n, i, j, k	Index variables for meta-lists
num	Numeric literals
nat	Internal literal numbers
hex	Bit vector literal, specified by C-style hex number
bin	Bit vector literal, specified by C-style binary number
$string$	String literals
$backtick\_string$	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
	|	$name\_t \to x^l$	M	Extract x from a name_t
ix^l	::=			Location-annotated infix names
	|	$ix\ l$		
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 lo
	|	α		
	|	N		
$tnvar^l$	::=			Union of type variables and Nexp type variables, with locati
	|	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 inde
	|	N		
	|	num		
	|	$Nexp_1 * Nexp_2$		
	|	$Nexp_1 + Nexp_2$		
	|	$(Nexp)$		

Nexp ::= Location-annotated vector lengths
| *Nexp_aux l*

Nexp_constraint_aux ::= Whether a vector is bounded or fixed size
| $Nexp = Nexp'$
| $Nexp \geq Nexp'$

Nexp_constraint ::= Location-annotated Nexp range
| *Nexp_constraint_aux l*

typ_aux ::= Types
| _ Unspecified type
| α^l Type variables
| $typ_1 \rightarrow typ_2$ Function types
| $typ_1 * .... * typ_n$ Tuple types
| *Nexp* As a typ to permit applications over Nexps, o
| $id\ typ_1 .. typ_n$ Type applications
| $backtick\_string\ typ_1 .. typ_n$ Backend-Type applications
| (typ)

typ ::= Location-annotated types
| *typ_aux l*

lit_aux ::= Literal constants
| **true**
| **false**
| *string*
| *hex* hex and bin are constant bit vectors, entered a
| *bin*
| *string*
| *string*
| ()
| **bitzero** bitzero and bitone are constant bits, if commo
| **bitone**

lit ::=
| *lit_aux l* Location-annotated literal constants

$;^?$ ::= Optional semi-colons
|
| ;

pat_aux ::= Patterns
| _ Wildcards
| (*pat* **as** x^l) Named patterns
| (*pat* : *typ*) Typed patterns
| $id\ pat_1 .. pat_n$ Single variable and constructor patterns

	|	$\langle\| fpat_1 ; \dots ; fpat_n \;;^? \|\rangle$	Record patterns
	|	$[\| pat_1 ; .. ; pat_n \;;^? \|]$	Vector patterns
	|	$[\| pat_1 \,..\, pat_n \|]$	Concatenated vector patterns
	|	$(pat_1 , \dots. , pat_n)$	Tuple patterns
	|	$[pat_1 ; .. ; pat_n \;;^?]$	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
	|	$backtick\_string$	identifier that should be literally used in outp
	|	N	Nexp var, has type num
	|	**fun** $psexp$	Curried functions
	|	**function** $\|^?\ pexp_1 \| \dots \| pexp_n$ **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** $\|^?\ pexp_1 \| \dots \| pexp_n\ l$ **end**	Pattern matching expressions
	|	$(exp : typ)$	Type-annotated expressions
	|	**let** $letbind$ **in** exp	Let expressions
	|	$(exp_1 , \dots. , exp_n)$	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
	|	$[exp_1 \| \textbf{forall}\ qbind_1 .. qbind_n \| exp_2]$	List comprehensions (all binders mu
	|	$\textbf{do}\ id\ pat_1 \leftarrow exp_1 ;\ .. pat_n \leftarrow exp_n ;\ \textbf{in}\ exp\ \textbf{end}$	Do notation for monads
exp	::=		Location-annotated expressions
	|	$exp\_aux\ l$	
q	::=		Quantifiers
	|	**forall**	
	|	**exists**	
qbind	::=		Bindings for quantifiers
	|	x^l	
	|	$(pat\ \textbf{IN}\ exp)$	Restricted quantifications over sets
	|	$(pat\ \textbf{MEM}\ exp)$	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
	|	$funcl\_aux\ l$	
name_t	::=		Name or name with type for inductive
	|	x^l	

	|	$(x^l : \mathit{typ})$	
name_ts	::=		Names with optional typ
	|	$\mathit{name\_t}_0 \mathrel{..} \mathit{name\_t}_n$	
rule_aux	::=		Inductively defined relat
	|	$x^l :$ **forall** $\mathit{name\_t}_1 \mathrel{..} \mathit{name\_t}_i . \mathit{exp} \Longrightarrow x_1^l\ \mathit{exp}_1 \mathrel{..} \mathit{exp}_n$	
rule	::=		Location-annotated indu
	|	$\mathit{rule\_aux}\ l$	
$\mathit{witness}^?$	::=		Optional witness type na
	|		
	|	**witness type** x^l;	
$\mathit{check}^?$	::=		Option check name decla
	|		
	|	**check** x^l;	
$\mathit{functions}^?$	::=		Optional names and typ
	|		
	|	$x^l : \mathit{typ}$	
	|	$x^l : \mathit{typ}; \mathit{functions}^?$	
indreln_name_aux	::=		Name for inductively de
	|	$[x^l : \mathit{typschm}\ \mathit{witness}^?\ \mathit{check}^?\ \mathit{functions}^?]$	
indreln_name	::=		Location-annotated nam
	|	$\mathit{indreln\_name\_aux}\ l$	
typs	::=		Type lists
	|	$\mathit{typ}_1 * \ldots * \mathit{typ}_n$	
ctor_def	::=		Datatype definition clau
	|	x^l **of** *typs*	
	|	x^l	S Constant constructors
texp	::=		Type definition bodies
	|	*typ*	Type abbreviations
	|	$\langle\| x_1^l : \mathit{typ}_1; \ldots ; x_n^l : \mathit{typ}_n ;^? \|\rangle$	Record types
	|	$\|^?\ \mathit{ctor\_def}_1 \| \ldots \| \mathit{ctor\_def}_n$	Variant types
$\mathit{name}^?$	::=		Optional name specificat
	|		
	|	$[\mathit{name} = \mathit{regexp}]$	
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 and length constraint
	|		
	|	$c_1, .., c_i \Rightarrow$	Must have > 0 constraints
	|	$Nexp\_constraint_1, .., Nexp\_constraint_i \Rightarrow$	Must have > 0 constraints
	|	$c_1, .., c_i; Nexp\_constraint_1, .., Nexp\_constraint_n \Rightarrow$	Must have > 0 of both form o
c_pre	::=		Type and instance scheme prefix
	|		
	|	**forall** $tnvar_1^l \, .. \, tnvar_n^l . cs$	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**	
	|	**lem**	
$open\_import$	::=		Open or import statements
	|	**open**	
	|	**import**	
	|	**open import**	
	|	**include**	
	|	**include import**	
τ	::=		Backend target name lists
	|	$\{target_1; \, ..; target_n\}$	
	|	$\{target_1; \, ..; target_n\}$	all targets except the listed or
	|	non_exec	all non-executable targets, use
$\tau^?$	::=		Optional targets
	|		
	|	τ	

lemma_typ	::=		Types of Lemmata
	|	**assert**	
	|	**lemma**	
	|	**theorem**	
lemma_decl	::=		Lemmata and Tests
	|	$lemma\_typ\ \tau^{?}\ x^{l} : exp$	
dexp	::=		declaration field-expressions
	|	$name\_s = string\ l$	
	|	**format** $= string\ l$	
	|	**arguments** $= exp_1 \ldots exp_n\ l$	
	|	**targuments** $= texp_1 \ldots texp_n\ l$	
declare_arg	::=		arguments to a declaration
	|	*string*	
	|	$\langle\| dexp_1 ; \ldots ; dexp_n\ ;^{?}\ l \|\rangle$	
component	::=		components
	|	**module**	
	|	**function**	
	|	**type**	
	|	**field**	
termination_setting	::=		termination settings
	|	**automatic**	
	|	**manual**	
exhaustivity_setting	::=		exhaustivity settings
	|	**exhaustive**	
	|	**inexhaustive**	
elim_opt	::=		optional terms used as eliminators for patter
	|		
	|	*id*	
fixity_decl	::=		fixity declarations for infix identifiers
	|	$right\_assoc\ nat$	
	|	$left\_assoc\ nat$	
	|	$non\_assoc\ nat$	
	|		
target_rep_rhs	::=		right hand side of a target representation de
	|	**infix** *fixity_decl backtick_string*	
	|	*exp*	
	|	*typ*	
	|	**special** $string\ exp_1 \ldots exp_n$	

|

target_rep_lhs ::=
| $\mathit{target\_repcomponent}\ \mathit{id}\ x_1^l \,..\, x_n^l$
| $\mathit{target\_repcomponent}\ \mathit{id}\ \mathit{tnvars}^l$

declare_def ::=
| **declare** $\tau^?\mathit{compile\_messageid} = \mathit{string}$
| **declare** $\tau^?$ **rename module** $= x^l$
| **declare** $\tau^?$ **rename** $\mathit{component}\ \mathit{id} = x^l$
| **declare** $\tau^?\mathit{ascii\_repcomponent}\ \mathit{id} = \mathit{backtick\_string}$
| **declare** $\mathit{targettarget\_reptarget\_rep\_lhs} = \mathit{target\_rep\_rhs}$
| **declare** $\mathit{set\_flag} x_1^l = x_2^l$
| **declare** $\tau^?\mathit{termination\_argumentid} = \mathit{termination\_setting}$
| **declare** $\tau^?\mathit{pattern\_matchexhaustivity\_setting}\ \mathit{id}\ \mathit{tnvars}^l = [\mathit{id}_1;\ \ldots;\mathit{id}_n\,;^?]\mathit{elim\_opt}$

val_def ::=
| **let** $\tau^?\ \mathit{letbind}$
| **let rec** $\tau^?\ \mathit{funcl}_1$ **and** ... **and** funcl_n
| **let inline** $\tau^?\ \mathit{letbind}$
| **let** $\mathit{lem\_transform}\tau^?\ \mathit{letbind}$

ascii_opt ::=
|
| $[\mathit{backtick\_string}]$

instance_decl ::=
| **instance**
| $\mathit{default\_instance}$

class_decl ::=
| **class**
| **class inline**

val_spec ::=
| **val** $x^l\ \mathit{ascii\_opt} : \mathit{typschm}$

def_aux ::=
| **type** td_1 **and** ... **and** td_n
| *val_def*
| *lemma_decl*
| *declare_def*
| **module** $x^l =$ **struct** *defs* **end**
| **module** $x^l = \mathit{id}$
| $\mathit{open\_import}\ \mathit{id}_1 \ldots \mathit{id}_n$
| $\mathit{open\_import}\ \tau^?\ \mathit{backtick\_string}_1 \ldots \mathit{backtick\_string}_n$
| **indreln** $\tau^?\ \mathit{indreln\_name}_1$ **and** ... **and** $\mathit{indreln\_name}_i\ \mathit{rule}_1$ **and** ... **and** rule_n

	|	$\mathit{val\_spec}$		Top-
	|	$\mathit{class\_decl}(x^l\ \mathit{tnvar}^l)$ **val** $\tau_1^?\ x_1^l\ \mathit{ascii\_opt}_1 : \mathit{typ}_1\ l_1 \ldots$ **val** $\tau_n^?\ x_n^l\ \mathit{ascii\_opt}_n : \mathit{typ}_n\ l_n$ **end**		Type
	|	$\mathit{instance\_decl}\ \mathit{instschm}\ \mathit{val\_def}_1\ l_1 \ldots \mathit{val\_def}_n\ l_n$ **end**		Type
def	::=			Locatio
	|	$\mathit{def\_aux}\ l$		
$;;^?$	::=			Option
	|			
	|	$;;$		
defs	::=			Definiti
	|	$\mathit{def}_1\ ;;_1^?\ ..\ \mathit{def}_n\ ;;_n^?$		
p	::=			Unique
	|	$x_1. \ldots x_n.x$		
	|	**__list**		
	|	**__bool**		
	|	**__num**		
	|	**__set**		
	|	**__string**		
	|	**__unit**		
	|	**__bit**		
	|	**__vector**		
σ	::=			Type v
	|	$\{\mathit{tnv}_1 \mapsto t_1\ ..\ \mathit{tnv}_n \mapsto t_n\}$		
$t,\ u$	::=			Interna
	|	α		
	|	$t_1 \rightarrow t_2$		
	|	$t_1 * \ldots. * t_n$		
	|	$p\ \mathit{t\_args}$		
	|	ne		
	|	$\sigma(t)$	M	Mult
	|	$\sigma(\mathit{tnv})$	M	Singl
	|	**curry** $(\mathit{t\_multi}, t)$	M	Curr
ne	::=			interna
	|	N		
	|	nat		
	|	$\mathit{ne}_1 * \mathit{ne}_2$		
	|	$\mathit{ne}_1 + \mathit{ne}_2$		
	|	$(-\mathit{ne})$		
	|	**normalize** (ne)	M	
	|	$\mathit{ne}_1 + \ldots + \mathit{ne}_n$	M	
	|	**bitlength** (bin)	M	

	|	**bitlength** (hex)	M	
	|	**length** $(pat_1 \,...\, pat_n)$	M	
	|	**length** $(exp_1 \,...\, exp_n)$	M	
t_args	::=			Lists of types
	|	$t_1 \,..\, t_n$		
	|	$\sigma(t\_args)$	M	Multiple substitutions
t_multi	::=			Lists of types
	|	$(t_1 * \,..\, * t_n)$		
	|	$\sigma(t\_multi)$	M	Multiple substitutions
nec	::=			Numeric expression constraints
	|	$ne \langle nec$		
	|	$ne = nec$		
	|	$ne <= nec$		
	|	ne		
$names$	::=			Sets of names
	|	$\{x_1, \,..\,, x_n\}$		
$\mathcal{C}$	::=			Typeclass constraint lists
	|	$(p_1\ tnv_1) \,..\, (p_n\ tnv_n)$		
env_tag	::=			Tags for the (non-constructor) value descripti
	|	**method**		Bound to a method
	|	**val**		Specified with val
	|	**let**		Defined with let or indreln
v_desc	::=			Value descriptions
	|	$\langle\, \textbf{forall}\, tnvs.t\_multi \rightarrow p, (x\ \textbf{of}\ names)\rangle$		Constructors
	|	$\langle\, \textbf{forall}\, tnvs.\mathcal{C} \Rightarrow t, env\_tag\rangle$		Values
f_desc	::=			
	|	$\langle\, \textbf{forall}\, tnvs.p \rightarrow t, (x\ \textbf{of}\ names)\rangle$		Fields
xs	::=			
	|	$x_1 \,..\, x_n$		
$\Sigma^{\mathcal{C}}$	::=			Typeclass constraints
	|	$\{(p_1\ t_1), \,..\,, (p_n\ t_n)\}$		
	|	$\Sigma^{\mathcal{C}}{}_1 \cup \,..\, \cup \Sigma^{\mathcal{C}}{}_n$	M	
$\Sigma^{\mathcal{N}}$	::=			Nexp constraint lists
	|	$\{nec_1, \,..\,, nec_n\}$		
	|	$\Sigma^{\mathcal{N}}{}_1 \cup \,..\, \cup \Sigma^{\mathcal{N}}{}_n$	M	

E	::=			Environments
	$\|$	$\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$		
	$\|$	$E_1 \uplus E_2$	M	
	$\|$	ϵ	M	
E^{X}	::=			Value environments
	$\|$	$\{x_1 \mapsto v\_desc_1, \,..\, , x_n \mapsto v\_desc_n\}$		
	$\|$	$E^{\mathrm{X}}_1 \uplus \,..\, \uplus E^{\mathrm{X}}_n$	M	
E^{F}	::=			Field environments
	$\|$	$\{x_1 \mapsto f\_desc_1, \,..\, , x_n \mapsto f\_desc_n\}$		
	$\|$	$E^{\mathrm{F}}_1 \uplus \,..\, \uplus E^{\mathrm{F}}_n$	M	
E^{M}	::=			Module environments
	$\|$	$\{x_1 \mapsto E_1, \,..\, , x_n \mapsto E_n\}$		
E^{P}	::=			Path environments
	$\|$	$\{x_1 \mapsto p_1, \,..\, , x_n \mapsto p_n\}$		
	$\|$	$E^{\mathrm{P}}_1 \uplus \,..\, \uplus E^{\mathrm{P}}_n$	M	
E^{L}	::=			Lexical bindings
	$\|$	$\{x_1 \mapsto t_1, \,..\, , x_n \mapsto t_n\}$		
	$\|$	$\{x^l_1 \mapsto t_1, \,..\, , x^l_n \mapsto t_n\}$		
	$\|$	$E^{\mathrm{L}}_1 \uplus \,..\, \uplus E^{\mathrm{L}}_n$	M	
tc_abbrev	::=			Type abbreviations
	$\|$	$.t$		
	$\|$			
tc_def	::=			Type and class constructor definitions
	$\|$	$tnvs\ tc\_abbrev$		Type constructors
Δ	::=			Type constructor definitions
	$\|$	$\{p_1 \mapsto tc\_def_1, \,..\, , p_n \mapsto tc\_def_n\}$		
	$\|$	$\Delta_1 \uplus \Delta_2$	M	
δ	::=			Typeclass definitions
	$\|$	$\{p_1 \mapsto xs_1, \,..\, , p_n \mapsto xs_n\}$		
	$\|$	$\delta_1 \uplus \delta_2$	M	
$inst$	::=			A typeclass instance, t must not contain nested ty
	$\|$	$\mathcal{C} \Rightarrow (p\ t)$		
I	::=			Global instances
	$\|$	$\{inst_1, \,..\, , inst_n\}$		
	$\|$	$I_1 \cup I_2$	M	

D	::=			Global type definition store
	|	$\langle \Delta, \delta, I \rangle$		
	|	$D_1 \uplus D_2$	M	
	|	ϵ	M	
terminals	::=			
	|	$\geq$		`>=`
	|	$\rightarrow$		`->`
	|	$\leftarrow$		`<-`
	|	$\Longrightarrow$		`==>`
	|	$\langle\vert$		`<|`
	|	$\vert\rangle$		`|>`
	|	$\cap$		
	|	$\cup$		
	|	$\uplus$		
	|	$\notin$		
	|	$\subset$		
	|	$\neq$		
	|	$\emptyset$		
	|	$\langle$		
	|	$\rangle$		
	|	$\vdash$		
	|	$'$		
	|	$\mapsto$		
	|	$\triangleright$		
	|	$\leadsto$		
	|	$\Rightarrow$		
	|	$-$		
	|	ϵ		
formula	::=			
	|	*judgement*		
	|	$formula_1 \quad .. \quad formula_n$		
	|	$E^{\mathrm{M}}(x) \triangleright E$		Module lookup
	|	$E^{\mathrm{P}}(x) \triangleright p$		Path lookup
	|	$E^{\mathrm{F}}(x) \triangleright f\_desc$		Field lookup
	|	$E^{\mathrm{X}}(x) \triangleright v\_desc$		Value lookup
	|	$E^{\mathrm{L}}(x) \triangleright t$		Lexical binding lookup
	|	$\Delta(p) \triangleright tc\_def$		Type constructor lookup
	|	$\delta(p) \triangleright xs$		Type constructor lookup
	|	$\mathbf{dom}\,(E_1^{\mathrm{M}}) \cap \mathbf{dom}\,(E_2^{\mathrm{M}}) = \emptyset$		
	|	$\mathbf{dom}\,(E_1^{\mathrm{X}}) \cap \mathbf{dom}\,(E_2^{\mathrm{X}}) = \emptyset$		
	|	$\mathbf{dom}\,(E_1^{\mathrm{F}}) \cap \mathbf{dom}\,(E_2^{\mathrm{F}}) = \emptyset$		
	|	$\mathbf{dom}\,(E_1^{\mathrm{P}}) \cap \mathbf{dom}\,(E_2^{\mathrm{P}}) = \emptyset$		
	|	$\mathbf{disjoint\,doms}\,(E_1^{\mathrm{L}}, \,....\,, E_n^{\mathrm{L}})$		Pairwise disjoint domains
	|	$\mathbf{disjoint\,doms}\,(E_1^{\mathrm{X}}, \,....\,, E_n^{\mathrm{X}})$		Pairwise disjoint domains

	|	$\mathbf{compatible\,overlap}\,(x_1 \mapsto t_1, \,..\, , x_n \mapsto t_n)$	$(x_i = x_j) \Longrightarrow (t_i = t_j)$
	|	$\mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset$	
	|	$\mathbf{duplicates}\,(x_1, \,..\, , x_n) = \emptyset$	
	|	$x \notin \mathbf{dom}\,(E^{\mathrm{L}})$	
	|	$x \notin \mathbf{dom}\,(E^{\mathrm{X}})$	
	|	$x \notin \mathbf{dom}\,(E^{\mathrm{F}})$	
	|	$p \notin \mathbf{dom}\,(\delta)$	
	|	$p \notin \mathbf{dom}\,(\Delta)$	
	|	$\mathbf{FV}\,(t) \subset \mathit{tnvs}$	Free type variables
	|	$\mathbf{FV}\,(\mathit{t\_multi}) \subset \mathit{tnvs}$	Free type variables
	|	$\mathbf{FV}\,(\mathcal{C}) \subset \mathit{tnvs}$	Free type variables
	|	$\mathit{inst}\,\mathbf{IN}\,I$	
	|	$(p\ t) \notin I$	
	|	$E_1^{\mathrm{L}} = E_2^{\mathrm{L}}$	
	|	$E_1^{\mathrm{X}} = E_2^{\mathrm{X}}$	
	|	$E_1^{\mathrm{F}} = E_2^{\mathrm{F}}$	
	|	$E_1 = E_2$	
	|	$\Delta_1 = \Delta_2$	
	|	$\delta_1 = \delta_2$	
	|	$I_1 = I_2$	
	|	$\mathit{names}_1 = \mathit{names}_2$	
	|	$t_1 = t_2$	
	|	$\sigma_1 = \sigma_2$	
	|	$p_1 = p_2$	
	|	$\mathit{xs}_1 = \mathit{xs}_2$	
	|	$\mathit{tnvs}_1 = \mathit{tnvs}_2$	
convert_tnvars	::=		
	|	$\mathit{tnvars}^l \leadsto \mathit{tnvs}$	
	|	$\mathit{tnvar}^l \leadsto \mathit{tnv}$	
look_m	::=		
	|	$E_1(x_1^l \,..\, x_n^l) \triangleright E_2$	Name path lookup
look_m_id	::=		
	|	$E_1(\mathit{id}) \triangleright E_2$	Module identifier lookup
look_tc	::=		
	|	$E(\mathit{id}) \triangleright p$	Path identifier lookup
check_t	::=		
	|	$\Delta \vdash t\,\mathbf{ok}$	Well-formed types
	|	$\Delta, \mathit{tnv} \vdash t\,\mathbf{ok}$	Well-formed type/Nexps matching
teq	::=		
	|	$\Delta \vdash t_1 = t_2$	Type equality

convert_typ	::=		
	|	$\Delta, E \vdash typ \leadsto t$	Convert source types to int
	|	$\vdash Nexp \leadsto ne$	Convert and normalize num
convert_typs	::=		
	|	$\Delta, E \vdash typs \leadsto t\_multi$	
check_lit	::=		
	|	$\vdash lit : t$	Typing literal constants
inst_field	::=		
	|	$\Delta, E \vdash \textbf{field}\ id : p\ t\_args \rightarrow t \rhd (x\ \textbf{of}\ names)$	Field typing (also returns c
inst_ctor	::=		
	|	$\Delta, E \vdash \textbf{ctor}\ id : t\_multi \rightarrow p\ t\_args \rhd (x\ \textbf{of}\ names)$	Data constructor typing (al
inst_val	::=		
	|	$\Delta, E \vdash \textbf{val}\ id : t \rhd \Sigma^{\mathcal{C}}$	Typing top-level bindings, c
not_ctor	::=		
	|	$E, E^{\text{L}} \vdash x\ \textbf{not ctor}$	v is not bound to a data co
not_shadowed	::=		
	|	$E^{\text{L}} \vdash id\ \textbf{not shadowed}$	id is not lexically shadowed
check_pat	::=		
	|	$\Delta, E, E_1^{\text{L}} \vdash pat : t \rhd E_2^{\text{L}}$	Typing patterns, building t
	|	$\Delta, E, E_1^{\text{L}} \vdash pat\_aux : t \rhd E_2^{\text{L}}$	Typing patterns, building t
id_field	::=		
	|	$E \vdash id\ \textbf{field}$	Check that the identifier is
id_value	::=		
	|	$E \vdash id\ \textbf{value}$	Check that the identifier is
check_exp	::=		
	|	$\Delta, E, E^{\text{L}} \vdash exp : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecti
	|	$\Delta, E, E^{\text{L}} \vdash exp\_aux : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecti
	|	$\Delta, E, E_1^{\text{L}} \vdash qbind_1 \mathrel{..} qbind_n \rhd E_2^{\text{L}}, \Sigma^{\mathcal{C}}$	Build the environment for c
	|	$\Delta, E, E_1^{\text{L}} \vdash \textbf{list}\ qbind_1 \mathrel{..} qbind_n \rhd E_2^{\text{L}}, \Sigma^{\mathcal{C}}$	Build the environment for q
	|	$\Delta, E, E^{\text{L}} \vdash funcl \rhd \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a
	|	$\Delta, E, E_1^{\text{L}} \vdash letbind \rhd E_2^{\text{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a
check_rule	::=		
	|	$\Delta, E, E^{\text{L}} \vdash rule \rhd \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a
check_texp_tc	::=		

	|	$xs, \Delta_1, E \vdash \textbf{tc}\, td \triangleright \Delta_2, E^{\mathrm{P}}$	Extract the type constructor informat
check_texps_tc	::=		
	|	$xs, \Delta_1, E \vdash \textbf{tc}\, td_1 \,..\, td_i \triangleright \Delta_2, E^{\mathrm{P}}$	Extract the type constructor informat
check_texp	::=		
	|	$\Delta, E \vdash tnvs\; p = texp \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	Check a type definition, with its path
check_texps	::=		
	|	$xs, \Delta, E \vdash td_1 \,..\, td_n \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	
convert_class	::=		
	|	$\delta, E \vdash id \rightsquigarrow p$	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	::=		
	|	$\Delta, (\alpha_1, \,..\,, \alpha_n) \vdash t\, \textbf{instance}$	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,
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*	

	|	$\mathit{id\_value}$
	|	$\mathit{check\_exp}$
	|	$\mathit{check\_rule}$
	|	$\mathit{check\_texp\_tc}$
	|	$\mathit{check\_texps\_tc}$
	|	$\mathit{check\_texp}$
	|	$\mathit{check\_texps}$
	|	$\mathit{convert\_class}$
	|	$\mathit{solve\_class\_constraint}$
	|	$\mathit{solve\_class\_constraints}$
	|	$\mathit{check\_val\_def}$
	|	$\mathit{check\_t\_instance}$
	|	$\mathit{check\_defs}$
$\mathit{user\_syntax}$	::=	
	|	n
	|	num
	|	nat
	|	hex
	|	bin
	|	string
	|	$\mathit{backtick\_string}$
	|	regexp
	|	x
	|	ix
	|	l
	|	x^l
	|	ix^l
	|	α
	|	α^l
	|	N
	|	N^l
	|	id
	|	tnv
	|	tnvar^l
	|	tnvs
	|	tnvars^l
	|	$\mathit{Nexp\_aux}$
	|	Nexp
	|	$\mathit{Nexp\_constraint\_aux}$
	|	$\mathit{Nexp\_constraint}$
	|	$\mathit{typ\_aux}$
	|	typ
	|	$\mathit{lit\_aux}$
	|	lit
	|	$;^?$

$\mid \quad pat\_aux$
$\mid \quad pat$
$\mid \quad fpat$
$\mid \quad |^?$
$\mid \quad exp\_aux$
$\mid \quad exp$
$\mid \quad q$
$\mid \quad qbind$
$\mid \quad fexp$
$\mid \quad fexps$
$\mid \quad pexp$
$\mid \quad psexp$
$\mid \quad tannot^?$
$\mid \quad funcl\_aux$
$\mid \quad letbind\_aux$
$\mid \quad letbind$
$\mid \quad funcl$
$\mid \quad name\_t$
$\mid \quad name\_ts$
$\mid \quad rule\_aux$
$\mid \quad rule$
$\mid \quad witness^?$
$\mid \quad check^?$
$\mid \quad functions^?$
$\mid \quad indreln\_name\_aux$
$\mid \quad indreln\_name$
$\mid \quad typs$
$\mid \quad ctor\_def$
$\mid \quad texp$
$\mid \quad name^?$
$\mid \quad td$
$\mid \quad c$
$\mid \quad cs$
$\mid \quad c\_pre$
$\mid \quad typschm$
$\mid \quad instschm$
$\mid \quad target$
$\mid \quad open\_import$
$\mid \quad \tau$
$\mid \quad \tau^?$
$\mid \quad lemma\_typ$
$\mid \quad lemma\_decl$
$\mid \quad dexp$
$\mid \quad declare\_arg$
$\mid \quad component$
$\mid \quad termination\_setting$

| $exhaustivity\_setting$
| $elim\_opt$
| $fixity\_decl$
| $target\_rep\_rhs$
| $target\_rep\_lhs$
| $declare\_def$
| val_def
| $ascii\_opt$
| $instance\_decl$
| $class\_decl$
| val_spec
| def_aux
| def
| $;;^?$
| $defs$
| p
| σ
| t
| ne
| t_args
| t_multi
| nec
| $names$
| $\mathcal{C}$
| env_tag
| v_desc
| f_desc
| xs
| $\Sigma^{\mathcal{C}}$
| $\Sigma^{\mathcal{N}}$
| E
| E^{X}
| E^{F}
| E^{M}
| E^{P}
| E^{L}
| tc_abbrev
| tc_def
| Δ
| δ
| $inst$
| I
| D
| $terminals$
| $formula$

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

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

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

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

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

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

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

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

$\boxed{E_1(id) \triangleright E_2}$ 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

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

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

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

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

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

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

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

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

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

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

$$\frac{\Delta \vdash t_1 = t_2 \quad \Delta \vdash t_2 = t_3}{\Delta \vdash t_1 = t_3} \quad \text{TEQ\_TRANS}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\texttt{<<no parses (char 10): |- num l :*** \_\_num >>}} \quad \text{CHECK\_LIT\_NUM}$$

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

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

$$\frac{}{\texttt{<<multiple parses>>}} \quad \text{CHECK\_LIT\_STRING}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

$\boxed{E^{\text{L}} \vdash id\ \textbf{not shadowed}}$ id is not lexically shadowed

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : t \triangleright E_1^{\text{L}} \quad \ldots \quad \Delta, E, E^{\text{L}} \vdash pat_n : t \triangleright E_n^{\text{L}} \\ \textbf{disjoint doms}\,(E_1^{\text{L}}, \ldots, E_n^{\text{L}}) \\ \textbf{length}\,(pat_1 \ldots pat_n) = nat\end{array}}{\Delta, E, E^{\text{L}} \vdash [|pat_1; \ldots; pat_n\ ;^?|] : \_\_\textbf{vector}\ nat\ t \triangleright E_1^{\text{L}} \uplus \ldots \uplus E_n^{\text{L}}} \quad \text{CHECK\_PAT\_AUX\_VECTOR}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : \_\_\mathbf{vector}\ ne_1\ t \rhd E_1^{\text{L}} \quad \ldots \quad \Delta, E, E^{\text{L}} \vdash pat_n : \_\_\mathbf{vector}\ ne_n\ t \rhd E_n^{\text{L}} \\ \mathbf{disjoint\ doms}\,(E_1^{\text{L}}, \ldots, E_n^{\text{L}}) \\ ne' = ne_1 + \ldots + ne_n\end{array}}{\Delta, E, E^{\text{L}} \vdash [|pat_1 \ldots pat_n|] : \_\_\mathbf{vector}\ ne'\ t \rhd E_1^{\text{L}} \uplus \ldots \uplus E_n^{\text{L}}} \quad \text{CHECK\_PAT\_AUX\_VECTOR}$$

$$\frac{\begin{array}{l}\Delta, E, E^{\text{L}} \vdash pat_1 : t_1 \rhd E_1^{\text{L}} \quad \ldots. \quad \Delta, E, E^{\text{L}} \vdash pat_n : t_n \rhd E_n^{\text{L}} \\ \mathbf{disjoint\ doms}\,(E_1^{\text{L}}, \ldots., E_n^{\text{L}})\end{array}}{\Delta, E, E^{\text{L}} \vdash (pat_1, \ldots., pat_n) : t_1 * \ldots. * t_n \rhd E_1^{\text{L}} \uplus \ldots. \uplus E_n^{\text{L}}} \quad \text{CHECK\_PAT\_AUX\_TUP}$$

$$\frac{\begin{array}{l}\Delta \vdash t\ \mathbf{ok} \\ \Delta, E, E^{\text{L}} \vdash pat_1 : t \rhd E_1^{\text{L}} \quad .. \quad \Delta, E, E^{\text{L}} \vdash pat_n : t \rhd E_n^{\text{L}} \\ \mathbf{disjoint\ doms}\,(E_1^{\text{L}}, .., E_n^{\text{L}})\end{array}}{\Delta, E, E^{\text{L}} \vdash [pat_1; ..; pat_n\, ;^?] : \_\_\mathbf{list}\ t \rhd E_1^{\text{L}} \uplus .. \uplus E_n^{\text{L}}} \quad \text{CHECK\_PAT\_AUX\_LIST}$$

$$\frac{\Delta, E, E_1^{\text{L}} \vdash pat : t \rhd E_2^{\text{L}}}{\Delta, E, E_1^{\text{L}} \vdash (pat) : t \rhd E_2^{\text{L}}} \quad \text{CHECK\_PAT\_AUX\_PAREN}$$

$$\frac{\begin{array}{l}\Delta, E, E_1^{\text{L}} \vdash pat_1 : t \rhd E_2^{\text{L}} \\ \Delta, E, E_1^{\text{L}} \vdash pat_2 : \_\_\mathbf{list}\ t \rhd E_3^{\text{L}} \\ \mathbf{disjoint\ doms}\,(E_2^{\text{L}}, E_3^{\text{L}})\end{array}}{\Delta, E, E_1^{\text{L}} \vdash pat_1 :: pat_2 : \_\_\mathbf{list}\ t \rhd E_2^{\text{L}} \uplus E_3^{\text{L}}} \quad \text{CHECK\_PAT\_AUX\_CONS}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash pat_1 : t_1 \triangleright E^{\mathrm{L}}_1 \quad \ldots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t_n \triangleright E^{\mathrm{L}}_n \\ \Delta, E, E^{\mathrm{L}} \uplus E^{\mathrm{L}}_1 \uplus \ldots \uplus E^{\mathrm{L}}_n \vdash exp : u \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \mathbf{disjoint\ doms}\,(E^{\mathrm{L}}_1, \ldots, E^{\mathrm{L}}_n) \end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{fun}\ pat_1 \ldots pat_n \rightarrow exp\ l : \mathbf{curry}\,((t_1 * \ldots * t_n), u) \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}} \quad \text{CHECK\_EXP\_AUX\_FN}$$

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

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

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

$$\begin{array}{l} \Delta, E, E^{\mathrm{L}} \vdash\ x\ \ : t_1 \rightarrow t_2 \rightarrow t_3 \triangleright {\Sigma^{\mathcal{C}}}_1, {\Sigma^{\mathcal{N}}}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_1 : t_1 \triangleright {\Sigma^{\mathcal{C}}}_2, {\Sigma^{\mathcal{N}}}_2 \\ \Delta, E, E^{\mathrm{L}} \vdash exp_2 : t_2 \triangleright {\Sigma^{\mathcal{C}}}_3, {\Sigma^{\mathcal{N}}}_3 \end{array}$$

```
<<no parses (char 18):  TD,E,E_l |- exp1 ‘***x‘ l exp2 :  t3 gives S_c1 union S_c2 union S_c3,S
```

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

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

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

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

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : \_\_\mathbf{vector}\ \mathit{ne}'\ t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \vdash \mathit{Nexp}_1 \rightsquigarrow \mathit{ne}_1 \\ \vdash \mathit{Nexp}_2 \rightsquigarrow \mathit{ne}_2 \\ \mathit{ne} = \mathit{ne}_2 + (-\mathit{ne}_1)\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}.(\mathit{Nexp}_1..\mathit{Nexp}_2) : \_\_\mathbf{vector}\ \mathit{ne}\ t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \cup \{\mathit{ne}_1 \langle \mathit{ne}_2 \langle \mathit{ne}'\}}$$
CHECK_EXP_AUX_VECTORSUB

$$\frac{\begin{array}{l}E \vdash \mathit{id}\ \mathbf{field} \\ \Delta, E \vdash\ \mathbf{field}\ \mathit{id} : p\ t\_\mathit{args} \to t \triangleright (x\ \mathbf{of}\ \mathit{names}) \\ \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : p\ t\_\mathit{args} \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}.\mathit{id} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$$
CHECK_EXP_AUX_FIELD

$$\frac{\begin{array}{l}\overline{\Delta, E, E^{\mathrm{L}} \vdash \mathit{pat}_i : t \triangleright E^{\mathrm{L}}_i}^{\,i} \\ \overline{\Delta, E, E^{\mathrm{L}} \uplus E^{\mathrm{L}}_i \vdash \mathit{exp}_i : u \triangleright \Sigma^{\mathcal{C}}{}_i, \Sigma^{\mathcal{N}}{}_i}^{\,i} \\ \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : t \triangleright \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime}\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{match}\ \mathit{exp}\ \mathbf{with}\ |^{?}\ \overline{\mathit{pat}_i \to \mathit{exp}_i\ l_i}^{\,i}\ l\ \mathbf{end} : u \triangleright \Sigma^{\mathcal{C}\prime} \cup \overline{\Sigma^{\mathcal{C}}{}_i}^{\,i}, \Sigma^{\mathcal{N}\prime} \cup \overline{\Sigma^{\mathcal{N}}{}_i}^{\,i}}$$
CHECK_EXP_AUX_CASE

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}} \\ \Delta, E \vdash \mathit{typ} \rightsquigarrow t\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash (\mathit{exp} : \mathit{typ}) : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$$
CHECK_EXP_AUX_TYPED

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

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_1 : t_1 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad .... \quad \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_n : t_n \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_n}{\Delta, E, E^{\mathrm{L}} \vdash (\mathit{exp}_1, \ldots., \mathit{exp}_n) : t_1 * \ldots. * t_n \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \ldots. \cup \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}{}_1 \cup \ldots. \cup \Sigma^{\mathcal{N}}{}_n}$$
CHECK_EXP_AUX_TUP

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

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}} \vdash (\mathit{exp}) : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$$
CHECK_EXP_AUX_PAREN

$$\frac{\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{begin}\ \mathit{exp}\ \mathbf{end} : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$$
CHECK_EXP_AUX_BEGIN

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_1 : \_\_\mathbf{bool}\ \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_2 : t \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\ \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_3 : t \triangleright \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_3\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathbf{if}\ \mathit{exp}_1\ \mathbf{then}\ \mathit{exp}_2\ \mathbf{else}\ \mathit{exp}_3 : t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2 \cup \Sigma^{\mathcal{C}}{}_3, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2 \cup \Sigma^{\mathcal{N}}{}_3}$$
CHECK_EXP_AUX_IF

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_1 : t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\ \Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_2 : \_\_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp}_1 :: \mathit{exp}_2 : \_\_\mathbf{list}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2}$$
CHECK_EXP_AUX_CONS

$$\frac{\vdash \mathit{lit} : t}{\Delta, E, E^{\mathrm{L}} \vdash \mathit{lit} : t \triangleright \{\,\}, \{\,\}}$$
CHECK_EXP_AUX_LIT

$$
\frac{\begin{array}{l}
\overline{\Delta \vdash t_i\ \mathbf{ok}}^{\,i} \\
\Delta, E, E^{\text{L}} \uplus \{\, \overline{x_i \mapsto t_i}^{\,i} \,\} \vdash \mathit{exp}_1 : t \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \\
\Delta, E, E^{\text{L}} \uplus \{\, \overline{x_i \mapsto t_i}^{\,i} \,\} \vdash \mathit{exp}_2 : \_\_\mathbf{bool}\ \triangleright \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_2 \\
\mathbf{disjoint\ doms}\,(E^{\text{L}}, \{\, \overline{x_i \mapsto t_i}^{\,i} \,\}) \\
E = \langle E^{\text{M}}, E^{\text{P}}, E^{\text{F}}, E^{\text{X}} \rangle \\
\overline{x_i \not\in \mathbf{dom}\,(E^{\text{X}})}^{\,i}
\end{array}}{\Delta, E, E^{\text{L}} \vdash \{\mathit{exp}_1 | \mathit{exp}_2\} : \_\_\mathbf{set}\ t \triangleright \Sigma^{\mathcal{C}}{}_1 \cup \Sigma^{\mathcal{C}}{}_2, \Sigma^{\mathcal{N}}{}_1 \cup \Sigma^{\mathcal{N}}{}_2} \quad \text{CHECK\_EXP\_AUX\_SET\_COMP}
$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l}\overline{\Delta \vdash t_i\, \textbf{ok}}^{\,i}\\ E^{\text{L}}_2 = \{\, \overline{name\_t_i \to x \;\mapsto t_i}^{\,i}\, \}\\ \Delta, E, E^{\text{L}}_1 \uplus E^{\text{L}}_2 \vdash exp' : \_\_\textbf{bool} \;\triangleright \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime}\\ \Delta, E, E^{\text{L}}_1 \uplus E^{\text{L}}_2 \vdash exp_1 : u_1 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad .. \quad \Delta, E, E^{\text{L}}_1 \uplus E^{\text{L}}_2 \vdash exp_n : u_n \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}\end{array}}{\Delta, E, E^{\text{L}}_1 \vdash x^l_1 : \textbf{forall}\,\overline{name\_t_i}^{\,i}\,.exp' \Longrightarrow x\ l\ exp_1 .. exp_n\ l' \triangleright \{x \mapsto \textbf{curry}\,((u_1 * .. * u_n), \_\_\textbf{bool}\,)\}, \Sigma^{\mathcal{C}\prime} \cup \Sigma^{\mathcal{C}}{}_1 \cup}$$

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

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

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

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

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

$\boxed{xs, \Delta_1, E \vdash \mathbf{tc}\, td_1 \,..\, td_i \triangleright \Delta_2, E^{\mathrm{P}}}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{array}{l} \overline{\Delta, E_1, E^{\mathrm{L}} \vdash \mathit{rule}_i \triangleright \{x_i \mapsto t_i\}, {\Sigma^{\mathcal{C}}}_i, {\Sigma^{\mathcal{N}}}_i}^{\,i} \\ I \vdash \overline{{\Sigma^{\mathcal{C}}}_i}^{\,i} \triangleright \mathcal{C} \\ \overline{\mathbf{FV}\,(t_i) \subset \mathit{tnvs}}^{\,i} \\ \mathbf{FV}\,(\mathcal{C}) \subset \mathit{tnvs} \\ \mathbf{compatible\ overlap}\,(\overline{x_i \mapsto t_i}^{\,i}) \\ E^{\mathrm{L}} = \{\overline{x_i\mapsto t_i}^{\,i}\} \\ E_2 = \langle\{\,\},\{\,\},\{\,\},\{\overline{x_i \mapsto \langle \mathbf{forall}\ \mathit{tnvs}.\mathcal{C} \Rightarrow t_i, \mathbf{let}\rangle}^{\,i}\}\rangle \end{array}$$

<<no parses (char 59): </zj//j/>,<TD,TC,I>,E1 |- indreln targets_opt indreln_names*** </rulei

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

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

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

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

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

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

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

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

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

```
Definition rules:        141 good    4 bad
Definition rule clauses: 435 good    4 bad
```