n, i, j, k	Index variables for meta-lists
num	Numeric literals
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
	$\mid$		
$x^l,\ y^l,\ z^l,\ name$	$::=$		Location-annotated names
	$\mid$	$x\,l$	
	$\mid$	$(ix)l$	Remove infix status
ix^l	$::=$		Location-annotated infix names
	$\mid$	$ix\,l$	
	$\mid$	$`x`l$	Add infix status
α	$::=$		Type variables
	$\mid$	$'x$	
α^l	$::=$		Location-annotated type variables
	$\mid$	$\alpha\,l$	
N	$::=$		numeric variables
	$\mid$	$''x$	
N^l	$::=$		Location-annotated numeric variables
	$\mid$	$N\,l$	
id	$::=$		Long identifers
	$\mid$	$x_1^l.\,..\,x_n^l.x^l\ l$	
tnv	$::=$		Union of type variables and Nexp type variables, without loca
	$\mid$	α	
	$\mid$	N	
$tnvar^l$	$::=$		Union of type variables and Nexp type variables, with location
	$\mid$	α^l	
	$\mid$	N^l	
$tnvs$	$::=$		Type variable lists
	$\mid$	$tnv_1\,..\,tnv_n$	
$tnvars^l$	$::=$		Type variable lists
	$\mid$	$tnvar_1^l\,..\,tnvar_n^l$	
$Nexp\_aux$	$::=$		Numerical expressions for specifying vector lengths and indexe
	$\mid$	N	
	$\mid$	num	
	$\mid$	$Nexp_1 * Nexp_2$	
	$\mid$	$Nexp_1 + Nexp_2$	
	$\mid$	$(Nexp)$	

Nexp	$::=$		Location-annotated vector lengths
	$\vert$	$\mathit{Nexp\_aux}\ l$	
$\mathit{Nexp\_constraint}$	$::=$		Whether a vector is bounded or fixed size
	$\vert$	Nexp	
	$\vert$	$\geq \mathit{Nexp}$	
$\mathit{typ\_aux}$	$::=$		Types
	$\vert$	$\_$	Unspecified type
	$\vert$	α^{l}	Type variables
	$\vert$	$\mathit{typ}_1 \to \mathit{typ}_2$	Function types
	$\vert$	$\mathit{typ}_1 * \ldots. * \mathit{typ}_n$	Tuple types
	$\vert$	Nexp	As a typ to permit applications over Nexps, otherwise no
	$\vert$	$\mathit{id}\ \mathit{typ}_1 \mathinner{..} \mathit{typ}_n$	Type applications
	$\vert$	(typ)	
typ	$::=$		Location-annotated types
	$\vert$	$\mathit{typ\_aux}\ l$	
$\mathit{lit\_aux}$	$::=$		Literal constants
	$\vert$	**true**	
	$\vert$	**false**	
	$\vert$	num	
	$\vert$	hex	hex and bin are constant bit vectors, entered as C-style
	$\vert$	bin	
	$\vert$	string	
	$\vert$	$()$	
	$\vert$	**bitzero**	bitzero and bitone are constant bits, if commonly used w
	$\vert$	**bitone**	
lit	$::=$		
	$\vert$	$\mathit{lit\_aux}\ l$	Location-annotated literal constants
$;^{?}$	$::=$		Optional semi-colons
	$\vert$		
	$\vert$	$;$	
$\mathit{pat\_aux}$	$::=$		Patterns
	$\vert$	$\_$	Wildcards
	$\vert$	$(\mathit{pat}\ \mathbf{as}\ x^{l})$	Named patterns
	$\vert$	$(\mathit{pat} : \mathit{typ})$	Typed patterns
	$\vert$	$\mathit{id}\ \mathit{pat}_1 \mathinner{..} \mathit{pat}_n$	Single variable and constructor patterns
	$\vert$	$\langle\vert \mathit{fpat}_1 ; \ldots ; \mathit{fpat}_n\ ;^{?} \vert\rangle$	Record patterns
	$\vert$	$[\vert \mathit{pat}_1 ; \mathinner{..} ; \mathit{pat}_n\ ;^{?} \vert]$	Vector patterns
	$\vert$	$[\vert \mathit{pat}_1 \mathinner{..} \mathit{pat}_n \vert]$	Concatenated vector patterns
	$\vert$	$(\mathit{pat}_1 , \ldots. , \mathit{pat}_n)$	Tuple patterns
	$\vert$	$[\mathit{pat}_1 ; \mathinner{..} ; \mathit{pat}_n\ ;^{?}]$	List patterns

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

q	::=		Quantifiers
	|	**forall**	
	|	**exists**	
$qbind$	::=		Bindings for quantifiers
	|	x^l	
	|	$(pat\,\mathbf{IN}\,exp)$	Restricted quantifications over sets
	|	$(pat\,\mathbf{MEM}\,exp)$	Restricted quantifications over lists
$fexp$	::=		Field-expressions
	|	$id = exp\ l$	
$fexps$	::=		Field-expression lists
	|	$fexp_1; \ldots; fexp_n\ ;^?\ l$	
$pexp$	::=		Pattern matches
	|	$pat \to exp\ l$	
$psexp$	::=		Multi-pattern matches
	|	$pat_1 \ldots pat_n \to exp\ l$	
$tannot^?$	::=		Optional type annotations
	|		
	|	$:\ typ$	
$funcl\_aux$	::=		Function clauses
	|	$x^l\ pat_1 \ldots pat_n\ tannot^? = exp$	
$letbind\_aux$	::=		Let bindings
	|	$pat\ tannot^? = exp$	Value bindings
	|	$funcl\_aux$	Function bindings
$letbind$	::=		Location-annotated let bindings
	|	$letbind\_aux\ l$	
$funcl$	::=		Location-annotated function clauses
	|	$funcl\_aux\ l$	
$id^?$	::=		Optional name for inductively defined relatio
	|		
	|	$x^l\ :$	
$rule\_aux$	::=		Inductively defined relation clauses
	|	$id^?\ \mathbf{forall}\ x_1^l \mathinner{..} x_n^l . exp \Longrightarrow x^l\ exp_1 \mathinner{..} exp_i$	
$rule$	::=		Location-annotated inductively defined relati
	|	$rule\_aux\ l$	

typs	::=			Type lists
	|	$typ_1 * \ldots * typ_n$		
ctor_def	::=			Datatype definition clauses
	|	x^l **of** *typs*		
	|	x^l	S	Constant constructors
texp	::=			Type definition bodies
	|	*typ*		Type abbreviations
	|	$\langle\| x_1^l : typ_1 ; \ldots ; x_n^l : typ_n \, ;^? \|\rangle$		Record types
	|	$\|^? \, ctor\_def_1 \| \ldots \| ctor\_def_n$		Variant types
$name^?$	::=			Optional name specification for variables of defined type
	|			
	|	$[name = regexp]$		
td	::=			Type definitions
	|	$x^l \, tnvars^l \, name^? = texp$		
	|	$x^l \, tnvars^l \, name^?$		Definitions of opaque types
c	::=			Typeclass constraints
	|	$id \, tnvar^l$		
cs	::=			Typeclass constraint lists
	|			
	|	$c_1 , \ldots , c_i \Rightarrow$		Must have $>$ 0 constraints
c_pre	::=			Type and instance scheme prefixes
	|			
	|	**forall** $tnvar_1^l \ldots tnvar_n^l . cs$		Must have $>$ 0 type variables
typschm	::=			Type schemes
	|	$c\_pre \, typ$		
instschm	::=			Instance schemes
	|	$c\_pre (id \, typ)$		
target	::=			Backend target names
	|	**hol**		
	|	**isabelle**		
	|	**ocaml**		
	|	**coq**		
	|	**tex**		
	|	**html**		
τ	::=			Backend target name lists
	|	$\{ target_1 ; \ldots ; target_n \}$		

$\tau^?$	::=		Optional targets
	|		
	|	τ	
lemma_typ	::=		Types of Lemmata
	|	**assert**	
	|	**lem**	
	|	**thm**	
lemma	::=		Lemmata and Tests
	|	$\mathit{lemma\_typ}\ x^l : \mathit{exp}$	
	|	$\mathit{lemma\_typ}\ \mathit{exp}$	
val_def	::=		Value definitions
	|	$\mathbf{let}\ \tau^?\ \mathit{letbind}$	Non-recursive value definition
	|	$\mathbf{let\ rec}\ \tau^?\ \mathit{funcl}_1\ \mathbf{and} \ldots \mathbf{and}\ \mathit{funcl}_n$	Recursive function definitions
	|	$\mathbf{let\ inline}\ \tau^?\ \mathit{letbind}$	Function definitions to be inli
val_spec	::=		Value type specifications
	|	$\mathbf{val}\ x^l : \mathit{typschm}$	
def_aux	::=		Top-level definitions
	|	$\mathbf{type}\ \mathit{td}_1\ \mathbf{and} \ldots \mathbf{and}\ \mathit{td}_n$	Type definitions
	|	*val_def*	Value definitions
	|	*lemma*	Lemmata
	|	$\mathbf{rename}\ \tau^?\ \mathit{id} = x^l$	Rename constant or type
	|	$\mathbf{module}\ x^l = \mathbf{struct}\ \mathit{defs}\ \mathbf{end}$	Module definitions
	|	$\mathbf{module}\ x^l = \mathit{id}$	Module renamings
	|	$\mathbf{open}\ \mathit{id}$	Opening modules
	|	$\mathbf{indreln}\ \tau^?\ \mathit{rule}_1\ \mathbf{and} \ldots \mathbf{and}\ \mathit{rule}_n$	Inductively defined relations
	|	*val_spec*	Top-level type constraints
	|	$\mathbf{class}\ (x^l\ \mathit{tnvar}^l)\ \mathbf{val}\ x_1^l : \mathit{typ}_1\ l_1 \ldots \mathbf{val}\ x_n^l : \mathit{typ}_n\ l_n\ \mathbf{end}$	Typeclass definitions
	|	$\mathbf{instance}\ \mathit{instschm}\ \mathit{val\_def}_1\ l_1 \ldots \mathit{val\_def}_n\ l_n\ \mathbf{end}$	Typeclass instantiations
def	::=		Location-annotated definitions
	|	$\mathit{def\_aux}\ l$	
$;;^?$	::=		Optional double-semi-colon
	|		
	|	$;;$	
defs	::=		Definition sequences
	|	$\mathit{def}_1\ ;;^?_1 \ldots \mathit{def}_n\ ;;^?_n$	
p	::=		Unique paths
	|	$x_1 . \ldots x_n . x$	
	|	**__list**	

	|	**__bool**		
	|	**__num**		
	|	**__set**		
	|	**__string**		
	|	**__unit**		
	|	**__bit**		
	|	**__vector**		
σ	::=			Type variable substitutions
	|	$\{ tnv_1 \mapsto t_1 \mathrel{..} tnv_n \mapsto t_n \}$		
$t,\ u$	::=			Internal types
	|	α		
	|	$t_1 \rightarrow t_2$		
	|	$t_1 * \ .... \ * t_n$		
	|	$p\ t\_args$		
	|	ne		
	|	$\sigma(t)$	M	Multiple substitutions
	|	$\sigma(tnv)$	M	Single variable substitution
	|	**curry** (t_multi, t)	M	Curried, multiple argument functions
ne	::=			internal numeric expressions
	|	N		
	|	num		
	|	$num * ne$		
	|	$ne_1 + ne_2$		
	|	$(-ne)$		
	|	**normalize** (ne)	M	
	|	$ne_1 + \ ... \ + ne_n$	M	
	|	**bitlength** (bin)	M	
	|	**bitlength** (hex)	M	
	|	**length** $(pat_1 \ ... \ pat_n)$	M	
	|	**length** $(exp_1 \ ... \ exp_n)$	M	
t_args	::=			Lists of types
	|	$t_1 \mathrel{..} t_n$		
	|	$\sigma(t\_args)$	M	Multiple substitutions
t_multi	::=			Lists of types
	|	$(t_1 * \mathrel{..} * t_n)$		
	|	$\sigma(t\_multi)$	M	Multiple substitutions
nec	::=			Numeric expression constraints
	|	$ne \langle nec$		
	|	$ne = nec$		
	|	$ne <= nec$		
	|	ne		

names	$::=$			Sets of names
	$\vert$	$\{x_1, \,..\,, x_n\}$		
$\mathcal{C}$	$::=$			Typeclass constraint lists
	$\vert$	$(p_1\ \mathit{tnv}_1) \,..\, (p_n\ \mathit{tnv}_n)$		
$\mathit{env\_tag}$	$::=$			Tags for the (non-constructor) value descripti
	$\vert$	**method**		Bound to a method
	$\vert$	**val**		Specified with val
	$\vert$	**let**		Defined with let or indreln
$\mathit{v\_desc}$	$::=$			Value descriptions
	$\vert$	$\langle\, \mathbf{forall}\, \mathit{tnvs}.\mathit{t\_multi} \rightarrow p, (x\ \mathbf{of}\ \mathit{names})\rangle$		Constructors
	$\vert$	$\langle\, \mathbf{forall}\, \mathit{tnvs}.\mathcal{C} \Rightarrow t, \mathit{env\_tag}\rangle$		Values
$\mathit{f\_desc}$	$::=$			
	$\vert$	$\langle\, \mathbf{forall}\, \mathit{tnvs}.p \rightarrow t, (x\ \mathbf{of}\ \mathit{names})\rangle$		Fields
$\Sigma^{\mathcal{C}}$	$::=$			Typeclass constraints
	$\vert$	$\{(p_1\ t_1), \,..\,, (p_n\ t_n)\}$		
	$\vert$	${\Sigma^{\mathcal{C}}}_1 \cup \,..\, \cup {\Sigma^{\mathcal{C}}}_n$	M	
$\Sigma^{\mathcal{N}}$	$::=$			Nexp constraint lists
	$\vert$	$\{\mathit{nec}_1, \,..\,, \mathit{nec}_n\}$		
	$\vert$	${\Sigma^{\mathcal{N}}}_1 \cup \,..\, \cup {\Sigma^{\mathcal{N}}}_n$	M	
E	$::=$			Environments
	$\vert$	$\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle$		
	$\vert$	$E_1 \uplus E_2$	M	
	$\vert$	ϵ	M	
E^{X}	$::=$			Value environments
	$\vert$	$\{x_1 \mapsto \mathit{v\_desc}_1, \,..\,, x_n \mapsto \mathit{v\_desc}_n\}$		
	$\vert$	$E^{\mathrm{X}}_1 \uplus \,..\, \uplus E^{\mathrm{X}}_n$	M	
E^{F}	$::=$			Field environments
	$\vert$	$\{x_1 \mapsto \mathit{f\_desc}_1, \,..\,, x_n \mapsto \mathit{f\_desc}_n\}$		
	$\vert$	$E^{\mathrm{F}}_1 \uplus \,..\, \uplus E^{\mathrm{F}}_n$	M	
E^{M}	$::=$			Module environments
	$\vert$	$\{x_1 \mapsto E_1, \,..\,, x_n \mapsto E_n\}$		
E^{P}	$::=$			Path environments
	$\vert$	$\{x_1 \mapsto p_1, \,..\,, x_n \mapsto p_n\}$		
	$\vert$	$E^{\mathrm{P}}_1 \uplus \,..\, \uplus E^{\mathrm{P}}_n$	M	
E^{L}	$::=$			Lexical bindings

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

	|	$\rangle$	
	|	$\vdash$	
	|	$,$	
	|	$\mapsto$	
	|	$\triangleright$	
	|	$\leadsto$	
	|	$\Rightarrow$	
	|	$-$	
	|	ϵ	
formula	$::=$		
	|	*judgement*	
	|	$\mathit{formula}_1 \quad .. \quad \mathit{formula}_n$	
	|	$E^{\text{M}}(x) \triangleright E$	Module lookup
	|	$E^{\text{P}}(x) \triangleright p$	Path lookup
	|	$E^{\text{F}}(x) \triangleright f\_desc$	Field lookup
	|	$E^{\text{X}}(x) \triangleright v\_desc$	Value lookup
	|	$E^{\text{L}}(x) \triangleright t$	Lexical binding lookup
	|	$\Delta(p) \triangleright tc\_def$	Type constructor lookup
	|	$\delta(p) \triangleright xs$	Type constructor lookup
	|	$\mathbf{dom}\,(E_1^{\text{M}}) \cap \mathbf{dom}\,(E_2^{\text{M}}) = \emptyset$	
	|	$\mathbf{dom}\,(E_1^{\text{X}}) \cap \mathbf{dom}\,(E_2^{\text{X}}) = \emptyset$	
	|	$\mathbf{dom}\,(E_1^{\text{F}}) \cap \mathbf{dom}\,(E_2^{\text{F}}) = \emptyset$	
	|	$\mathbf{dom}\,(E_1^{\text{P}}) \cap \mathbf{dom}\,(E_2^{\text{P}}) = \emptyset$	
	|	$\mathbf{disjoint\,doms}\,(E_1^{\text{L}}, \,....\,, E_n^{\text{L}})$	Pairwise disjoint domains
	|	$\mathbf{disjoint\,doms}\,(E_1^{\text{X}}, \,....\,, E_n^{\text{X}})$	Pairwise disjoint domains
	|	$\mathbf{compatible\,overlap}\,(x_1 \mapsto t_1, \,..\,, x_n \mapsto t_n)$	$(x_i = x_j) \Longrightarrow (t_i = t_j)$
	|	$\mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset$	
	|	$\mathbf{duplicates}\,(x_1, \,..\,, x_n) = \emptyset$	
	|	$x \notin \mathbf{dom}\,(E^{\text{L}})$	
	|	$x \notin \mathbf{dom}\,(E^{\text{X}})$	
	|	$x \notin \mathbf{dom}\,(E^{\text{F}})$	
	|	$p \notin \mathbf{dom}\,(\delta)$	
	|	$p \notin \mathbf{dom}\,(\Delta)$	
	|	$\mathbf{FV}\,(t) \subset \mathit{tnvs}$	Free type variables
	|	$\mathbf{FV}\,(t\_multi) \subset \mathit{tnvs}$	Free type variables
	|	$\mathbf{FV}\,(\mathcal{C}) \subset \mathit{tnvs}$	Free type variables
	|	$\mathit{inst}\,\mathbf{IN}\,I$	
	|	$(p\ t) \notin I$	
	|	$E_1^{\text{L}} = E_2^{\text{L}}$	
	|	$E_1^{\text{X}} = E_2^{\text{X}}$	
	|	$E_1^{\text{F}} = E_2^{\text{F}}$	
	|	$E_1 = E_2$	
	|	$\Delta_1 = \Delta_2$	
	|	$\delta_1 = \delta_2$	
	|	$I_1 = I_2$	

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

	|	$E, E^{\mathrm{L}} \vdash x\,\mathbf{not\,ctor}$	v is not bound to a data cons
not_shadowed	::=		
	|	$E^{\mathrm{L}} \vdash \mathit{id}\,\mathbf{not\,shadowed}$	id is not lexically shadowed
check_pat	::=		
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{pat} : t \triangleright E_2^{\mathrm{L}}$	Typing patterns, building thei
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{pat\_aux} : t \triangleright E_2^{\mathrm{L}}$	Typing patterns, building thei
id_field	::=		
	|	$E \vdash \mathit{id}\,\mathbf{field}$	Check that the identifier is a
id_value	::=		
	|	$E \vdash \mathit{id}\,\mathbf{value}$	Check that the identifier is a
check_exp	::=		
	|	$\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecting
	|	$\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp\_aux} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecting
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{qbind}_1 \mathrel{..} \mathit{qbind}_n \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for qua
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathbf{list}\ \mathit{qbind}_1 \mathrel{..} \mathit{qbind}_n \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for qua
	|	$\Delta, E, E^{\mathrm{L}} \vdash \mathit{funcl} \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a fu
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{letbind} \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a le
check_rule	::=		
	|	$\Delta, E, E^{\mathrm{L}} \vdash \mathit{rule} \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for an
check_texp_tc	::=		
	|	$\mathit{xs}, \Delta_1, E \vdash\ \mathbf{tc}\,\mathit{td} \triangleright \Delta_2, E^{\mathrm{P}}$	Extract the type constructor i
check_texps_tc	::=		
	|	$\mathit{xs}, \Delta_1, E \vdash\ \mathbf{tc}\,\mathit{td}_1 \mathrel{..} \mathit{td}_i \triangleright \Delta_2, E^{\mathrm{P}}$	Extract the type constructor i
check_texp	::=		
	|	$\Delta, E \vdash \mathit{tnvs}\ p = \mathit{texp} \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	Check a type definition, with
check_texps	::=		
	|	$\mathit{xs}, \Delta, E \vdash \mathit{td}_1 \mathrel{..} \mathit{td}_n \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	
convert_class	::=		
	|	$\delta, E \vdash \mathit{id} \rightsquigarrow p$	Lookup a type class
solve_class_constraint	::=		
	|	$I \vdash (p\ t)\,\mathbf{IN}\,\mathcal{C}$	Solve class constraint
solve_class_constraints	::=		
	|	$I \vdash \Sigma^{\mathcal{C}} \triangleright \mathcal{C}$	Solve class constraints

check_val_def	::=		
	|	$\Delta, I, E \vdash val\_def \triangleright E^{\mathrm{X}}$	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	::=		
	|	$\overline{z_j}^{\,j}, D_1, E_1 \vdash def \triangleright D_2, E_2$	Check a definition
	|	$\overline{z_j}^{\,j}, D_1, E_1 \vdash defs \triangleright D_2, E_2$	Check definitions, given module path, definitions
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*	
	|	*hex*	
	|	*bin*	
	|	*string*	

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

$$
\begin{array}{ll}
| & name^{?} \\
| & td \\
| & c \\
| & cs \\
| & c\_pre \\
| & typschm \\
| & instschm \\
| & target \\
| & \tau \\
| & \tau^{?} \\
| & lemma\_typ \\
| & lemma \\
| & val\_def \\
| & val\_spec \\
| & def\_aux \\
| & def \\
| & ;;^{?} \\
| & defs \\
| & p \\
| & \sigma \\
| & t \\
| & ne \\
| & t\_args \\
| & t\_multi \\
| & nec \\
| & names \\
| & \mathcal{C} \\
| & env\_tag \\
| & v\_desc \\
| & f\_desc \\
| & \Sigma^{\mathcal{C}} \\
| & \Sigma^{\mathcal{N}} \\
| & E \\
| & E^{\mathrm{X}} \\
| & E^{\mathrm{F}} \\
| & E^{\mathrm{M}} \\
| & E^{\mathrm{P}} \\
| & E^{\mathrm{L}} \\
| & tc\_abbrev \\
| & tc\_def \\
| & \Delta \\
| & \delta \\
| & inst \\
| & I \\
| & D \\
| & xs
\end{array}
$$

$|$ *terminals*
$|$ *formula*

$\boxed{tnvars^l \rightsquigarrow tnvs}$

$$\frac{tnvar_1^l \rightsquigarrow tnv_1 \quad .. \quad tnvar_n^l \rightsquigarrow tnv_n}{tnvar_1^l \,..\, tnvar_n^l \rightsquigarrow tnv_1 \,..\, tnv_n} \quad \text{CONVERT\_TNVARS\_NONE}$$

$\boxed{tnvar^l \rightsquigarrow tnv}$

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{c} \Delta \vdash t_1\, \mathbf{ok} \\ \Delta \vdash t_2\, \mathbf{ok} \end{array}}{\Delta \vdash t_1 \to t_2\, \mathbf{ok}} \quad \text{CHECK\_T\_FN}$$

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

$$\frac{\begin{array}{l} \Delta(p) \triangleright tnv_1 \,..\, tnv_n\; tc\_abbrev \\ \Delta, tnv_1 \vdash t_1\, \mathbf{ok} \quad .. \quad \Delta, tnv_n \vdash t_n\, \mathbf{ok} \end{array}}{\Delta \vdash p\; t_1 \,..\, t_n\, \mathbf{ok}} \quad \text{CHECK\_T\_APP}$$

$\boxed{\Delta, tnv \vdash t\, \mathbf{ok}}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\vdash N\,l \leadsto N}\quad\textsc{convert\_nexp\_var}$$

$$\frac{}{\vdash num \leadsto num}\quad\textsc{convert\_nexp\_num}$$

$$\frac{}{\vdash num * N \leadsto num * N}\quad\textsc{convert\_nexp\_mult}$$

$$\frac{\begin{array}{c}\vdash Nexp_1 \leadsto ne_1\\ \vdash Nexp_2 \leadsto ne_2\end{array}}{\vdash Nexp_1 + Nexp_2 \leadsto ne_1 + ne_2}\quad\textsc{convert\_nexp\_add}$$

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

$$\frac{\Delta, E \vdash typ_1 \leadsto t_1 \quad .. \quad \Delta, E \vdash typ_n \leadsto t_n}{\Delta, E \vdash typ_1 * \,..\, * typ_n \leadsto (t_1 * \,..\, * t_n)}\quad\textsc{convert\_typs\_all}$$

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

$$\frac{}{\vdash \mathbf{true}\,l : \_\_\mathbf{bool}}\quad\textsc{check\_lit\_true}$$

$$\frac{}{\vdash \mathbf{false}\,l : \_\_\mathbf{bool}}\quad\textsc{check\_lit\_false}$$

$$\frac{}{\vdash num\,l : \_\_\mathbf{num}}\quad\textsc{check\_lit\_num}$$

$$\frac{num = \mathbf{bitlength}\,(hex)}{\vdash hex\,l : \_\_\mathbf{vector}\,num\,\_\_\mathbf{bit}}\quad\textsc{check\_lit\_hex}$$

$$\frac{num = \mathbf{bitlength}\,(bin)}{\vdash bin\,l : \_\_\mathbf{vector}\,num\,\_\_\mathbf{bit}}\quad\textsc{check\_lit\_bin}$$

$$\frac{}{\vdash string\,l : \_\_\mathbf{string}}\quad\textsc{check\_lit\_string}$$

$$\frac{}{\vdash ()\,l : \_\_\mathbf{unit}}\quad\textsc{check\_lit\_unit}$$

$$\frac{}{\vdash \mathbf{bitzero}\,l : \_\_\mathbf{bit}}\quad\textsc{check\_lit\_bitzero}$$

$$\frac{}{\vdash \mathbf{bitone}\,l : \_\_\mathbf{bit}}\quad\textsc{check\_lit\_bitone}$$

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

$$\frac{\begin{array}{l}E(\overline{x_i^l.}^{\,i}) \rhd \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle\\ E^{\mathrm{F}}(y) \rhd \langle \mathbf{forall}\,tnv_1\,..\,tnv_n.p \rightarrow t, (z\ \mathbf{of}\ names)\rangle\\ \Delta \vdash t_1\ \mathbf{ok} \quad .. \quad \Delta \vdash t_n\ \mathbf{ok}\end{array}}{\Delta, E \vdash \mathbf{field}\,\overline{x_i^l.}^{\,i}\ y\ l_1\ l_2 : p\ t_1\,..\,t_n \rightarrow \{tnv_1 \mapsto t_1\,..\,tnv_n \mapsto t_n\}(t) \rhd (z\ \mathbf{of}\ names)}\quad\textsc{inst\_field\_all}$$

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

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

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

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

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

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

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

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

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

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

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

$\boxed{\Delta, E, E_1^{\mathrm{L}} \vdash pat : t \rhd E_2^{\mathrm{L}}}$ 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

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

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

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

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

$$\frac{\begin{array}{l} \Delta \vdash t\, \mathbf{ok} \\ E, E^{\mathrm{L}} \vdash x\, \mathbf{not\ ctor} \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash\ x\; l_1\; l_2\ : t \rhd \{x \mapsto t\}} \quad \text{CHECK\_PAT\_AUX\_VAR}$$

$$\dfrac{\begin{array}{l}\overline{\Delta, E \vdash \textbf{field}\ id_i : p\ t\_args \rightarrow t_i \rhd (x_i\ \textbf{of}\ names)}^{i}\\ \overline{\Delta, E, E^{\text{L}} \vdash pat_i : t_i \rhd E^{\text{L}}_i}^{i}\\ \textbf{disjoint doms}\ (\ \overline{E^{\text{L}}_i}^{i}\ )\\ \textbf{duplicates}\ (\ \overline{x_i}^{i}\ ) = \emptyset\end{array}}{\Delta, E, E^{\text{L}} \vdash \langle|\ \overline{id_i = pat_i\ l_i}^{i}\ ;^?|\rangle : p\ t\_args \rhd \uplus\,\overline{E^{\text{L}}_i}^{i}}\quad \text{CHECK\_PAT\_AUX\_RECORD}$$

$$\dfrac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : t \rhd E^{\text{L}}_1 \quad ... \quad \Delta, E, E^{\text{L}} \vdash pat_n : t \rhd E^{\text{L}}_n\\ \textbf{disjoint doms}\ (E^{\text{L}}_1, \ ... , E^{\text{L}}_n)\\ \textbf{length}\ (pat_1\ ...\ pat_n) = num\end{array}}{\Delta, E, E^{\text{L}} \vdash [|pat_1; \ ... ; pat_n\ |] : \_\_\textbf{vector}\ num\ t \rhd E^{\text{L}}_1 \uplus\ ...\ \uplus E^{\text{L}}_n}\quad \text{CHECK\_PAT\_AUX\_VECTOR}$$

$$\dfrac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : \_\_\textbf{vector}\ ne_1\ t \rhd E^{\text{L}}_1 \quad ... \quad \Delta, E, E^{\text{L}} \vdash pat_n : \_\_\textbf{vector}\ ne_n\ t \rhd E^{\text{L}}_n\\ \textbf{disjoint doms}\ (E^{\text{L}}_1, \ ... , E^{\text{L}}_n)\\ ne' = ne_1 +\ ...\ + ne_n\end{array}}{\Delta, E, E^{\text{L}} \vdash [|pat_1\ ...\ pat_n|] : \_\_\textbf{vector}\ ne'\ t \rhd E^{\text{L}}_1 \uplus\ ...\ \uplus E^{\text{L}}_n}\quad \text{CHECK\_PAT\_AUX\_VECTOR}$$

$$\dfrac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : t_1 \rhd E^{\text{L}}_1 \quad .... \quad \Delta, E, E^{\text{L}} \vdash pat_n : t_n \rhd E^{\text{L}}_n\\ \textbf{disjoint doms}\ (E^{\text{L}}_1, \ .... , E^{\text{L}}_n)\end{array}}{\Delta, E, E^{\text{L}} \vdash (pat_1, \ .... , pat_n) : t_1 * \ .... \ * t_n \rhd E^{\text{L}}_1 \uplus\ ....\ \uplus E^{\text{L}}_n}\quad \text{CHECK\_PAT\_AUX\_TUP}$$

$$\dfrac{\begin{array}{l}\Delta \vdash t\ \textbf{ok}\\ \Delta, E, E^{\text{L}} \vdash pat_1 : t \rhd E^{\text{L}}_1 \quad .. \quad \Delta, E, E^{\text{L}} \vdash pat_n : t \rhd E^{\text{L}}_n\\ \textbf{disjoint doms}\ (E^{\text{L}}_1, \ .. , E^{\text{L}}_n)\end{array}}{\Delta, E, E^{\text{L}} \vdash [pat_1; \ .. ; pat_n\ ;^?] : \_\_\textbf{list}\ t \rhd E^{\text{L}}_1 \uplus\ ..\ \uplus E^{\text{L}}_n}\quad \text{CHECK\_PAT\_AUX\_LIST}$$

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

$$\dfrac{\begin{array}{l}\Delta, E, E^{\text{L}}_1 \vdash pat_1 : t \rhd E^{\text{L}}_2\\ \Delta, E, E^{\text{L}}_1 \vdash pat_2 : \_\_\textbf{list}\ t \rhd E^{\text{L}}_3\\ \textbf{disjoint doms}\ (E^{\text{L}}_2, E^{\text{L}}_3)\end{array}}{\Delta, E, E^{\text{L}}_1 \vdash pat_1 :: pat_2 : \_\_\textbf{list}\ t \rhd E^{\text{L}}_2 \uplus E^{\text{L}}_3}\quad \text{CHECK\_PAT\_AUX\_CONS}$$

$$\dfrac{\vdash lit : t}{\Delta, E, E^{\text{L}} \vdash lit : t \rhd \{\,\}}\quad \text{CHECK\_PAT\_AUX\_LIT}$$

$$\dfrac{E, E^{\text{L}} \vdash x\ \textbf{not ctor}}{\Delta, E, E^{\text{L}} \vdash x\ l + num : \_\_\textbf{num} \ \rhd \{x \mapsto \_\_\textbf{num}\ \}}\quad \text{CHECK\_PAT\_AUX\_NUM\_ADD}$$

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

$$\dfrac{E^{\text{F}}(x) \rhd f\_desc}{\langle E^{\text{M}}, E^{\text{P}}, E^{\text{F}}, E^{\text{X}}\rangle \vdash\ x\ l_1\ l_2\ \textbf{field}}\quad \text{ID\_FIELD\_EMPTY}$$

$$\dfrac{\begin{array}{l}E^{\text{M}}(x) \rhd E\\ x \notin \textbf{dom}\ (E^{\text{F}})\\ E \vdash \overline{y^l_i.}^{i}\ z^l\ l_2\ \textbf{field}\end{array}}{\langle E^{\text{M}}, E^{\text{P}}, E^{\text{F}}, E^{\text{X}}\rangle \vdash x\ l_1.\ \overline{y^l_i.}^{i}\ z^l\ l_2\ \textbf{field}}\quad \text{ID\_FIELD\_CONS}$$

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

$$\dfrac{E^{\text{X}}(x) \rhd v\_desc}{\langle E^{\text{M}}, E^{\text{P}}, E^{\text{F}}, E^{\text{X}}\rangle \vdash\ x\ l_1\ l_2\ \textbf{value}}\quad \text{ID\_VALUE\_EMPTY}$$

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash \mathit{pat}_1 : t_1 \rhd E_1^{\mathrm{L}} \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash \mathit{pat}_n : t_n \rhd E_n^{\mathrm{L}} \\ \Delta, E, E^{\mathrm{L}} \uplus E_1^{\mathrm{L}} \uplus \ldots \uplus E_n^{\mathrm{L}} \vdash \mathit{exp} : u \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \mathbf{disjoint\ doms}\,(E_1^{\mathrm{L}}, \ldots, E_n^{\mathrm{L}}) \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{fun}\; \mathit{pat}_1 \ldots \mathit{pat}_n \to \mathit{exp}\; l : \mathbf{curry}\,((t_1 * \ldots * t_n), u) \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_EXP\_AUX\_FN}$$

$$\frac{\begin{array}{l} \overline{\Delta, E, E^{\mathrm{L}} \vdash \mathit{pat}_i : t \rhd E_i^{\mathrm{L}}}^{\,i} \\ \overline{\Delta, E, E^{\mathrm{L}} \uplus E_i^{\mathrm{L}} \vdash \mathit{exp}_i : u \rhd \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{\,i} \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{function}\; |^{?}\; \overline{\mathit{pat}_i \to \mathit{exp}_i\; l_i}^{\,i}\; \mathbf{end} : t \to u \rhd \overline{\Sigma^{\mathcal{C}}{}_i}^{\,i}, \overline{\Sigma^{\mathcal{N}}{}_i}^{\,i}} \quad \text{CHECK\_EXP\_AUX\_FUNCTION}$$

$$\frac{\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_1 : t_1 \to t_2 \rhd \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_2 : t_1 \rhd \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_1\; \mathit{exp}_2 : t_2 \rhd \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2} \quad \text{CHECK\_EXP\_AUX\_APP}$$

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

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

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

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

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

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

$$\dfrac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash \mathit{exp} : \_\_\mathbf{vector}\ \mathit{ne}'\ t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \vdash \mathit{Nexp}_1 \leadsto \mathit{ne}_1\\ \vdash \mathit{Nexp}_2 \leadsto \mathit{ne}_2\\ \mathit{ne} = \mathit{ne}_2 + (-\mathit{ne}_1)\end{array}}{\Delta, E, E^{\text{L}} \vdash \mathit{exp}.(\mathit{Nexp}_1..\mathit{Nexp}_2) : \_\_\mathbf{vector}\ \mathit{ne}\ t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \cup \{\mathit{ne}_1\langle \mathit{ne}_2\langle \mathit{ne}'\}} \quad \text{CHECK\_EXP\_AUX\_VECTORSUB}$$

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

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

$$\dfrac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash \mathit{exp} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \Delta, E \vdash \mathit{typ} \leadsto t\end{array}}{\Delta, E, E^{\text{L}} \vdash (\mathit{exp} : \mathit{typ}) : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_EXP\_AUX\_TYPED}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

$\boxed{\Delta, E, E^{\mathrm{L}}_1 \vdash \mathit{qbind}_1\ ..\ \mathit{qbind}_n \triangleright E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}}$ Build the environment for quantifier bindings, collecting typeclass cons

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

$$\frac{\begin{array}{l}\Delta \vdash t\ \mathbf{ok} \\ \Delta, E, E^{\mathrm{L}}_1 \uplus \{x \mapsto t\} \vdash\ \overline{\mathit{qbind}_i}^{\,i}\ \triangleright E^{\mathrm{L}}_2, {\Sigma^{\mathcal{C}}}_1 \\ \mathbf{disjoint\ doms}\,(\{x \mapsto t\}, E^{\mathrm{L}}_2)\end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash x\ l\ \overline{\mathit{qbind}_i}^{\,i}\ \triangleright \{x \mapsto t\} \uplus E^{\mathrm{L}}_2, {\Sigma^{\mathcal{C}}}_1} \quad \text{CHECK\_LISTQUANT\_BINDING\_VAR}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} \overline{\Delta \vdash t_i\ \textbf{ok}}^{\,i} \\ E^{\mathrm{L}}_2 = \{\,\overline{y_i \mapsto t_i}^{\,i}\,\} \\ \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_2 \vdash \mathit{exp}' : \_\_\textbf{bool} \triangleright \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime} \\ \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_2 \vdash \mathit{exp}_1 : u_1 \triangleright {\Sigma^{\mathcal{C}}}_1, {\Sigma^{\mathcal{N}}}_1 \quad .. \quad \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_2 \vdash \mathit{exp}_n : u_n \triangleright {\Sigma^{\mathcal{C}}}_n, {\Sigma^{\mathcal{N}}}_n \end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash \mathit{id}^{?}\ \textbf{forall}\ \overline{y_i\ l_i}^{\,i}\ .\mathit{exp}' \Longrightarrow x\ l\ \mathit{exp}_1 \,..\, \mathit{exp}_n\ l' \triangleright \{x \mapsto \textbf{curry}\,((u_1 * \,..\, * u_n), \_\_\textbf{bool}\,)\}, \Sigma^{\mathcal{C}\prime} \cup {\Sigma^{\mathcal{C}}}_1 \cup \,..\, \cup \Sigma}$$

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

$$\frac{\begin{array}{l} \mathit{tnvars}^{l} \leadsto \mathit{tnvs} \\ \Delta, E \vdash \mathit{typ} \leadsto t \\ \textbf{duplicates}\,(\mathit{tnvs}) = \emptyset \\ \textbf{FV}\,(t) \subset \mathit{tnvs} \\ \overline{y_i.}^{\,i}\ x \notin \textbf{dom}\,(\Delta) \end{array}}{\overline{y_i}^{\,i}, \Delta, E \vdash\ \textbf{tc}\ x\ l\ \mathit{tnvars}^{l}\ = \mathit{typ} \triangleright \{\overline{y_i.}^{\,i}\ x \mapsto \mathit{tnvs}\ .t\}, \{x \mapsto \overline{y_i.}^{\,i}\ x\}} \quad \text{CHECK\_TEXP\_TC\_ABBREV}$$

$$\frac{\begin{array}{l} tnvars^l \rightsquigarrow tnvs \\ \mathbf{duplicates}\,(tnvs) = \emptyset \\ \overline{y_i.}^{\,i}\, x \notin \mathbf{dom}\,(\Delta) \end{array}}{\overline{y_i}^{\,i}, \Delta, E_1 \vdash \mathbf{tc}\, x\, l\, tnvars^l \triangleright \{\overline{y_i.}^{\,i}\, x \mapsto tnvs\}, \{x \mapsto \overline{y_i.}^{\,i}\, x\}}\quad \text{CHECK\_TEXP\_TC\_ABSTRACT}$$

$$\frac{\begin{array}{l} tnvars^l \rightsquigarrow tnvs \\ \mathbf{duplicates}\,(tnvs) = \emptyset \\ \overline{y_i.}^{\,i}\, x \notin \mathbf{dom}\,(\Delta) \end{array}}{\overline{y_i}^{\,i}, \Delta_1, E \vdash \mathbf{tc}\, x\, l\, tnvars^l = \langle| x_1^l : typ_1; \ldots; x_j^l : typ_j\, ;^? |\rangle \triangleright \{\overline{y_i.}^{\,i}\, x \mapsto tnvs\}, \{x \mapsto \overline{y_i.}^{\,i}\, x\}}\quad \text{CHECK\_TEXP\_TC\_REC}$$

$$\frac{\begin{array}{l} tnvars^l \rightsquigarrow tnvs \\ \mathbf{duplicates}\,(tnvs) = \emptyset \\ \overline{y_i.}^{\,i}\, x \notin \mathbf{dom}\,(\Delta) \end{array}}{\overline{y_i}^{\,i}, \Delta_1, E \vdash \mathbf{tc}\, x\, l\, tnvars^l = |^?\, ctor\_def_1 | \ldots | ctor\_def_j \triangleright \{\overline{y_i.}^{\,i}\, x \mapsto tnvs\}, \{x \mapsto \overline{y_i.}^{\,i}\, x\}}\quad \text{CHECK\_TEXP\_TC\_VAR}$$

$\boxed{xs, \Delta_1, E \vdash \mathbf{tc}\, td_1 \mathinner{..} td_i \triangleright \Delta_2, E^{\mathrm{P}}}$ Extract the type constructor information

$$\frac{}{xs, \Delta, E \vdash \mathbf{tc} \triangleright \{\,\}, \{\,\}}\quad \text{CHECK\_TEXPS\_TC\_EMPTY}$$

$$\frac{\begin{array}{l} xs, \Delta_1, E \vdash \mathbf{tc}\, td \triangleright \Delta_2, E_2^{\mathrm{P}} \\ xs, \Delta_1 \uplus \Delta_2, E \uplus \langle \{\,\}, E_2^{\mathrm{P}}, \{\,\}, \{\,\} \rangle \vdash \mathbf{tc}\, \overline{td_i}^{\,i} \triangleright \Delta_3, E_3^{\mathrm{P}} \\ \mathbf{dom}\,(E_2^{\mathrm{P}}) \cap \mathbf{dom}\,(E_3^{\mathrm{P}}) = \emptyset \end{array}}{xs, \Delta_1, E \vdash \mathbf{tc}\, td\, \overline{td_i}^{\,i} \triangleright \Delta_2 \uplus \Delta_3, E_2^{\mathrm{P}} \uplus E_3^{\mathrm{P}}}\quad \text{CHECK\_TEXPS\_TC\_ABBREV}$$

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

$$\frac{}{\Delta, E \vdash tnvs\, p = typ \triangleright \langle \{\,\}, \{\,\} \rangle}\quad \text{CHECK\_TEXP\_ABBREV}$$

$$\frac{\begin{array}{l} \overline{\Delta, E \vdash typ_i \rightsquigarrow t_i}^{\,i} \\ names = \{\, \overline{x_i}^{\,i} \,\} \\ \mathbf{duplicates}\,(\, \overline{x_i}^{\,i}\,) = \emptyset \\ \overline{\mathbf{FV}\,(t_i) \subset tnvs}^{\,i} \\ E^{\mathrm{F}} = \{\, \overline{x_i \mapsto \langle\, \mathbf{forall}\, tnvs.p \to t_i, (x_i\, \mathbf{of}\, names) \rangle}^{\,i} \,\} \end{array}}{\Delta, E \vdash tnvs\, p = \langle|\, \overline{x_i^l : typ_i}^{\,i}\, ;^? |\rangle \triangleright \langle E^{\mathrm{F}}, \{\,\} \rangle}\quad \text{CHECK\_TEXP\_REC}$$

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

$\boxed{xs, \Delta, E \vdash td_1 \mathinner{..} td_n \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle}$

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

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

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

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

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

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

$$\frac{}{I \vdash (p\; \alpha)\, \mathbf{IN}\, (p_1\; tnv_1) \,..\, (p_i\; tnv_i)(p\; \alpha)(p'_1\; tnv'_1) \,..\, (p'_j\; tnv'_j)} \quad \text{SOLVE\_CLASS\_CONSTRAINT\_IMMEDIATE}$$

$$\frac{\begin{array}{l} (p_1\; tnv_1) \,..\, (p_n\; tnv_n) \Rightarrow (p\; t)\, \mathbf{IN}\, I \\ I \vdash (p_1\, \sigma(tnv_1))\, \mathbf{IN}\, \mathcal{C} \quad .. \quad I \vdash (p_n\, \sigma(tnv_n))\, \mathbf{IN}\, \mathcal{C} \end{array}}{I \vdash (p\, \sigma(t))\, \mathbf{IN}\, \mathcal{C}} \quad \text{SOLVE\_CLASS\_CONSTRAINT\_CHAIN}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Definition rules:        145 good    0 bad
Definition rule clauses: 437 good    0 bad
```