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
	$\|$		
$x^l,\ y^l,\ z^l,\ name$	$::=$		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
	$\|$	$x^l_1. \ldots x^l_n.x^l\ l$	
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
	$\|$	$tnv_1 \ldots tnv_n$	
$tnvars^l$	$::=$		Type variable lists
	$\|$	$tnvar^l_1 \ldots tnvar^l_n$	
$Nexp\_aux$	$::=$		Numerical expressions for specifying vector lengths and indexe
	$\|$	N	
	$\|$	num	
	$\|$	$Nexp_1 * Nexp_2$	
	$\|$	$Nexp_1 + Nexp_2$	
	$\|$	$(Nexp)$	

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

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

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
	|	$pat \rightarrow exp\ l$	
$psexp$	::=		Multi-pattern matches
	|	$pat_1 \ldots pat_n \rightarrow exp\ l$	
$tannot^?$	::=		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
	|	$id^?\ \mathbf{forall}\ x_1^l\ ..\ x_n^l.exp \Longrightarrow x^l\ exp_1\ ..\ exp_i$	
$rule$	::=		Location-annotated inductively defined relatio
	|	$rule\_aux\ l$	

$typs$	::=			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
	$\vert$		
	$\vert$	τ	
lemma_typ	$::=$		Types of Lemmata
	$\vert$	**assert**	
	$\vert$	**lem**	
	$\vert$	**thm**	
lemma	$::=$		Lemmata and Tests
	$\vert$	$\mathit{lemma\_typ}\ \tau^?\ x^l : \mathit{exp}$	
	$\vert$	$\mathit{lemma\_typ}\ \tau^?\ \mathit{exp}$	
val_def	$::=$		Value definitions
	$\vert$	**let** $\tau^?$ *letbind*	Non-recursive value definition
	$\vert$	**let rec** $\tau^?$ funcl_1 **and** ... **and** funcl_n	Recursive function definitions
	$\vert$	**let inline** $\tau^?$ *letbind*	Function definitions to be inli
val_spec	$::=$		Value type specifications
	$\vert$	**val** $x^l : \mathit{typschm}$	
def_aux	$::=$		Top-level definitions
	$\vert$	**type** td_1 **and** ... **and** td_n	Type definitions
	$\vert$	*val_def*	Value definitions
	$\vert$	*lemma*	Lemmata
	$\vert$	**rename** $\tau^?\ \mathit{id} = x^l$	Rename constant or type
	$\vert$	**module** $x^l =$ **struct** *defs* **end**	Module definitions
	$\vert$	**module** $x^l = \mathit{id}$	Module renamings
	$\vert$	**open** *id*	Opening modules
	$\vert$	**indreln** $\tau^?$ rule_1 **and** ... **and** rule_n	Inductively defined relations
	$\vert$	*val_spec*	Top-level type constraints
	$\vert$	**class** $(x^l\ \mathit{tnvar}^l)$ **val** $x_1^l : \mathit{typ}_1\ l_1$ ... **val** $x_n^l : \mathit{typ}_n\ l_n$ **end**	Typeclass definitions
	$\vert$	**instance** *instschm* $\mathit{val\_def}_1\ l_1$ ... $\mathit{val\_def}_n\ l_n$ **end**	Typeclass instantiations
def	$::=$		Location-annotated definitions
	$\vert$	$\mathit{def\_aux}\ l$	
$;;^?$	$::=$		Optional double-semi-colon
	$\vert$		
	$\vert$	$;;$	
defs	$::=$		Definition sequences
	$\vert$	$\mathit{def}_1\ ;;^?_1\ ..\ \mathit{def}_n\ ;;^?_n$	
p	$::=$		Unique paths
	$\vert$	$x_1 .\ ..\ x_n . x$	
	$\vert$	**__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		
	|	$t_1 \to t_2$		
	|	$t_1 * \ldots. * t_n$		
	|	$p\ t\_args$		
	|	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	
	|	$ne_1 + \ldots + ne_n$	M	
	|	**bitlength** (bin)	M	
	|	**bitlength** (hex)	M	
	|	**length** $(pat_1 \ldots pat_n)$	M	
	|	**length** $(exp_1 \ldots exp_n)$	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
	$\mid$ $\{x_1, \,..\,, x_n\}$		
$\mathcal{C}$	$::=$		Typeclass constraint lists
	$\mid$ $(p_1\ tnv_1)\,..\,(p_n\ tnv_n)$		
env_tag	$::=$		Tags for the (non-constructor) value descripti
	$\mid$ $\textbf{method}$		Bound to a method
	$\mid$ $\textbf{val}$		Specified with val
	$\mid$ $\textbf{let}$		Defined with let or indreln
v_desc	$::=$		Value descriptions
	$\mid$ $\langle\,\textbf{forall}\,tnvs.t\_multi \rightarrow p, (x\ \textbf{of}\ names)\rangle$		Constructors
	$\mid$ $\langle\,\textbf{forall}\,tnvs.\mathcal{C} \Rightarrow t, env\_tag\rangle$		Values
f_desc	$::=$		
	$\mid$ $\langle\,\textbf{forall}\,tnvs.p \rightarrow t, (x\ \textbf{of}\ names)\rangle$		Fields
$\Sigma^{\mathcal{C}}$	$::=$		Typeclass constraints
	$\mid$ $\{(p_1\ t_1), \,..\,, (p_n\ t_n)\}$		
	$\mid$ $\Sigma^{\mathcal{C}}{}_1 \cup \,..\, \cup \Sigma^{\mathcal{C}}{}_n$	M	
$\Sigma^{\mathcal{N}}$	$::=$		Nexp constraint lists
	$\mid$ $\{nec_1, \,..\,, nec_n\}$		
	$\mid$ $\Sigma^{\mathcal{N}}{}_1 \cup \,..\, \cup \Sigma^{\mathcal{N}}{}_n$	M	
E	$::=$		Environments
	$\mid$ $\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle$		
	$\mid$ $E_1 \uplus E_2$	M	
	$\mid$ ϵ	M	
E^{X}	$::=$		Value environments
	$\mid$ $\{x_1 \mapsto v\_desc_1, \,..\,, x_n \mapsto v\_desc_n\}$		
	$\mid$ $E_1^{\mathrm{X}} \uplus \,..\, \uplus E_n^{\mathrm{X}}$	M	
E^{F}	$::=$		Field environments
	$\mid$ $\{x_1 \mapsto f\_desc_1, \,..\,, x_n \mapsto f\_desc_n\}$		
	$\mid$ $E_1^{\mathrm{F}} \uplus \,..\, \uplus E_n^{\mathrm{F}}$	M	
E^{M}	$::=$		Module environments
	$\mid$ $\{x_1 \mapsto E_1, \,..\,, x_n \mapsto E_n\}$		
E^{P}	$::=$		Path environments
	$\mid$ $\{x_1 \mapsto p_1, \,..\,, x_n \mapsto p_n\}$		
	$\mid$ $E_1^{\mathrm{P}} \uplus \,..\, \uplus E_n^{\mathrm{P}}$	M	
E^{L}	$::=$		Lexical bindings

$\mid$ $\{x_1 \mapsto t_1, \, .. \, , x_n \mapsto t_n\}$
$\mid$ $E_1^{\mathrm{L}} \uplus \, .. \, \uplus E_n^{\mathrm{L}}$ 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 $::=$
$\mid$ $x_1 \, .. \, x_n$

$terminals$ $::=$
$\mid$ $\geq$ `>=`
$\mid$ $\rightarrow$ `->`
$\mid$ $\Longrightarrow$ `==>`
$\mid$ $\langle|$ `<|`
$\mid$ $|\rangle$ `|>`
$\mid$ $\cap$
$\mid$ $\cup$
$\mid$ $\uplus$
$\mid$ $\notin$
$\mid$ $\subset$
$\mid$ $\neq$
$\mid$ $\emptyset$
$\mid$ $\langle$

	$\mid$	$\rangle$	
	$\mid$	$\vdash$	
	$\mid$	$,$	
	$\mid$	$\mapsto$	
	$\mid$	$\triangleright$	
	$\mid$	$\leadsto$	
	$\mid$	$\Rightarrow$	
	$\mid$	$-$	
	$\mid$	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
	$\mid$	$\mathbf{dom}\,(E_1^{\mathrm{M}}) \cap \mathbf{dom}\,(E_2^{\mathrm{M}}) = \emptyset$	
	$\mid$	$\mathbf{dom}\,(E_1^{\mathrm{X}}) \cap \mathbf{dom}\,(E_2^{\mathrm{X}}) = \emptyset$	
	$\mid$	$\mathbf{dom}\,(E_1^{\mathrm{F}}) \cap \mathbf{dom}\,(E_2^{\mathrm{F}}) = \emptyset$	
	$\mid$	$\mathbf{dom}\,(E_1^{\mathrm{P}}) \cap \mathbf{dom}\,(E_2^{\mathrm{P}}) = \emptyset$	
	$\mid$	$\mathbf{disjoint\,doms}\,(E_1^{\mathrm{L}}, \ldots., E_n^{\mathrm{L}})$	Pairwise disjoint domains
	$\mid$	$\mathbf{disjoint\,doms}\,(E_1^{\mathrm{X}}, \ldots., E_n^{\mathrm{X}})$	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
	$\mid$	$inst\,\mathbf{IN}\,I$	
	$\mid$	$(p\ t) \notin I$	
	$\mid$	$E_1^{\mathrm{L}} = E_2^{\mathrm{L}}$	
	$\mid$	$E_1^{\mathrm{X}} = E_2^{\mathrm{X}}$	
	$\mid$	$E_1^{\mathrm{F}} = E_2^{\mathrm{F}}$	
	$\mid$	$E_1 = E_2$	
	$\mid$	$\Delta_1 = \Delta_2$	
	$\mid$	$\delta_1 = \delta_2$	
	$\mid$	$I_1 = I_2$	

	$\vert$	$names_1 = names_2$	
	$\vert$	$t_1 = t_2$	
	$\vert$	$\sigma_1 = \sigma_2$	
	$\vert$	$p_1 = p_2$	
	$\vert$	$xs_1 = xs_2$	
	$\vert$	$tnvs_1 = tnvs_2$	
$convert\_tnvars$	$::=$		
	$\vert$	$tnvars^l \leadsto tnvs$	
	$\vert$	$tnvar^l \leadsto tnv$	
$look\_m$	$::=$		
	$\vert$	$E_1(x_1^l \,..\, x_n^l) \triangleright E_2$	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
$convert\_typ$	$::=$		
	$\vert$	$\Delta, E \vdash typ \leadsto t$	Convert source types to in
	$\vert$	$\vdash Nexp \leadsto ne$	Convert and normalize nu
$convert\_typs$	$::=$		
	$\vert$	$\Delta, E \vdash typs \leadsto t\_multi$	
$check\_lit$	$::=$		
	$\vert$	$\vdash lit : t$	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	$::=$		

	$\vert$	$E, E^{\mathrm{L}} \vdash x\ \mathbf{not\ ctor}$	v is not bound to a data cons
$not\_shadowed$	$::=$		
	$\vert$	$E^{\mathrm{L}} \vdash id\ \mathbf{not\ shadowed}$	id is not lexically shadowed
$check\_pat$	$::=$		
	$\vert$	$\Delta, E, E_1^{\mathrm{L}} \vdash pat : t \rhd E_2^{\mathrm{L}}$	Typing patterns, building thei
	$\vert$	$\Delta, E, E_1^{\mathrm{L}} \vdash pat\_aux : t \rhd E_2^{\mathrm{L}}$	Typing patterns, building thei
id_field	$::=$		
	$\vert$	$E \vdash id\ \mathbf{field}$	Check that the identifier is a
id_value	$::=$		
	$\vert$	$E \vdash id\ \mathbf{value}$	Check that the identifier is a
$check\_exp$	$::=$		
	$\vert$	$\Delta, E, E^{\mathrm{L}} \vdash exp : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecting
	$\vert$	$\Delta, E, E^{\mathrm{L}} \vdash exp\_aux : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecting
	$\vert$	$\Delta, E, E_1^{\mathrm{L}} \vdash qbind_1 \mathrel{..} qbind_n \rhd E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for qua
	$\vert$	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathbf{list}\ qbind_1 \mathrel{..} qbind_n \rhd E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for qua
	$\vert$	$\Delta, E, E^{\mathrm{L}} \vdash funcl \rhd \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a fu
	$\vert$	$\Delta, E, E_1^{\mathrm{L}} \vdash letbind \rhd E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a le
$check\_rule$	$::=$		
	$\vert$	$\Delta, E, E^{\mathrm{L}} \vdash rule \rhd \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for an
$check\_texp\_tc$	$::=$		
	$\vert$	$xs, \Delta_1, E \vdash \mathbf{tc}\ td \rhd \Delta_2, E^{\mathrm{P}}$	Extract the type constructor i
$check\_texps\_tc$	$::=$		
	$\vert$	$xs, \Delta_1, E \vdash \mathbf{tc}\ td_1 \mathrel{..} td_i \rhd \Delta_2, E^{\mathrm{P}}$	Extract the type constructor i
$check\_texp$	$::=$		
	$\vert$	$\Delta, E \vdash tnvs\ p = texp \rhd \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	Check a type definition, with
$check\_texps$	$::=$		
	$\vert$	$xs, \Delta, E \vdash td_1 \mathrel{..} td_n \rhd \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	
$convert\_class$	$::=$		
	$\vert$	$\delta, E \vdash id \rightsquigarrow p$	Lookup a type class
$solve\_class\_constraint$	$::=$		
	$\vert$	$I \vdash (p\ t)\ \mathbf{IN}\ \mathcal{C}$	Solve class constraint
$solve\_class\_constraints$	$::=$		
	$\vert$	$I \vdash \Sigma^{\mathcal{C}} \rhd \mathcal{C}$	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, 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*	

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

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

| *terminals*
| *formula*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\vdash \mathit{num} \rightsquigarrow \mathit{num}}\quad \text{CONVERT\_NEXP\_NUM}$$

$$\frac{}{\vdash \mathit{num} * N \rightsquigarrow \mathit{num} * N}\quad \text{CONVERT\_NEXP\_MULT}$$

$$\frac{\begin{array}{l}\vdash \mathit{Nexp}_1 \rightsquigarrow \mathit{ne}_1\\ \vdash \mathit{Nexp}_2 \rightsquigarrow \mathit{ne}_2\end{array}}{\vdash \mathit{Nexp}_1 + \mathit{Nexp}_2 \rightsquigarrow \mathit{ne}_1 + \mathit{ne}_2}\quad \text{CONVERT\_NEXP\_ADD}$$

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

$$\frac{\Delta, E \vdash \mathit{typ}_1 \rightsquigarrow t_1 \quad .. \quad \Delta, E \vdash \mathit{typ}_n \rightsquigarrow t_n}{\Delta, E \vdash \mathit{typ}_1 * .. * \mathit{typ}_n \rightsquigarrow (t_1 * .. * t_n)}\quad \text{CONVERT\_TYPS\_ALL}$$

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

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

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

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

$$\frac{\mathit{num} = \mathbf{bitlength}\,(\mathit{hex})}{\vdash \mathit{hex}\, l : \mathbf{\_\_vector}\, \mathit{num}\, \mathbf{\_\_bit}}\quad \text{CHECK\_LIT\_HEX}$$

$$\frac{\mathit{num} = \mathbf{bitlength}\,(\mathit{bin})}{\vdash \mathit{bin}\, l : \mathbf{\_\_vector}\, \mathit{num}\, \mathbf{\_\_bit}}\quad \text{CHECK\_LIT\_BIN}$$

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

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

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

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

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

$$\frac{\begin{array}{l}E(\,\overline{x_i^l.}^{\,i}\,) \rhd \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle\\ E^{\mathrm{F}}(y) \rhd \langle \mathbf{forall}\, \mathit{tnv}_1 .. \mathit{tnv}_n.p \to t, (z\, \mathbf{of}\, \mathit{names})\rangle\\ \Delta \vdash t_1\, \mathbf{ok} \quad .. \quad \Delta \vdash t_n\, \mathbf{ok}\end{array}}{\Delta, E \vdash \mathbf{field}\, \overline{x_i^l.}^{\,i}\; y\, l_1\, l_2 : p\; t_1 .. t_n \to \{\mathit{tnv}_1 \mapsto t_1 .. \mathit{tnv}_n \mapsto t_n\}(t) \rhd (z\, \mathbf{of}\, \mathit{names})}\quad \text{INST\_FIELD\_ALL}$$

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

$$\frac{\begin{array}{l}E(\,\overline{x_i^l.}^{\,i}\,) \rhd \langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle\\ E^{\mathrm{X}}(y) \rhd \langle \mathbf{forall}\, \mathit{tnv}_1 .. \mathit{tnv}_n.\mathit{t\_multi} \to p, (z\, \mathbf{of}\, \mathit{names})\rangle\\ \Delta \vdash t_1\, \mathbf{ok} \quad .. \quad \Delta \vdash t_n\, \mathbf{ok}\end{array}}{\Delta, E \vdash \mathbf{ctor}\, \overline{x_i^l.}^{\,i}\; y\, l_1\, l_2 : \{\mathit{tnv}_1 \mapsto t_1 .. \mathit{tnv}_n \mapsto t_n\}(\mathit{t\_multi}) \to p\; t_1 .. t_n \rhd (z\, \mathbf{of}\, \mathit{names})}\quad \text{INST\_CTOR\_ALL}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\boxed{E \vdash id\ \textbf{field}}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l}\overline{\Delta, E \vdash \textbf{field}\ id_i : p\ t\_args \to t_i \rhd (x_i\ \textbf{of}\ names)}^{i}\\ \overline{\Delta, E, E^{\mathrm{L}} \vdash exp_i : t_i \rhd \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{i}\\ \textbf{duplicates}\,(\,\overline{x_i}^{i}\,) = \emptyset\\ \Delta, E, E^{\mathrm{L}} \vdash exp : p\ t\_args \rhd \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \langle|\, exp\ \textbf{with}\ \overline{id_i = exp_i\ l_i}^{i}\ ;^{?}\ l\,|\rangle : p\ t\_args \rhd \Sigma^{\mathcal{C}\prime} \cup \overline{\Sigma^{\mathcal{C}}{}_i}^{i}, \Sigma^{\mathcal{N}\prime} \cup \overline{\Sigma^{\mathcal{N}}{}_i}^{i}}\ \textsc{check\_exp\_aux\_recup}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp_1 : t \rhd \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash exp_n : t \rhd \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n\\ \textbf{length}\,(exp_1 \ldots exp_n) = num\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash [|exp_1; \ldots; exp_n|] : \_\_\textbf{vector}\ num\ t \rhd \Sigma^{\mathcal{C}}{}_1 \cup \ldots \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup \ldots \cup \Sigma^{\mathcal{N}}{}_n}\ \textsc{check\_exp\_aux\_vector}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp : \_\_\textbf{vector}\ ne'\ t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \vdash Nexp \rightsquigarrow ne\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp.(Nexp) : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \cup \{ne\langle ne'\}}\ \textsc{check\_exp\_aux\_vectorget}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp : \_\_\textbf{vector}\ ne'\ t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \vdash Nexp_1 \rightsquigarrow ne_1\\ \vdash Nexp_2 \rightsquigarrow ne_2\\ ne = ne_2 + (-ne_1)\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp.(Nexp_1..Nexp_2) : \_\_\textbf{vector}\ ne\ t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \cup \{ne_1\langle ne_2\langle ne'\}}\ \textsc{check\_exp\_aux\_vectorsub}$$

$$\frac{\begin{array}{l}E \vdash id\ \textbf{field}\\ \Delta, E \vdash \textbf{field}\ id : p\ t\_args \to t \rhd (x\ \textbf{of}\ names)\\ \Delta, E, E^{\mathrm{L}} \vdash exp : p\ t\_args \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash exp.id : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}\ \textsc{check\_exp\_aux\_field}$$

$$\frac{\begin{array}{l}\overline{\Delta, E, E^{\mathrm{L}} \vdash pat_i : t \rhd E^{\mathrm{L}}_i}^{i}\\ \overline{\Delta, E, E^{\mathrm{L}} \uplus E^{\mathrm{L}}_i \vdash exp_i : u \rhd \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{i}\\ \Delta, E, E^{\mathrm{L}} \vdash exp : t \rhd \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \textbf{match}\ exp\ \textbf{with}\ |^{?}\ \overline{pat_i \to exp_i\ l_i}^{i}\ l\ \textbf{end} : u \rhd \Sigma^{\mathcal{C}\prime} \cup \overline{\Sigma^{\mathcal{C}}{}_i}^{i}, \Sigma^{\mathcal{N}\prime} \cup \overline{\Sigma^{\mathcal{N}}{}_i}^{i}}\ \textsc{check\_exp\_aux\_case}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\\ \Delta, E \vdash typ \rightsquigarrow t\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash (exp : typ) : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}\ \textsc{check\_exp\_aux\_typed}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}}_1 \vdash letbind \rhd E^{\mathrm{L}}_2, \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1\\ \Delta, E, E^{\mathrm{L}}_1 \uplus E^{\mathrm{L}}_2 \vdash exp : t \rhd \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \textbf{let}\ letbind\ \textbf{in}\ exp : t \rhd \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2}\ \textsc{check\_exp\_aux\_let}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash exp_1 : t_1 \rhd \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad \ldots. \quad \Delta, E, E^{\mathrm{L}} \vdash exp_n : t_n \rhd \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n}{\Delta, E, E^{\mathrm{L}} \vdash (exp_1, \ldots., exp_n) : t_1 * \ldots. * t_n \rhd \Sigma^{\mathcal{C}}{}_1 \cup \ldots. \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup \ldots. \cup \Sigma^{\mathcal{N}}{}_n}\ \textsc{check\_exp\_aux\_tup}$$

$$\frac{\begin{array}{l}\Delta \vdash t\ \textbf{ok}\\ \Delta, E, E^{\mathrm{L}} \vdash exp_1 : t \rhd \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad .. \quad \Delta, E, E^{\mathrm{L}} \vdash exp_n : t \rhd \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash [exp_1; ..; exp_n\ ;^{?}] : \_\_\textbf{list}\ t \rhd \Sigma^{\mathcal{C}}{}_1 \cup .. \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup .. \cup \Sigma^{\mathcal{N}}{}_n}\ \textsc{check\_exp\_aux\_list}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash exp : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}} \vdash (exp) : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}\ \textsc{check\_exp\_aux\_paren}$$

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash exp : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}} \vdash \textbf{begin}\ exp\ \textbf{end} : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}\ \textsc{check\_exp\_aux\_begin}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash exp_1 : \_\_\textbf{bool}\ \rhd \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1\\ \Delta, E, E^{\mathrm{L}} \vdash exp_2 : t \rhd \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2\\ \Delta, E, E^{\mathrm{L}} \vdash exp_3 : t \rhd \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_3\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \textbf{if}\ exp_1\ \textbf{then}\ exp_2\ \textbf{else}\ exp_3 : t \rhd \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2 \cup \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2 \cup \Sigma^{\mathcal{N}}{}_3}\ \textsc{check\_exp\_aux\_if}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l}\Delta, E, E_1^{\text{L}} \vdash pat : t \triangleright E_3^{\text{L}} \\ \Delta, E, E_1^{\text{L}} \vdash exp : \_\_\textbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E_1^{\text{L}} \uplus E_3^{\text{L}} \vdash \overline{qbind_i}^{\,i} \triangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}{}_2 \\ \textbf{disjoint doms}\,(E_3^{\text{L}}, E_2^{\text{L}})\end{array}}{\Delta, E, E_1^{\text{L}} \vdash \textbf{list}\,(pat\,\textbf{MEM}\,exp)\,\overline{qbind_i}^{\,i} \triangleright E_2^{\text{L}} \uplus E_3^{\text{L}}, \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2} \quad \text{CHECK\_QUANT\_BINDING\_RESTR}$$

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

$$\frac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : t_1 \triangleright E_1^{\text{L}} \quad \dots \quad \Delta, E, E^{\text{L}} \vdash pat_n : t_n \triangleright E_n^{\text{L}} \\ \Delta, E, E^{\text{L}} \uplus E_1^{\text{L}} \uplus \dots \uplus E_n^{\text{L}} \vdash exp : u \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \textbf{disjoint doms}\,(E_1^{\text{L}}, \dots, E_n^{\text{L}}) \\ \Delta, E \vdash typ \rightsquigarrow u\end{array}}{\Delta, E, E^{\text{L}} \vdash x\ l_1\ pat_1 \dots pat_n\ : typ = exp\ l_2 \triangleright \{x \mapsto \textbf{curry}\,((t_1 * \dots * t_n), u)\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_FUNCL\_ANNOT}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : t_1 \triangleright E_1^{\text{L}} \quad \dots \quad \Delta, E, E^{\text{L}} \vdash pat_n : t_n \triangleright E_n^{\text{L}} \\ \Delta, E, E^{\text{L}} \uplus E_1^{\text{L}} \uplus \dots \uplus E_n^{\text{L}} \vdash exp : u \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \textbf{disjoint doms}\,(E_1^{\text{L}}, \dots, E_n^{\text{L}})\end{array}}{\Delta, E, E^{\text{L}} \vdash x\ l_1\ pat_1 \dots pat_n\ = exp\ l_2 \triangleright \{x \mapsto \textbf{curry}\,((t_1 * \dots * t_n), u)\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_FUNCL\_NOANNOT}$$

$\boxed{\Delta, E, E_1^{\text{L}} \vdash letbind \triangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ Build the environment for a let binding, collecting typeclass and index con

$$\frac{\begin{array}{l}\Delta, E, E_1^{\text{L}} \vdash pat : t \triangleright E_2^{\text{L}} \\ \Delta, E, E_1^{\text{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \Delta, E \vdash typ \rightsquigarrow t\end{array}}{\Delta, E, E_1^{\text{L}} \vdash pat\ : typ = exp\ l \triangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_LETBIND\_VAL\_ANNOT}$$

$$\frac{\begin{array}{l}\Delta, E, E_1^{\text{L}} \vdash pat : t \triangleright E_2^{\text{L}} \\ \Delta, E, E_1^{\text{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\end{array}}{\Delta, E, E_1^{\text{L}} \vdash pat\ = exp\ l \triangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_LETBIND\_VAL\_NOANNOT}$$

$$\frac{\Delta, E, E_1^{\text{L}} \vdash funcl\_aux\ l \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E_1^{\text{L}} \vdash funcl\_aux\ l \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_LETBIND\_FN}$$

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

$$\frac{\begin{array}{l}\overline{\Delta \vdash t_i\ \textbf{ok}}^{\,i} \\ E_2^{\text{L}} = \{\,\overline{y_i \mapsto t_i}^{\,i}\,\} \\ \Delta, E, E_1^{\text{L}} \uplus E_2^{\text{L}} \vdash exp' : \_\_\textbf{bool} \triangleright \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime} \\ \Delta, E, E_1^{\text{L}} \uplus E_2^{\text{L}} \vdash exp_1 : u_1 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad .. \quad \Delta, E, E_1^{\text{L}} \uplus E_2^{\text{L}} \vdash exp_n : u_n \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n\end{array}}{\Delta, E, E_1^{\text{L}} \vdash id^?\ \textbf{forall}\ \overline{y_i\ l_i}^{\,i}\ .exp' \Longrightarrow x\ l\ exp_1 \,..\, exp_n\ l' \triangleright \{x \mapsto \textbf{curry}\,((u_1 * \,..\, * u_n), \_\_\textbf{bool}\,)\}, \Sigma^{\mathcal{C}\prime} \cup \Sigma^{\mathcal{C}}{}_1 \cup \,..\, \cup \Sigma}$$

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

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

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

$$\frac{\begin{array}{l} \mathit{tnvars}^l \leadsto \mathit{tnvs} \\ \mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset \\ \overline{y_i.}^{\,i}\, x \notin \mathbf{dom}\,(\Delta) \end{array}}{\overline{y_i}^{\,i}, \Delta_1, E \vdash \mathbf{tc}\, x\, l\, \mathit{tnvars}^l = \langle| x_1^l : \mathit{typ}_1; \ldots; x_j^l : \mathit{typ}_j\, ;^? |\rangle \triangleright \{\overline{y_i.}^{\,i}\, x \mapsto \mathit{tnvs}\}, \{x \mapsto \overline{y_i.}^{\,i}\, x\}} \quad \text{CHECK\_TEXP\_TC\_REC}$$

$$\frac{\begin{array}{l} \mathit{tnvars}^l \leadsto \mathit{tnvs} \\ \mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset \\ \overline{y_i.}^{\,i}\, x \notin \mathbf{dom}\,(\Delta) \end{array}}{\overline{y_i}^{\,i}, \Delta_1, E \vdash \mathbf{tc}\, x\, l\, \mathit{tnvars}^l = |^?\, \mathit{ctor\_def}_1 | \ldots | \mathit{ctor\_def}_j \triangleright \{\overline{y_i.}^{\,i}\, x \mapsto \mathit{tnvs}\}, \{x \mapsto \overline{y_i.}^{\,i}\, x\}} \quad \text{CHECK\_TEXP\_TC\_VAR}$$

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

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

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

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

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

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

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

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

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

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

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

$\boxed{\delta, E \vdash id \rightsquigarrow p}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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