm{L}} \uplus E_1^{\mathrm{L}} \uplus \ldots \uplus E_n^{\mathrm{L}} \vdash exp : u \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \mathbf{disjoint\ doms}\,(E_1^{\mathrm{L}}, \ldots, E_n^{\mathrm{L}}) \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{fun}\ pat_1 \ldots pat_n \rightarrow exp\ l : \mathbf{curry}\,((t_1 * \ldots * t_n), u) \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_EXP\_AUX\_FN}$$

$$\frac{\begin{array}{l} \overline{\Delta, E, E^{\mathrm{L}} \vdash pat_i : t \triangleright E_i^{\mathrm{L}}}^{\,i} \\ \overline{\Delta, E, E^{\mathrm{L}} \uplus E_i^{\mathrm{L}} \vdash exp_i : u \triangleright \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{\,i} \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{function}\ |^{?}\ \overline{pat_i \rightarrow exp_i\ l_i}^{\,i}\ \mathbf{end} : t \rightarrow u \triangleright \overline{\Sigma^{\mathcal{C}}{}_i}^{\,i}, \overline{\Sigma^{\mathcal{N}}{}_i}^{\,i}} \quad \text{CHECK\_EXP\_AUX\_FUNCTION}$$

$$\frac{\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash exp_1 : t_1 \rightarrow t_2 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_2 : t_1 \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp_1\ exp_2 : t_2 \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2} \quad \text{CHECK\_EXP\_AUX\_APP}$$

$$\frac{\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash (ix) : t_1 \rightarrow t_2 \rightarrow t_3 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_1 : t_1 \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_2 : t_2 \triangleright \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_3 \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp_1\ ix\ l\ exp_2 : t_3 \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2 \cup \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2 \cup \Sigma^{\mathcal{N}}{}_3} \quad \text{CHECK\_EXP\_AUX\_INFIX\_APP1}$$

$$\frac{\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash x : t_1 \rightarrow t_2 \rightarrow t_3 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_1 : t_1 \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_2 : t_2 \triangleright \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_3 \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp_1\ `x`l\ exp_2 : t_3 \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2 \cup \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2 \cup \Sigma^{\mathcal{N}}{}_3} \quad \text{CHECK\_EXP\_AUX\_INFIX\_APP2}$$

$$\frac{\begin{array}{l}\overline{\Delta, E \vdash \mathbf{field}\ id_i : p\ t\_args \to t_i \triangleright (x_i\ \mathbf{of}\ names)}^{i}\\ \overline{\Delta, E, E^{\mathrm{L}} \vdash exp_i : t_i \triangleright \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{i}\\ \mathbf{duplicates}\,(\,\overline{x_i}^{i}\,) = \emptyset\\ names = \{\,\overline{x_i}^{i}\,\}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \langle|\overline{id_i = exp_i\ l_i}^{i} ;^?\ l|\rangle : p\ t\_args \triangleright \overline{\Sigma^{\mathcal{C}}{}_i}^{i}, \overline{\Sigma^{\mathcal{N}}{}_i}^{i}}\quad \text{CHECK\_EXP\_AUX\_RECORD}$$

$$\frac{\begin{array}{l}\overline{\Delta, E \vdash \mathbf{field}\ id_i : p\ t\_args \to t_i \triangleright (x_i\ \mathbf{of}\ names)}^{i}\\ \overline{\Delta, E, E^{\mathrm{L}} \vdash exp_i : t_i \triangleright \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{i}\\ \mathbf{duplicates}\,(\,\overline{x_i}^{i}\,) = \emptyset\\ \Delta, E, E^{\mathrm{L}} \vdash exp : p\ t\_args \triangleright \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \langle|exp\ \mathbf{with}\ \overline{id_i = exp_i\ l_i}^{i} ;^?\ l|\rangle : p\ t\_args \triangleright \Sigma^{\mathcal{C}\prime} \cup \overline{\Sigma^{\mathcal{C}}{}_i}^{i}, \Sigma^{\mathcal{N}\prime} \cup \overline{\Sigma^{\mathcal{N}}{}_i}^{i}}\quad \text{CHECK\_EXP\_AUX\_RECUP}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp_1 : t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash exp_n : t \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n\\ \mathbf{length}\,(exp_1 \ldots exp_n) = nat\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash [|exp_1; \ldots ; exp_n ;^?|] : \_\_\mathbf{vector}\ nat\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \ldots \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup \ldots \cup \Sigma^{\mathcal{N}}{}_n}\quad \text{CHECK\_EXP\_AUX\_VECTOR}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp : \_\_\mathbf{vector}\ ne'\ t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \vdash Nexp \rightsquigarrow ne\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp.(Nexp) : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \cup \{ne\langle ne'\}}\quad \text{CHECK\_EXP\_AUX\_VECTORGET}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp : \_\_\mathbf{vector}\ ne'\ t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \vdash Nexp_1 \rightsquigarrow ne_1\\ \vdash Nexp_2 \rightsquigarrow ne_2\\ ne = ne_2 + (-ne_1)\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp.(Nexp_1..Nexp_2) : \_\_\mathbf{vector}\ ne\ t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \cup \{ne_1\langle ne_2\langle ne'\}}\quad \text{CHECK\_EXP\_AUX\_VECTORSUB}$$

$$\frac{\begin{array}{l}E \vdash id\ \mathbf{field}\\ \Delta, E \vdash \mathbf{field}\ id : p\ t\_args \to t \triangleright (x\ \mathbf{of}\ names)\\ \Delta, E, E^{\mathrm{L}} \vdash exp : p\ t\_args \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp.id : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}\quad \text{CHECK\_EXP\_AUX\_FIELD}$$

$$\frac{\begin{array}{l}\overline{\Delta, E, E^{\mathrm{L}} \vdash pat_i : t \triangleright E^{\mathrm{L}}_i}^{i}\\ \overline{\Delta, E, E^{\mathrm{L}} \uplus E^{\mathrm{L}}_i \vdash exp_i : u \triangleright \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{i}\\ \Delta, E, E^{\mathrm{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{match}\ exp\ \mathbf{with}\ |^?\ \overline{pat_i \to exp_i\ l_i}^{i}\ l\ \mathbf{end} : u \triangleright \Sigma^{\mathcal{C}\prime} \cup \overline{\Sigma^{\mathcal{C}}{}_i}^{i}, \Sigma^{\mathcal{N}\prime} \cup \overline{\Sigma^{\mathcal{N}}{}_i}^{i}}\quad \text{CHECK\_EXP\_AUX\_CASE}$$

$$\frac{\begin{array}{c}\Delta, E, E^{\mathrm{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \Delta, E \vdash typ \rightsquigarrow t\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash (exp : typ) : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}\quad \text{CHECK\_EXP\_AUX\_TYPED}$$

$$\frac{\begin{array}{c}\Delta, E, E^{\mathrm{L}}_1 \vdash letbind \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1\\ \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_2 \vdash exp : t \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{let}\ letbind\ \mathbf{in}\ exp : t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2}\quad \text{CHECK\_EXP\_AUX\_LET}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash exp_1 : t_1 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad \ldots. \quad \Delta, E, E^{\mathrm{L}} \vdash exp_n : t_n \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n}{\Delta, E, E^{\mathrm{L}} \vdash (exp_1, \ldots., exp_n) : t_1 * \ldots. * t_n \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \ldots. \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup \ldots. \cup \Sigma^{\mathcal{N}}{}_n}\quad \text{CHECK\_EXP\_AUX\_TUP}$$

$$\frac{\begin{array}{l}\Delta \vdash t\ \mathbf{ok}\\ \Delta, E, E^{\mathrm{L}} \vdash exp_1 : t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad .. \quad \Delta, E, E^{\mathrm{L}} \vdash exp_n : t \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash [exp_1; ..; exp_n ;^?] : \_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup .. \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup .. \cup \Sigma^{\mathcal{N}}{}_n}\quad \text{CHECK\_EXP\_AUX\_LIST}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}} \vdash (exp) : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}\quad \text{CHECK\_EXP\_AUX\_PAREN}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{begin}\ exp\ \mathbf{end} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_EXP\_AUX\_BEGIN}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp_1 : \_\_\mathbf{bool} \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_2 : t \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_3 : t \triangleright \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_3\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{if}\ exp_1\ \mathbf{then}\ exp_2\ \mathbf{else}\ exp_3 : t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2 \cup \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2 \cup \Sigma^{\mathcal{N}}{}_3} \quad \text{CHECK\_EXP\_AUX\_IF}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp_1 : t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_2 : \_\_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp_1 :: exp_2 : \_\_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2} \quad \text{CHECK\_EXP\_AUX\_CONS}$$

$$\frac{\vdash lit : t}{\Delta, E, E^{\mathrm{L}} \vdash lit : t \triangleright \{\,\}, \{\,\}} \quad \text{CHECK\_EXP\_AUX\_LIT}$$

$$\frac{\begin{array}{l}\overline{\Delta \vdash t_i\ \mathbf{ok}}^{\,i} \\ \Delta, E, E^{\mathrm{L}} \uplus \{\,\overline{x_i \mapsto t_i}^{\,i}\,\} \vdash exp_1 : t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \uplus \{\,\overline{x_i \mapsto t_i}^{\,i}\,\} \vdash exp_2 : \_\_\mathbf{bool} \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\ \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}, \{\,\overline{x_i \mapsto t_i}^{\,i}\,\}) \\ E = \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \\ \overline{x_i \notin \mathbf{dom}\,(E^{\mathrm{X}})}^{\,i}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \{exp_1 | exp_2\} : \_\_\mathbf{set}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2} \quad \text{CHECK\_EXP\_AUX\_SET\_COMP}$$

$$\frac{\begin{array}{l}\Delta, E, E_1^{\mathrm{L}} \vdash \overline{qbind_i}^{\,i} \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}{}_1 \\ \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash exp_1 : t \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\ \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash exp_2 : \_\_\mathbf{bool} \triangleright \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_3\end{array}}{\Delta, E, E_1^{\mathrm{L}} \vdash \{exp_1 |\ \mathbf{forall}\ \overline{qbind_i}^{\,i}\ | exp_2\} : \_\_\mathbf{set}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2 \cup \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_2 \cup \Sigma^{\mathcal{N}}{}_3} \quad \text{CHECK\_EXP\_AUX\_SET\_COMP\_}$$

$$\frac{\begin{array}{l}\Delta \vdash t\ \mathbf{ok} \\ \Delta, E, E^{\mathrm{L}} \vdash exp_1 : t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad .. \quad \Delta, E, E^{\mathrm{L}} \vdash exp_n : t \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \{exp_1;\, ..;\, exp_n\, ;^?\} : \_\_\mathbf{set}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup .. \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup .. \cup \Sigma^{\mathcal{N}}{}_n} \quad \text{CHECK\_EXP\_AUX\_SET}$$

$$\frac{\begin{array}{l}\Delta, E, E_1^{\mathrm{L}} \vdash \overline{qbind_i}^{\,i} \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}{}_1 \\ \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash exp : \_\_\mathbf{bool} \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2\end{array}}{\Delta, E, E_1^{\mathrm{L}} \vdash q\ \overline{qbind_i}^{\,i}\,.exp : \_\_\mathbf{bool} \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2} \quad \text{CHECK\_EXP\_AUX\_QUANT}$$

$$\frac{\begin{array}{l}\Delta, E, E_1^{\mathrm{L}} \vdash \mathbf{list}\ \overline{qbind_i}^{\,i} \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}{}_1 \\ \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash exp_1 : t \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\ \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash exp_2 : \_\_\mathbf{bool} \triangleright \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_3\end{array}}{\Delta, E, E_1^{\mathrm{L}} \vdash [exp_1 |\ \mathbf{forall}\ \overline{qbind_i}^{\,i}\ | exp_2] : \_\_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2 \cup \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_2 \cup \Sigma^{\mathcal{N}}{}_3} \quad \text{CHECK\_EXP\_AUX\_LIST\_COMP\_}$$

$\boxed{\Delta, E, E_1^{\mathrm{L}} \vdash qbind_1\, ..\, qbind_n \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}}$ Build the environment for quantifier bindings, collecting typeclass cons

$$\frac{}{\Delta, E, E^{\mathrm{L}} \vdash\ \triangleright \{\,\}, \{\,\}} \quad \text{CHECK\_LISTQUANT\_BINDING\_EMPTY}$$

$$\frac{\begin{array}{l}\Delta \vdash t\ \mathbf{ok} \\ \Delta, E, E_1^{\mathrm{L}} \uplus \{x \mapsto t\} \vdash \overline{qbind_i}^{\,i} \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}{}_1 \\ \mathbf{disjoint\ doms}\,(\{x \mapsto t\}, E_2^{\mathrm{L}})\end{array}}{\Delta, E, E_1^{\mathrm{L}} \vdash x\ l\ \overline{qbind_i}^{\,i} \triangleright \{x \mapsto t\} \uplus E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}{}_1} \quad \text{CHECK\_LISTQUANT\_BINDING\_VAR}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}}_1 \vdash pat : t \triangleright E^{\mathrm{L}}_3\\ \Delta, E, E^{\mathrm{L}}_1 \vdash exp : \_\_\mathbf{set}\ t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1\\ \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_3 \vdash \overline{qbind_i}^{\,i} \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}{}_2\\ \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_3, E^{\mathrm{L}}_2)\end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash (pat\,\mathbf{IN}\ exp)\,\overline{qbind_i}^{\,i} \triangleright E^{\mathrm{L}}_2 \uplus E^{\mathrm{L}}_3, \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2} \quad \text{CHECK\_LISTQUANT\_BINDING\_RESTR}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}}_1 \vdash pat : t \triangleright E^{\mathrm{L}}_3\\ \Delta, E, E^{\mathrm{L}}_1 \vdash exp : \_\_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1\\ \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_3 \vdash \overline{qbind_i}^{\,i} \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}{}_2\\ \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_3, E^{\mathrm{L}}_2)\end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash (pat\,\mathbf{MEM}\ exp)\,\overline{qbind_i}^{\,i} \triangleright E^{\mathrm{L}}_2 \uplus E^{\mathrm{L}}_3, \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2} \quad \text{CHECK\_LISTQUANT\_BINDING\_LIST\_RESTR}$$

$\boxed{\Delta, E, E^{\mathrm{L}}_1 \vdash \mathbf{list}\ qbind_1\, ..\, qbind_n \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}}$ Build the environment for quantifier bindings, collecting typeclass c

$$\frac{}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{list} \triangleright \{\,\}, \{\,\}} \quad \text{CHECK\_QUANT\_BINDING\_EMPTY}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}}_1 \vdash pat : t \triangleright E^{\mathrm{L}}_3\\ \Delta, E, E^{\mathrm{L}}_1 \vdash exp : \_\_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1\\ \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_3 \vdash \overline{qbind_i}^{\,i} \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}{}_2\\ \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_3, E^{\mathrm{L}}_2)\end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash \mathbf{list}\ (pat\,\mathbf{MEM}\ exp)\,\overline{qbind_i}^{\,i} \triangleright E^{\mathrm{L}}_2 \uplus E^{\mathrm{L}}_3, \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2} \quad \text{CHECK\_QUANT\_BINDING\_RESTR}$$

$\boxed{\Delta, E, E^{\mathrm{L}} \vdash funcl \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ Build the environment for a function definition clause, collecting typecl

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash pat_1 : t_1 \triangleright E^{\mathrm{L}}_1 \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t_n \triangleright E^{\mathrm{L}}_n\\ \Delta, E, E^{\mathrm{L}} \uplus E^{\mathrm{L}}_1 \uplus \,\ldots\, \uplus E^{\mathrm{L}}_n \vdash exp : u \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_1, \,\ldots\,, E^{\mathrm{L}}_n)\\ \Delta, E \vdash typ \leadsto u\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash x\ l_1\ pat_1 \ldots pat_n\ : typ = exp\ l_2 \triangleright \{x \mapsto \mathbf{curry}\,((t_1 * \,\ldots\, * t_n), u)\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_FUNCL\_ANNOT}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash pat_1 : t_1 \triangleright E^{\mathrm{L}}_1 \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t_n \triangleright E^{\mathrm{L}}_n\\ \Delta, E, E^{\mathrm{L}} \uplus E^{\mathrm{L}}_1 \uplus \,\ldots\, \uplus E^{\mathrm{L}}_n \vdash exp : u \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_1, \,\ldots\,, E^{\mathrm{L}}_n)\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash x\ l_1\ pat_1 \ldots pat_n\ = exp\ l_2 \triangleright \{x \mapsto \mathbf{curry}\,((t_1 * \,\ldots\, * t_n), u)\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_FUNCL\_NOANNOT}$$

$\boxed{\Delta, E, E^{\mathrm{L}}_1 \vdash letbind \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ Build the environment for a let binding, collecting typeclass and index con

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}}_1 \vdash pat : t \triangleright E^{\mathrm{L}}_2\\ \Delta, E, E^{\mathrm{L}}_1 \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \Delta, E \vdash typ \leadsto t\end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash pat\ : typ = exp\ l \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_LETBIND\_VAL\_ANNOT}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}}_1 \vdash pat : t \triangleright E^{\mathrm{L}}_2\\ \Delta, E, E^{\mathrm{L}}_1 \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash pat\ = exp\ l \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_LETBIND\_VAL\_NOANNOT}$$

$$\frac{\Delta, E, E^{\mathrm{L}}_1 \vdash funcl\_aux\ l \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}}_1 \vdash funcl\_aux\ l \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_LETBIND\_FN}$$

$\boxed{\Delta, E, E^{\mathrm{L}} \vdash rule \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ Build the environment for an inductive relation clause, collecting typecl

$$\frac{\begin{array}{l}\overline{\Delta \vdash t_i\ \mathbf{ok}}^{\,i}\\ E_2^{\mathrm{L}} = \{\, \overline{y_i \mapsto t_i}^{\,i}\, \}\\ \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash \mathit{exp}' : \_\_\mathbf{bool}\ \triangleright \Sigma^{\mathcal{C}'}, \Sigma^{\mathcal{N}'}\\ \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash \mathit{exp}_1 : u_1 \triangleright {\Sigma^{\mathcal{C}}}_1, {\Sigma^{\mathcal{N}}}_1 \quad .. \quad \Delta, E, E_1^{\mathrm{L}} \uplus E_2^{\mathrm{L}} \vdash \mathit{exp}_n : u_n \triangleright {\Sigma^{\mathcal{C}}}_n, {\Sigma^{\mathcal{N}}}_n\end{array}}{\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{id}^{?}\ \mathbf{forall}\ \overline{y_i\ l_i}^{\,i}\ .\mathit{exp}' \Longrightarrow x\ l\ \mathit{exp}_1\ ..\ \mathit{exp}_n\ l' \triangleright \{x \mapsto \mathbf{curry}\,((u_1 * .. * u_n), \_\_\mathbf{bool}\,)\}, \Sigma^{\mathcal{C}'} \cup {\Sigma^{\mathcal{C}}}_1 \cup .. \cup \Sigma}$$

$\boxed{\mathit{xs}, \Delta_1, E \vdash\ \mathbf{tc}\ \mathit{td} \triangleright \Delta_2, E^{\mathrm{P}}}$ Extract the type constructor information

$$\frac{\begin{array}{l}\mathit{tnvars}^{l} \rightsquigarrow \mathit{tnvs}\\ \Delta, E \vdash \mathit{typ} \rightsquigarrow t\\ \mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset\\ \mathbf{FV}\,(t) \subset \mathit{tnvs}\\ \overline{y_i.}^{\,i}\ x \notin \mathbf{dom}\,(\Delta)\end{array}}{\overline{y_i}^{\,i}, \Delta, E \vdash\ \mathbf{tc}\ x\ l\ \mathit{tnvars}^{l} = \mathit{typ} \triangleright \{\overline{y_i.}^{\,i}\ x \mapsto \mathit{tnvs}\ .t\}, \{x \mapsto \overline{y_i.}^{\,i}\ x\}}\ \textsc{check\_texp\_tc\_abbrev}$$

$$\frac{\begin{array}{l}\mathit{tnvars}^{l} \rightsquigarrow \mathit{tnvs}\\ \mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset\\ \overline{y_i.}^{\,i}\ x \notin \mathbf{dom}\,(\Delta)\end{array}}{\overline{y_i}^{\,i}, \Delta, E_1 \vdash\ \mathbf{tc}\ x\ l\ \mathit{tnvars}^{l}\ \triangleright \{\overline{y_i.}^{\,i}\ x \mapsto \mathit{tnvs}\,\}, \{x \mapsto \overline{y_i.}^{\,i}\ x\}}\ \textsc{check\_texp\_tc\_abstract}$$

$$\frac{\begin{array}{l}\mathit{tnvars}^{l} \rightsquigarrow \mathit{tnvs}\\ \mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset\\ \overline{y_i.}^{\,i}\ x \notin \mathbf{dom}\,(\Delta)\end{array}}{\overline{y_i}^{\,i}, \Delta_1, E \vdash\ \mathbf{tc}\ x\ l\ \mathit{tnvars}^{l} = \langle| x_1^{l} : \mathit{typ}_1; \ldots; x_j^{l} : \mathit{typ}_j\ ;^{?} |\rangle \triangleright \{\overline{y_i.}^{\,i}\ x \mapsto \mathit{tnvs}\,\}, \{x \mapsto \overline{y_i.}^{\,i}\ x\}}\ \textsc{check\_texp\_tc\_rec}$$

$$\frac{\begin{array}{l}\mathit{tnvars}^{l} \rightsquigarrow \mathit{tnvs}\\ \mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset\\ \overline{y_i.}^{\,i}\ x \notin \mathbf{dom}\,(\Delta)\end{array}}{\overline{y_i}^{\,i}, \Delta_1, E \vdash\ \mathbf{tc}\ x\ l\ \mathit{tnvars}^{l} = |^{?}\ \mathit{ctor\_def}_1 | \ldots | \mathit{ctor\_def}_j \triangleright \{\overline{y_i.}^{\,i}\ x \mapsto \mathit{tnvs}\,\}, \{x \mapsto \overline{y_i.}^{\,i}\ x\}}\ \textsc{check\_texp\_tc\_var}$$

$\boxed{\mathit{xs}, \Delta_1, E \vdash\ \mathbf{tc}\ \mathit{td}_1\ ..\ \mathit{td}_i \triangleright \Delta_2, E^{\mathrm{P}}}$ Extract the type constructor information

$$\frac{}{\mathit{xs}, \Delta, E \vdash\ \mathbf{tc} \triangleright \{\,\}, \{\,\}}\ \textsc{check\_texps\_tc\_empty}$$

$$\frac{\begin{array}{l}\mathit{xs}, \Delta_1, E \vdash\ \mathbf{tc}\ \mathit{td} \triangleright \Delta_2, E_2^{\mathrm{P}}\\ \mathit{xs}, \Delta_1 \uplus \Delta_2, E \uplus \langle\{\,\}, E_2^{\mathrm{P}}, \{\,\}, \{\,\}\rangle \vdash\ \mathbf{tc}\ \overline{\mathit{td}_i}^{\,i} \triangleright \Delta_3, E_3^{\mathrm{P}}\\ \mathbf{dom}\,(E_2^{\mathrm{P}}) \cap \mathbf{dom}\,(E_3^{\mathrm{P}}) = \emptyset\end{array}}{\mathit{xs}, \Delta_1, E \vdash\ \mathbf{tc}\ \mathit{td}\ \overline{\mathit{td}_i}^{\,i} \triangleright \Delta_2 \uplus \Delta_3, E_2^{\mathrm{P}} \uplus E_3^{\mathrm{P}}}\ \textsc{check\_texps\_tc\_abbrev}$$

$\boxed{\Delta, E \vdash \mathit{tnvs}\ p = \mathit{texp} \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}}\rangle}$ Check a type definition, with its path already resolved

$$\frac{}{\Delta, E \vdash \mathit{tnvs}\ p = \mathit{typ} \triangleright \langle\{\,\}, \{\,\}\rangle}\ \textsc{check\_texp\_abbrev}$$

$$\frac{\begin{array}{l}\overline{\Delta, E \vdash \mathit{typ}_i \rightsquigarrow t_i}^{\,i}\\ \mathit{names} = \{\, \overline{x_i}^{\,i}\, \}\\ \mathbf{duplicates}\,(\, \overline{x_i}^{\,i}\,) = \emptyset\\ \overline{\mathbf{FV}\,(t_i) \subset \mathit{tnvs}}^{\,i}\\ E^{\mathrm{F}} = \{\, \overline{x_i \mapsto \langle\, \mathbf{forall}\ \mathit{tnvs}.p \to t_i, (x_i\ \mathbf{of}\ \mathit{names})\rangle}^{\,i}\, \}\end{array}}{\Delta, E \vdash \mathit{tnvs}\ p = \langle|\ \overline{x_i^{l} : \mathit{typ}_i}^{\,i}\ ;^{?} |\rangle \triangleright \langle E^{\mathrm{F}}, \{\,\}\rangle}\ \textsc{check\_texp\_rec}$$

$$\frac{\begin{array}{l}\overline{\Delta, E \vdash \mathit{typs}_i \rightsquigarrow \mathit{t\_multi}_i}^{\,i} \\ \mathit{names} = \{\, \overline{x_i}^{\,i} \,\} \\ \mathbf{duplicates}\,(\, \overline{x_i}^{\,i} \,) = \emptyset \\ \overline{\mathbf{FV}\,(\mathit{t\_multi}_i) \ \subset \ \mathit{tnvs}}^{\,i} \\ E^{\mathrm{X}} = \{\, \overline{x_i \mapsto \langle\, \mathbf{forall}\, \mathit{tnvs}.\mathit{t\_multi}_i \rightarrow p, (x_i\, \mathbf{of}\, \mathit{names})\rangle}^{\,i} \,\}\end{array}}{\Delta, E \vdash \mathit{tnvs}\ p = |^{?}\, \overline{x_i^l\, \mathbf{of}\, \mathit{typs}_i}^{\,i} \ \triangleright \langle \{\,\}, E^{\mathrm{X}}\rangle} \quad \text{CHECK\_TEXP\_VAR}$$

$\boxed{\mathit{xs}, \Delta, E \vdash \mathit{td}_1 \,..\, \mathit{td}_n \ \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}}\rangle}$

$$\frac{}{\overline{y_i}^{\,i}, \Delta, E \vdash \ \triangleright \langle \{\,\}, \{\,\}\rangle} \quad \text{CHECK\_TEXPS\_EMPTY}$$

$$\frac{\begin{array}{l}\mathit{tnvars}^l \rightsquigarrow \mathit{tnvs} \\ \Delta, E_1 \vdash \mathit{tnvs}\, \overline{y_i.}^{\,i}\, x = \mathit{texp} \ \triangleright \langle E_1^{\mathrm{F}}, E_1^{\mathrm{X}}\rangle \\ \overline{y_i}^{\,i}, \Delta, E \vdash \ \overline{\mathit{td}_j}^{\,j} \ \triangleright \langle E_2^{\mathrm{F}}, E_2^{\mathrm{X}}\rangle \\ \mathbf{dom}\,(E_1^{\mathrm{X}}) \cap \mathbf{dom}\,(E_2^{\mathrm{X}}) = \emptyset \\ \mathbf{dom}\,(E_1^{\mathrm{F}}) \cap \mathbf{dom}\,(E_2^{\mathrm{F}}) = \emptyset\end{array}}{\overline{y_i}^{\,i}, \Delta, E \vdash x\ l\ \mathit{tnvars}^l \ = \mathit{texp}\ \overline{\mathit{td}_j}^{\,j} \ \triangleright \langle E_1^{\mathrm{F}} \uplus E_2^{\mathrm{F}}, E_1^{\mathrm{X}} \uplus E_2^{\mathrm{X}}\rangle} \quad \text{CHECK\_TEXPS\_CONS\_CONCRETE}$$

$$\frac{\overline{y_i}^{\,i}, \Delta, E \vdash \ \overline{\mathit{td}_j}^{\,j} \ \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}}\rangle}{\overline{y_i}^{\,i}, \Delta, E \vdash x\ l\ \mathit{tnvars}^l \ \overline{\mathit{td}_j}^{\,j} \ \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}}\rangle} \quad \text{CHECK\_TEXPS\_CONS\_ABSTRACT}$$

$\boxed{\delta, E \vdash \mathit{id} \rightsquigarrow p}$ Lookup a type class

$$\frac{\begin{array}{c}E(\mathit{id}) \ \triangleright\ p \\ \delta(p) \ \triangleright\ \mathit{xs}\end{array}}{\delta, E \vdash \mathit{id} \rightsquigarrow p} \quad \text{CONVERT\_CLASS\_ALL}$$

$\boxed{I \vdash (p\ t)\, \mathbf{IN}\, \mathcal{C}}$ Solve class constraint

$$\frac{}{I \vdash (p\ \alpha)\, \mathbf{IN}\, (p_1\ \mathit{tnv}_1) \,..\, (p_i\ \mathit{tnv}_i)(p\ \alpha)(p'_1\ \mathit{tnv}'_1) \,..\, (p'_j\ \mathit{tnv}'_j)} \quad \text{SOLVE\_CLASS\_CONSTRAINT\_IMMEDIATE}$$

$$\frac{\begin{array}{l}(p_1\ \mathit{tnv}_1) \,..\, (p_n\ \mathit{tnv}_n) \Rightarrow (p\ t)\, \mathbf{IN}\, I \\ I \vdash (p_1\, \sigma(\mathit{tnv}_1))\, \mathbf{IN}\, \mathcal{C} \quad .. \quad I \vdash (p_n\, \sigma(\mathit{tnv}_n))\, \mathbf{IN}\, \mathcal{C}\end{array}}{I \vdash (p\, \sigma(t))\, \mathbf{IN}\, \mathcal{C}} \quad \text{SOLVE\_CLASS\_CONSTRAINT\_CHAIN}$$

$\boxed{I \vdash \Sigma^{\mathcal{C}} \ \triangleright\ \mathcal{C}}$ Solve class constraints

$$\frac{I \vdash (p_1\ t_1)\, \mathbf{IN}\, \mathcal{C} \quad .. \quad I \vdash (p_n\ t_n)\, \mathbf{IN}\, \mathcal{C}}{I \vdash \{(p_1\ t_1), \,..\,, (p_n\ t_n)\} \ \triangleright\ \mathcal{C}} \quad \text{SOLVE\_CLASS\_CONSTRAINTS\_ALL}$$

$\boxed{\Delta, I, E \vdash \mathit{val\_def} \ \triangleright\ E^{\mathrm{X}}}$ Check a value definition

$$\frac{\begin{array}{l}\Delta, E, \{\,\} \vdash \mathit{letbind} \ \triangleright\ \{\, \overline{x_i \mapsto t_i}^{\,i} \,\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ I \vdash \Sigma^{\mathcal{C}} \ \triangleright\ \mathcal{C} \\ \overline{\mathbf{FV}\,(t_i) \ \subset \ \mathit{tnvs}}^{\,i} \\ \mathbf{FV}\,(\mathcal{C}) \ \subset \ \mathit{tnvs}\end{array}}{\Delta, I, E_1 \vdash \mathbf{let}\, \tau^{?}\ \mathit{letbind} \ \triangleright\ \{\, \overline{x_i \mapsto \langle\, \mathbf{forall}\, \mathit{tnvs}.\mathcal{C} \Rightarrow t_i, \mathbf{let}\rangle}^{\,i} \,\}} \quad \text{CHECK\_VAL\_DEF\_VAL}$$

$$\frac{\begin{array}{l}\overline{\Delta, E, E^{\mathrm{L}} \vdash \mathit{funcl}_i \triangleright \{x_i \mapsto t_i\}, \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{i}\\ I \vdash \Sigma^{\mathcal{C}} \triangleright \mathcal{C}\\ \overline{\mathbf{FV}\,(t_i) \subset \mathit{tnvs}}^{i}\\ \mathbf{FV}\,(\mathcal{C}) \subset \mathit{tnvs}\\ \mathbf{compatible\,overlap}\,(\overline{x_i \mapsto t_i}^{i})\\ E^{\mathrm{L}} = \{\overline{x_i \mapsto t_i}^{i}\}\end{array}}{\Delta, I, E \vdash \mathbf{let\,rec}\,\tau^{?}\,\overline{\mathit{funcl}_i}^{i} \triangleright \{\overline{x_i \mapsto \langle\,\mathbf{forall}\,\mathit{tnvs}.\mathcal{C} \Rightarrow t_i, \mathbf{let}\rangle}^{i}\}} \quad \text{CHECK\_VAL\_DEF\_RECFUN}$$

$\boxed{\Delta, (\alpha_1, .., \alpha_n) \vdash t\,\mathbf{instance}}$ Check that t be a typeclass instance

$$\frac{}{\Delta, (\alpha) \vdash \alpha\,\mathbf{instance}} \quad \text{CHECK\_T\_INSTANCE\_VAR}$$

$$\frac{}{\Delta, (\alpha_1, ..., \alpha_n) \vdash \alpha_1 * .... * \alpha_n\,\mathbf{instance}} \quad \text{CHECK\_T\_INSTANCE\_TUP}$$

$$\frac{}{\Delta, (\alpha_1, \alpha_2) \vdash \alpha_1 \rightarrow \alpha_n\,\mathbf{instance}} \quad \text{CHECK\_T\_INSTANCE\_FN}$$

$$\frac{\Delta(p) \triangleright \alpha'_1 .. \alpha'_n}{\Delta, (\alpha_1, .., \alpha_n) \vdash p\,\alpha_1 .. \alpha_n\,\mathbf{instance}} \quad \text{CHECK\_T\_INSTANCE\_TC}$$

$\boxed{\overline{z_j}^{j}, D_1, E_1 \vdash \mathit{def} \triangleright D_2, E_2}$ Check a definition

$$\frac{\begin{array}{l}\overline{z_j}^{j}, \Delta_1, E \vdash \mathbf{tc}\,\overline{\mathit{td}_i}^{i} \triangleright \Delta_2, E^{\mathrm{P}}\\ \overline{z_j}^{j}, \Delta_1 \uplus \Delta_2, E \uplus \langle\{\}, E^{\mathrm{P}}, \{\}, \{\}\rangle \vdash \overline{\mathit{td}_i}^{i} \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}}\rangle\end{array}}{\overline{z_j}^{j}, \langle\Delta_1, \delta, I\rangle, E \vdash \mathbf{type}\,\overline{\mathit{td}_i}^{i}\,l \triangleright \langle\Delta_2, \{\}, \{\}\rangle, \langle\{\}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle} \quad \text{CHECK\_DEF\_TYPE}$$

$$\frac{\Delta, I, E \vdash \mathit{val\_def} \triangleright E^{\mathrm{X}}}{\overline{z_j}^{j}, \langle\Delta, \delta, I\rangle, E \vdash \mathit{val\_def}\,l \triangleright \epsilon, \langle\{\}, \{\}, \{\}, E^{\mathrm{X}}\rangle} \quad \text{CHECK\_DEF\_VAL\_DEF}$$

$$\frac{\begin{array}{l}\overline{\Delta, E_1, E^{\mathrm{L}} \vdash \mathit{rule}_i \triangleright \{x_i \mapsto t_i\}, \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{i}\\ I \vdash \overline{\Sigma^{\mathcal{C}}{}_i}^{i} \triangleright \mathcal{C}\\ \overline{\mathbf{FV}\,(t_i) \subset \mathit{tnvs}}^{i}\\ \mathbf{FV}\,(\mathcal{C}) \subset \mathit{tnvs}\\ \mathbf{compatible\,overlap}\,(\overline{x_i \mapsto t_i}^{i})\\ E^{\mathrm{L}} = \{\overline{x_i \mapsto t_i}^{i}\}\\ E_2 = \langle\{\}, \{\}, \{\}, \{\overline{x_i \mapsto \langle\,\mathbf{forall}\,\mathit{tnvs}.\mathcal{C} \Rightarrow t_i, \mathbf{let}\rangle}^{i}\}\rangle\end{array}}{\overline{z_j}^{j}, \langle\Delta, \delta, I\rangle, E_1 \vdash \mathbf{indreln}\,\tau^{?}\,\overline{\mathit{rule}_i}^{i}\,l \triangleright \epsilon, E_2} \quad \text{CHECK\_DEF\_INDRELN}$$

$$\frac{\overline{z_j}^{j}\,x, D_1, E_1 \vdash \mathit{defs} \triangleright D_2, E_2}{\overline{z_j}^{j}, D_1, E_1 \vdash \mathbf{module}\,x\,l_1 = \mathbf{struct}\,\mathit{defs}\,\mathbf{end}\,l_2 \triangleright D_2, \langle\{x \mapsto E_2\}, \{\}, \{\}, \{\}\rangle} \quad \text{CHECK\_DEF\_MODULE}$$

$$\frac{E_1(\mathit{id}) \triangleright E_2}{\overline{z_j}^{j}, D, E_1 \vdash \mathbf{module}\,x\,l_1 = \mathit{id}\,l_2 \triangleright \epsilon, \langle\{x \mapsto E_2\}, \{\}, \{\}, \{\}\rangle} \quad \text{CHECK\_DEF\_MODULE\_RENAME}$$

$$\frac{\begin{array}{l}\Delta, E \vdash \mathit{typ} \rightsquigarrow t\\ \mathbf{FV}\,(t) \subset \overline{\alpha_i}^{i}\\ \mathbf{FV}\,(\overline{\alpha'_k}^{k}) \subset \overline{\alpha_i}^{i}\\ \overline{\delta, E \vdash \mathit{id}_k \rightsquigarrow p_k}^{k}\\ E' = \langle\{\}, \{\}, \{\}, \{x \mapsto \langle\,\mathbf{forall}\,\overline{\alpha_i}^{i}.\overline{(p_k\,\alpha'_k)}^{k} \Rightarrow t, \mathbf{val}\rangle\}\rangle\end{array}}{\overline{z_j}^{j}, \langle\Delta, \delta, I\rangle, E \vdash \mathbf{val}\,x\,l_1 : \mathbf{forall}\,\overline{\alpha_i\,l''_i}^{i}.\overline{\mathit{id}_k\,\alpha'_k\,l'_k}^{k} \Rightarrow \mathit{typ}\,l_2 \triangleright \epsilon, E'} \quad \text{CHECK\_DEF\_SPEC}$$

$$\frac{\begin{array}{l}\overline{\Delta, E_1 \vdash typ_i \rightsquigarrow t_i}^{\,i}\\ \overline{\mathbf{FV}\,(t_i) \subset \alpha}^{\,i}\\ p = \overline{z_j.}^{\,j}\, x\\ E_2 = \langle \{\,\}, \{x \mapsto p\}, \{\,\}, \{\overline{y_i \mapsto \langle\,\mathbf{forall}\,\alpha.(p\,\alpha) \Rightarrow t_i, \mathbf{method}\rangle}^{\,i}\}\rangle\\ \delta_2 = \{p \mapsto \overline{y_i}^{\,i}\}\\ p \notin \mathbf{dom}\,(\delta_1)\end{array}}{\overline{z_j}^{\,j}, \langle \Delta, \delta_1, I\rangle, E_1 \vdash \mathbf{class}\,(x\,l\,\alpha\,l'')\,\overline{\mathbf{val}\,y_i\,l_i : typ_i\,l_i}^{\,i}\,\mathbf{end}\,l' \rhd \langle \{\,\}, \delta_2, \{\,\}\rangle, E_2}\quad \text{CHECK\_DEF\_CLASS}$$

$$\frac{\begin{array}{l}E = \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle\\ \Delta, E \vdash typ' \rightsquigarrow t'\\ \Delta, (\,\overline{\alpha_i}^{\,i}\,) \vdash t'\,\mathbf{instance}\\ tnvs = \overline{\alpha_i}^{\,i}\\ \mathbf{duplicates}\,(tnvs) = \emptyset\\ \overline{\delta, E \vdash id_k \rightsquigarrow p_k}^{\,k}\\ \mathbf{FV}\,(\overline{\alpha'_k}^{\,k}) \subset tnvs\\ E(id) \rhd p\\ \delta(p) \rhd \overline{z_j}^{\,j}\\ I_2 = \{\,\overline{\Rightarrow (p_k\,\alpha'_k)}^{\,k}\,\}\\ \overline{\Delta, I \cup I_2, E \vdash val\_def_n \rhd E^{\mathrm{X}}_n}^{\,n}\\ \mathbf{disjoint\ doms}\,(\,\overline{E^{\mathrm{X}}_n}^{\,n}\,)\\ \overline{E^{\mathrm{X}}(x_k) \rhd \langle\,\mathbf{forall}\,\alpha''.(p\,\alpha'') \Rightarrow t_k, \mathbf{method}\rangle}^{\,k}\\ \{\,\overline{x_k \mapsto \langle\,\mathbf{forall}\,tnvs.\, \Rightarrow \{\alpha'' \mapsto t'\}(t_k), \mathbf{let}\rangle}^{\,k}\,\} = \overline{E^{\mathrm{X}}_n}^{\,n}\\ \overline{x_k}^{\,k} = \overline{z_j}^{\,j}\\ I_3 = \{\,\overline{(p_k\,\alpha'_k) \Rightarrow (p\,t')}^{\,k}\,\}\\ (p\,\{\,\overline{\alpha_i \mapsto \alpha'''_i}^{\,i}\,\}(t')) \notin I\end{array}}{\overline{z_j}^{\,j}, \langle \Delta, \delta, I\rangle, E \vdash \mathbf{instance}\,\mathbf{forall}\,\overline{\alpha_i\,l'_i}^{\,i}\,\overline{.id_k\,\alpha'_k\,l''_k}^{\,k} \Rightarrow (id\,typ')\,\overline{val\_def_n\,l_n}^{\,n}\,\mathbf{end}\,l' \rhd \langle \{\,\}, \{\,\}, I_3\rangle, \epsilon}\quad \text{CHECK\_DEF\_}$$

$\boxed{\overline{z_j}^{\,j}, D_1, E_1 \vdash defs \rhd D_2, E_2}$ Check definitions, given module path, definitions and environment

$$\frac{}{\overline{z_j}^{\,j}, D, E \vdash\ \rhd \epsilon, \epsilon}\quad \text{CHECK\_DEFS\_EMPTY}$$

$$\frac{\begin{array}{c}\overline{z_j}^{\,j}, D_1, E_1 \vdash def \rhd D_2, E_2\\ \overline{z_j}^{\,j}, D_1 \uplus D_2, E_1 \uplus E_2 \vdash \overline{def_i\,;;^?_i}^{\,i} \rhd D_3, E_3\end{array}}{\overline{z_j}^{\,j}, D_1, E_1 \vdash def\,;;^?\,\overline{def_i\,;;^?_i}^{\,i} \rhd D_2 \uplus D_3, E_2 \uplus E_3}\quad \text{CHECK\_DEFS\_RELEVANT\_DEF}$$

$$\frac{\begin{array}{c}E_1(id) \rhd E_2\\ \overline{z_j}^{\,j}, D_1, E_1 \uplus E_2 \vdash \overline{def_i\,;;^?_i}^{\,i} \rhd D_3, E_3\end{array}}{\overline{z_j}^{\,j}, D_1, E_1 \vdash \mathbf{open}\,id\,l\,;;^?\,\overline{def_i\,;;^?_i}^{\,i} \rhd D_3, E_3}\quad \text{CHECK\_DEFS\_OPEN}$$

```
Definition rules:        145 good    0 bad
Definition rule clauses: 439 good    0 bad
```