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
	$\|$	$name\_t \rightarrow x^l$	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
	$\|$	$x^l_1. \, .. \, x^l_n.x^l\ l$		
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
	$\|$	$tnv_1\ ..\ tnv_n$		
$tnvars^l$	$::=$			Type variable lists
	$\|$	$tnvar^l_1\ ..\ tnvar^l_n$		
$Nexp\_aux$	$::=$			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

	$\vert$	$\langle\vert fpat_1 ; \ldots ; fpat_n \;;^? \vert\rangle$	Record patterns
	$\vert$	$[\vert pat_1 ; .. ; pat_n \;;^? \vert]$	Vector patterns
	$\vert$	$[\vert pat_1 \,..\, pat_n \vert]$	Concatenated vector patterns
	$\vert$	$(pat_1 , \ldots. , pat_n)$	Tuple patterns
	$\vert$	$[pat_1 ; .. ; pat_n \;;^?]$	List patterns
	$\vert$	(pat)	
	$\vert$	$pat_1 :: pat_2$	Cons patterns
	$\vert$	$x^l + num$	constant addition patterns
	$\vert$	lit	Literal constant patterns
pat	$::=$		Location-annotated patterns
	$\vert$	$pat\_aux\ l$	
$fpat$	$::=$		Field patterns
	$\vert$	$id = pat\ l$	
$\vert^?$	$::=$		Optional bars
	$\vert$		
	$\vert$	$\vert$	
exp_aux	$::=$		Expressions
	$\vert$	id	Identifiers
	$\vert$	$backtick\_string$	identifier that should be literally used in out
	$\vert$	N	Nexp var, has type num
	$\vert$	**fun** $psexp$	Curried functions
	$\vert$	**function** $\vert^?\ pexp_1 \vert \ldots \vert pexp_n$ **end**	Functions with pattern matching
	$\vert$	$exp_1\ exp_2$	Function applications
	$\vert$	$exp_1\ ix^l\ exp_2$	Infix applications
	$\vert$	$\langle\vert fexps \vert\rangle$	Records
	$\vert$	$\langle\vert exp$ **with** $fexps \vert\rangle$	Functional update for records
	$\vert$	$exp.id$	Field projection for records
	$\vert$	$[\vert exp_1 ; .. ; exp_n \;;^? \vert]$	Vector instantiation
	$\vert$	$exp.(Nexp)$	Vector access
	$\vert$	$exp.(Nexp_1 .. Nexp_2)$	Subvector extraction
	$\vert$	**match** exp **with** $\vert^?\ pexp_1 \vert \ldots \vert pexp_n\ l$ **end**	Pattern matching expressions
	$\vert$	$(exp : typ)$	Type-annotated expressions
	$\vert$	**let** $letbind$ **in** exp	Let expressions
	$\vert$	$(exp_1 , \ldots. , exp_n)$	Tuples
	$\vert$	$[exp_1 ; .. ; exp_n \;;^?]$	Lists
	$\vert$	(exp)	
	$\vert$	**begin** exp **end**	Alternate syntax for (exp)
	$\vert$	**if** exp_1 **then** exp_2 **else** exp_3	Conditionals
	$\vert$	$exp_1 :: exp_2$	Cons expressions
	$\vert$	lit	Literal constants
	$\vert$	$\{exp_1 \vert exp_2\}$	Set comprehensions
	$\vert$	$\{exp_1 \vert$ **forall** $qbind_1 .. qbind_n \vert exp_2\}$	Set comprehensions with explicit binding
	$\vert$	$\{exp_1 ; .. ; exp_n \;;^? \}$	Sets

	$\|$	$q\ qbind_1\ ...\ qbind_n . exp$	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
	$\|$	$\textbf{forall}$	
	$\|$	$\textbf{exists}$	
$qbind$	$::=$		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
	$\|$	$fexp_1 ;\ ... ; fexp_n\ ;^?\ l$	
$pexp$	$::=$		Pattern matches
	$\|$	$pat \rightarrow exp\ l$	
$psexp$	$::=$		Multi-pattern matches
	$\|$	$pat_1\ ...\ pat_n \rightarrow exp\ l$	
$tannot^?$	$::=$		Optional type annotations
	$\|$		
	$\|$	$:\ typ$	
$funcl\_aux$	$::=$		Function clauses
	$\|$	$x^l\ pat_1\ ...\ pat_n\ tannot^? = exp$	
$letbind\_aux$	$::=$		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	

	$\vert$	$(x^l : typ)$	
$name\_ts$	$::=$		Names with optional typ
	$\vert$	$name\_t_0 \,..\, name\_t_n$	
$rule\_aux$	$::=$		Inductively defined relat
	$\vert$	x^l : **forall** $name\_t_1 \,..\, name\_t_i . exp \Longrightarrow x_1^l \; exp_1 \,..\, exp_n$	
$rule$	$::=$		Location-annotated indu
	$\vert$	$rule\_aux \; l$	
$witness^?$	$::=$		Optional witness type na
	$\vert$		
	$\vert$	**witness type** x^l;	
$check^?$	$::=$		Option check name decla
	$\vert$		
	$\vert$	**check** x^l;	
$functions^?$	$::=$		Optional names and typ
	$\vert$		
	$\vert$	$x^l : typ$	
	$\vert$	$x^l : typ; functions^?$	
$indreln\_name\_aux$	$::=$		Name for inductively de
	$\vert$	$[x^l : typschm \; witness^? \; check^? \; functions^?]$	
$indreln\_name$	$::=$		Location-annotated nam
	$\vert$	$indreln\_name\_aux \; l$	
$typs$	$::=$		Type lists
	$\vert$	$typ_1 * \ldots * typ_n$	
$ctor\_def$	$::=$		Datatype definition clau
	$\vert$	x^l **of** $typs$	
	$\vert$	x^l	S Constant constructors
$texp$	$::=$		Type definition bodies
	$\vert$	typ	Type abbreviations
	$\vert$	$\langle\vert x_1^l : typ_1 ; \ldots ; x_n^l : typ_n \; ;^? \vert\rangle$	Record types
	$\vert$	$\vert^? \; ctor\_def_1 \vert \ldots \vert ctor\_def_n$	Variant types
$name^?$	$::=$		Optional name specificat
	$\vert$		
	$\vert$	$[name = regexp]$	
td	$::=$		Type definitions

	$\vert$	$x^l\, tnvars^l\, name^? = texp$	
	$\vert$	$x^l\, tnvars^l\, name^?$	Definitions of opaque types
c	$::=$		Typeclass constraints
	$\vert$	$id\, tnvar^l$	
cs	$::=$		Typeclass and length constraint
	$\vert$		
	$\vert$	$c_1, .., c_i \Rightarrow$	Must have > 0 constraints
	$\vert$	$Nexp\_constraint_1, .., Nexp\_constraint_i \Rightarrow$	Must have > 0 constraints
	$\vert$	$c_1, .., c_i; Nexp\_constraint_1, .., Nexp\_constraint_n \Rightarrow$	Must have > 0 of both form o
c_pre	$::=$		Type and instance scheme prefix
	$\vert$		
	$\vert$	**forall** $tnvar_1^l .. tnvar_n^l . cs$	Must have > 0 type variables
$typschm$	$::=$		Type schemes
	$\vert$	$c\_pre\ typ$	
$instschm$	$::=$		Instance schemes
	$\vert$	$c\_pre(id\ typ)$	
$target$	$::=$		Backend target names
	$\vert$	**hol**	
	$\vert$	**isabelle**	
	$\vert$	**ocaml**	
	$\vert$	**coq**	
	$\vert$	**tex**	
	$\vert$	**html**	
	$\vert$	**lem**	
$open\_import$	$::=$		Open or import statements
	$\vert$	**open**	
	$\vert$	**import**	
	$\vert$	**open import**	
	$\vert$	**include**	
	$\vert$	**include import**	
τ	$::=$		Backend target name lists
	$\vert$	$\{target_1; ..; target_n\}$	
	$\vert$	$\{target_1; ..; target_n\}$	all targets except the listed on
	$\vert$	non_exec	all non-executable targets, use
$\tau^?$	$::=$		Optional targets
	$\vert$		
	$\vert$	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 $::=$

$|$ $target\_rep component\ id\ x_1^l\ ..\ x_n^l$

$|$ $target\_rep component\ id\ tnvars^l$

declare_def $::=$

$|$ $\mathbf{declare}\ \tau^?\ compile\_message id = string$

$|$ $\mathbf{declare}\ \tau^?\ \mathbf{rename}\ \mathbf{module}\ = x^l$

$|$ $\mathbf{declare}\ \tau^?\ \mathbf{rename}\ component\ id = x^l$

$|$ $\mathbf{declare}\ \tau^?\ ascii\_rep component\ id = backtick\_string$

$|$ $\mathbf{declare}\ target target\_rep target\_rep\_lhs = target\_rep\_rhs$

$|$ $\mathbf{declare}\ set\_flag x_1^l = x_2^l$

$|$ $\mathbf{declare}\ \tau^?\ termination\_argument id = termination\_setting$

$|$ $\mathbf{declare}\ \tau^?\ pattern\_match exhaustivity\_setting\ id\ tnvars^l = [id_1;\ \ldots;\ id_n\ ;^?]\ elim\_opt$

val_def $::=$

$|$ $\mathbf{let}\ \tau^?\ letbind$

$|$ $\mathbf{let}\ \mathbf{rec}\ \tau^?\ funcl_1\ \mathbf{and}\ \ldots\ \mathbf{and}\ funcl_n$

$|$ $\mathbf{let}\ \mathbf{inline}\ \tau^?\ letbind$

$|$ $\mathbf{let}\ lem\_transform \tau^?\ letbind$

ascii_opt $::=$

$|$

$|$ $[backtick\_string]$

instance_decl $::=$

$|$ $\mathbf{instance}$

$|$ $default\_instance$

class_decl $::=$

$|$ $\mathbf{class}$

$|$ $\mathbf{class}\ \mathbf{inline}$

val_spec $::=$

$|$ $\mathbf{val}\ x^l\ ascii\_opt : typschm$

def_aux $::=$

$|$ $\mathbf{type}\ td_1\ \mathbf{and}\ \ldots\ \mathbf{and}\ td_n$

$|$ val_def

$|$ $lemma\_decl$

$|$ $declare\_def$

$|$ $\mathbf{module}\ x^l = \mathbf{struct}\ defs\ \mathbf{end}$

$|$ $\mathbf{module}\ x^l = id$

$|$ $open\_import\ id_1\ \ldots\ id_n$

$|$ $open\_import\ \tau^?\ backtick\_string_1\ \ldots\ backtick\_string_n$

$|$ $\mathbf{indreln}\ \tau^?\ indreln\_name_1\ \mathbf{and}\ \ldots\ \mathbf{and}\ indreln\_name_i\ rule_1\ \mathbf{and}\ \ldots\ \mathbf{and}\ rule_n$

	|	val_spec		Top-
	|	$class\_decl(x^l\ tnvar^l)\ \mathbf{val}\ \tau_1^?\ x_1^l\ ascii\_opt_1 : typ_1\ l_1 \ldots \mathbf{val}\ \tau_n^?\ x_n^l\ ascii\_opt_n : typ_n\ l_n\ \mathbf{end}$		Type
	|	$instance\_decl\ instschm\ val\_def_1\ l_1 \ldots val\_def_n\ l_n\ \mathbf{end}$		Type
def	::=			Locatio
	|	$def\_aux\ l$		
$;;^?$	::=			Option
	|			
	|	$;;$		
$defs$	::=			Definiti
	|	$def_1\ ;;_1^? \ldots def_n\ ;;_n^?$		
p	::=			Unique
	|	$x_1. \ldots x_n.x$		
	|	**__list**		
	|	**__bool**		
	|	**__num**		
	|	**__set**		
	|	**__string**		
	|	**__unit**		
	|	**__bit**		
	|	**__vector**		
σ	::=			Type v
	|	$\{tnv_1 \mapsto t_1 \ldots tnv_n \mapsto t_n\}$		
$t,\ u$	::=			Interna
	|	α		
	|	$t_1 \rightarrow t_2$		
	|	$t_1 * \ldots * t_n$		
	|	$p\ t\_args$		
	|	ne		
	|	$\sigma(t)$	M	Mult
	|	$\sigma(tnv)$	M	Singl
	|	$\mathbf{curry}\ (t\_multi, t)$	M	Curr
ne	::=			interna
	|	N		
	|	nat		
	|	$ne_1 * ne_2$		
	|	$ne_1 + ne_2$		
	|	$(-ne)$		
	|	$\mathbf{normalize}\ (ne)$	M	
	|	$ne_1 + \ldots + ne_n$	M	
	|	$\mathbf{bitlength}\ (bin)$	M	

	|	$\textbf{bitlength}\,(hex)$	M	
	|	$\textbf{length}\,(pat_1 \,...\, pat_n)$	M	
	|	$\textbf{length}\,(exp_1 \,...\, exp_n)$	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
	|	$\langle\, \textbf{forall}\, tnvs.t\_multi \to p, (x\ \textbf{of}\ names)\rangle$		Constructors
	|	$\langle\, \textbf{forall}\, tnvs.\mathcal{C} \Rightarrow t, env\_tag\rangle$		Values
f_desc	::=			
	|	$\langle\, \textbf{forall}\, tnvs.p \to t, (x\ \textbf{of}\ names)\rangle$		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
	$\|$	$\{x_1 \mapsto v\_desc_1, \,..\,, x_n \mapsto v\_desc_n\}$		
	$\|$	$E_1^{\mathrm{X}} \uplus \,..\, \uplus E_n^{\mathrm{X}}$	M	
E^{F}	$::=$			Field environments
	$\|$	$\{x_1 \mapsto f\_desc_1, \,..\,, x_n \mapsto f\_desc_n\}$		
	$\|$	$E_1^{\mathrm{F}} \uplus \,..\, \uplus E_n^{\mathrm{F}}$	M	
E^{M}	$::=$			Module environments
	$\|$	$\{x_1 \mapsto E_1, \,..\,, x_n \mapsto E_n\}$		
E^{P}	$::=$			Path environments
	$\|$	$\{x_1 \mapsto p_1, \,..\,, x_n \mapsto p_n\}$		
	$\|$	$E_1^{\mathrm{P}} \uplus \,..\, \uplus E_n^{\mathrm{P}}$	M	
E^{L}	$::=$			Lexical bindings
	$\|$	$\{x_1 \mapsto t_1, \,..\,, x_n \mapsto t_n\}$		
	$\|$	$\{x_1^l \mapsto t_1, \,..\,, x_n^l \mapsto t_n\}$		
	$\|$	$E_1^{\mathrm{L}} \uplus \,..\, \uplus E_n^{\mathrm{L}}$	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 ::=
| $\geq$ >=
| $\rightarrow$ ->
| $\leftarrow$ <-
| $\Longrightarrow$ ==>
| $\langle|$ <|
| $|\rangle$ |>
| $\cap$
| $\cup$
| $\uplus$
| $\notin$
| $\subset$
| $\neq$
| $\emptyset$
| $\langle$
| $\rangle$
| $\vdash$
| $,$
| $\mapsto$
| $\triangleright$
| $\rightsquigarrow$
| $\Rightarrow$
| $-$
| 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
| $\mathbf{dom}\,(E_1^{\mathrm{M}}) \cap \mathbf{dom}\,(E_2^{\mathrm{M}}) = \emptyset$
| $\mathbf{dom}\,(E_1^{\mathrm{X}}) \cap \mathbf{dom}\,(E_2^{\mathrm{X}}) = \emptyset$
| $\mathbf{dom}\,(E_1^{\mathrm{F}}) \cap \mathbf{dom}\,(E_2^{\mathrm{F}}) = \emptyset$
| $\mathbf{dom}\,(E_1^{\mathrm{P}}) \cap \mathbf{dom}\,(E_2^{\mathrm{P}}) = \emptyset$
| $\mathbf{disjoint\,doms}\,(E_1^{\mathrm{L}}, \ldots., E_n^{\mathrm{L}})$ Pairwise disjoint domains
| $\mathbf{disjoint\,doms}\,(E_1^{\mathrm{X}}, \ldots., E_n^{\mathrm{X}})$ Pairwise disjoint domains

	$\mid$	$\textbf{compatible overlap}\,(x_1 \mapsto t_1, \,..\,, x_n \mapsto t_n)$	$(x_i = x_j) \Longrightarrow (t_i = t_j)$
	$\mid$	$\textbf{duplicates}\,(\mathit{tnvs}) = \emptyset$	
	$\mid$	$\textbf{duplicates}\,(x_1, \,..\,, x_n) = \emptyset$	
	$\mid$	$x \notin \textbf{dom}\,(E^{\mathrm{L}})$	
	$\mid$	$x \notin \textbf{dom}\,(E^{\mathrm{X}})$	
	$\mid$	$x \notin \textbf{dom}\,(E^{\mathrm{F}})$	
	$\mid$	$p \notin \textbf{dom}\,(\delta)$	
	$\mid$	$p \notin \textbf{dom}\,(\Delta)$	
	$\mid$	$\textbf{FV}\,(t) \subset \mathit{tnvs}$	Free type variables
	$\mid$	$\textbf{FV}\,(t\_\mathit{multi}) \subset \mathit{tnvs}$	Free type variables
	$\mid$	$\textbf{FV}\,(\mathcal{C}) \subset \mathit{tnvs}$	Free type variables
	$\mid$	$\mathit{inst}\,\textbf{IN}\,I$	
	$\mid$	$(p\ t) \notin I$	
	$\mid$	$E_1^{\mathrm{L}} = E_2^{\mathrm{L}}$	
	$\mid$	$E_1^{\mathrm{X}} = E_2^{\mathrm{X}}$	
	$\mid$	$E_1^{\mathrm{F}} = E_2^{\mathrm{F}}$	
	$\mid$	$E_1 = E_2$	
	$\mid$	$\Delta_1 = \Delta_2$	
	$\mid$	$\delta_1 = \delta_2$	
	$\mid$	$I_1 = I_2$	
	$\mid$	$\mathit{names}_1 = \mathit{names}_2$	
	$\mid$	$t_1 = t_2$	
	$\mid$	$\sigma_1 = \sigma_2$	
	$\mid$	$p_1 = p_2$	
	$\mid$	$\mathit{xs}_1 = \mathit{xs}_2$	
	$\mid$	$\mathit{tnvs}_1 = \mathit{tnvs}_2$	
$\mathit{convert\_tnvars}$	::=		
	$\mid$	$\mathit{tnvars}^l \leadsto \mathit{tnvs}$	
	$\mid$	$\mathit{tnvar}^l \leadsto \mathit{tnv}$	
$\mathit{look\_m}$	::=		
	$\mid$	$E_1(x_1^l \,..\, x_n^l) \triangleright E_2$	Name path lookup
$\mathit{look\_m\_id}$	::=		
	$\mid$	$E_1(\mathit{id}) \triangleright E_2$	Module identifier lookup
$\mathit{look\_tc}$	::=		
	$\mid$	$E(\mathit{id}) \triangleright p$	Path identifier lookup
$\mathit{check\_t}$	::=		
	$\mid$	$\Delta \vdash t\ \textbf{ok}$	Well-formed types
	$\mid$	$\Delta, \mathit{tnv} \vdash t\ \textbf{ok}$	Well-formed type/Nexps matching
teq	::=		
	$\mid$	$\Delta \vdash t_1 = t_2$	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	::=		
	|	$\Delta, E \vdash \textbf{field}\, id : p\ t\_args \to t \triangleright (x\ \textbf{of}\ names)$	Field typing (also returns c
inst_ctor	::=		
	|	$\Delta, E \vdash \textbf{ctor}\, id : t\_multi \to p\ t\_args \triangleright (x\ \textbf{of}\ names)$	Data constructor typing (al
inst_val	::=		
	|	$\Delta, E \vdash \textbf{val}\, id : t \triangleright \Sigma^{\mathcal{C}}$	Typing top-level bindings, c
not_ctor	::=		
	|	$E, E^{\mathrm{L}} \vdash x\ \textbf{not ctor}$	v is not bound to a data co
not_shadowed	::=		
	|	$E^{\mathrm{L}} \vdash id\ \textbf{not shadowed}$	id is not lexically shadowed
check_pat	::=		
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash pat : t \triangleright E_2^{\mathrm{L}}$	Typing patterns, building t
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash pat\_aux : t \triangleright E_2^{\mathrm{L}}$	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	::=		
	|	$\Delta, E, E^{\mathrm{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecti
	|	$\Delta, E, E^{\mathrm{L}} \vdash exp\_aux : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecti
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash qbind_1 \mathrel{..} qbind_n \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for c
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \textbf{list}\ qbind_1 \mathrel{..} qbind_n \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for q
	|	$\Delta, E, E^{\mathrm{L}} \vdash funcl \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash letbind \triangleright E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a
check_rule	::=		
	|	$\Delta, E, E^{\mathrm{L}} \vdash rule \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for a
check_texp_tc	::=		

	$\mid$	$xs, \Delta_1, E \vdash \ \mathbf{tc}\, td \ \triangleright\ \Delta_2, E^{\mathrm{P}}$	Extract the type constructor informat
check_texps_tc	$::=$		
	$\mid$	$xs, \Delta_1, E \vdash \ \mathbf{tc}\, td_1 \,..\, td_i \ \triangleright\ \Delta_2, E^{\mathrm{P}}$	Extract the type constructor informat
check_texp	$::=$		
	$\mid$	$\Delta, E \vdash tnvs\ p = texp \ \triangleright\ \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	Check a type definition, with its path
check_texps	$::=$		
	$\mid$	$xs, \Delta, E \vdash td_1 \,..\, td_n \ \triangleright\ \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	
convert_class	$::=$		
	$\mid$	$\delta, E \vdash id \rightsquigarrow p$	Lookup a type class
solve_class_constraint	$::=$		
	$\mid$	$I \vdash (p\ t)\, \mathbf{IN}\, \mathcal{C}$	Solve class constraint
solve_class_constraints	$::=$		
	$\mid$	$I \vdash \Sigma^{\mathcal{C}} \ \triangleright\ \mathcal{C}$	Solve class constraints
check_val_def	$::=$		
	$\mid$	$\Delta, I, E \vdash val\_def \ \triangleright\ E^{\mathrm{X}}$	Check a value definition
check_t_instance	$::=$		
	$\mid$	$\Delta, (\alpha_1, \,..\,, \alpha_n) \vdash t\, \mathbf{instance}$	Check that t be a typeclass instance
check_defs	$::=$		
	$\mid$	$\overline{z_j}^{\,j}, D_1, E_1 \vdash def \ \triangleright\ D_2, E_2$	Check a definition
	$\mid$	$\overline{z_j}^{\,j}, D_1, E_1 \vdash defs \ \triangleright\ D_2, E_2$	Check definitions, given module path,
judgement	$::=$		
	$\mid$	*convert_tnvars*	
	$\mid$	*look_m*	
	$\mid$	*look_m_id*	
	$\mid$	*look_tc*	
	$\mid$	*check_t*	
	$\mid$	*teq*	
	$\mid$	*convert_typ*	
	$\mid$	*convert_typs*	
	$\mid$	*check_lit*	
	$\mid$	*inst_field*	
	$\mid$	*inst_ctor*	
	$\mid$	*inst_val*	
	$\mid$	*not_ctor*	
	$\mid$	*not_shadowed*	
	$\mid$	*check_pat*	
	$\mid$	*id_field*	

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

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

$$
\begin{array}{ll}
| & \mathit{exhaustivity\_setting} \\
| & \mathit{elim\_opt} \\
| & \mathit{fixity\_decl} \\
| & \mathit{target\_rep\_rhs} \\
| & \mathit{target\_rep\_lhs} \\
| & \mathit{declare\_def} \\
| & \mathit{val\_def} \\
| & \mathit{ascii\_opt} \\
| & \mathit{instance\_decl} \\
| & \mathit{class\_decl} \\
| & \mathit{val\_spec} \\
| & \mathit{def\_aux} \\
| & \mathit{def} \\
| & ;;^{?} \\
| & \mathit{defs} \\
| & p \\
| & \sigma \\
| & t \\
| & \mathit{ne} \\
| & \mathit{t\_args} \\
| & \mathit{t\_multi} \\
| & \mathit{nec} \\
| & \mathit{names} \\
| & \mathcal{C} \\
| & \mathit{env\_tag} \\
| & \mathit{v\_desc} \\
| & \mathit{f\_desc} \\
| & \mathit{xs} \\
| & \Sigma^{\mathcal{C}} \\
| & \Sigma^{\mathcal{N}} \\
| & E \\
| & E^{\mathrm{X}} \\
| & E^{\mathrm{F}} \\
| & E^{\mathrm{M}} \\
| & E^{\mathrm{P}} \\
| & E^{\mathrm{L}} \\
| & \mathit{tc\_abbrev} \\
| & \mathit{tc\_def} \\
| & \Delta \\
| & \delta \\
| & \mathit{inst} \\
| & I \\
| & D \\
| & \mathit{terminals} \\
| & \mathit{formula}
\end{array}
$$

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

$$
\frac{\mathit{tnvar}_1^{l} \rightsquigarrow \mathit{tnv}_1 \quad .. \quad \mathit{tnvar}_n^{l} \rightsquigarrow \mathit{tnv}_n}{\mathit{tnvar}_1^{l} \, .. \, \mathit{tnvar}_n^{l} \rightsquigarrow \mathit{tnv}_1 \, .. \, \mathit{tnv}_n} \quad \textsc{convert\_tnvars\_none}
$$

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

$$\frac{}{\alpha\, l \rightsquigarrow \alpha}\quad \textsc{convert\_tnvar\_a}$$

$$\frac{}{N\, l \rightsquigarrow N}\quad \textsc{convert\_tnvar\_N}$$

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

$$\frac{}{E(\,) \triangleright E}\quad \textsc{look\_m\_none}$$

$$\frac{\begin{array}{l} E^{\mathrm{M}}(x) \triangleright E_1 \\ E_1(\,\overline{y_i^l}^{\,i}\,) \triangleright E_2 \end{array}}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle(x\; l\; \overline{y_i^l}^{\,i}\,) \triangleright E_2}\quad \textsc{look\_m\_some}$$

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

$$\frac{E_1(\,\overline{y_i^l}^{\,i}\; x\; l_1) \triangleright E_2}{E_1(\overline{y_i^l.}^{\,i}\; x\; l_1\; l_2) \triangleright E_2}\quad \textsc{look\_m\_id\_all}$$

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

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

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

$$\frac{}{\Delta \vdash \alpha\,\mathbf{ok}}\quad \textsc{check\_t\_var}$$

$$\frac{\begin{array}{c} \Delta \vdash t_1\,\mathbf{ok} \\ \Delta \vdash t_2\,\mathbf{ok} \end{array}}{\Delta \vdash t_1 \rightarrow t_2\,\mathbf{ok}}\quad \textsc{check\_t\_fn}$$

$$\frac{\Delta \vdash t_1\,\mathbf{ok} \quad .... \quad \Delta \vdash t_n\,\mathbf{ok}}{\Delta \vdash t_1 * \;....\; * t_n\,\mathbf{ok}}\quad \textsc{check\_t\_tup}$$

$$\frac{\begin{array}{l} \Delta(p) \triangleright \mathit{tnv}_1 \,..\, \mathit{tnv}_n\; \mathit{tc\_abbrev} \\ \Delta, \mathit{tnv}_1 \vdash t_1\,\mathbf{ok} \quad .. \quad \Delta, \mathit{tnv}_n \vdash t_n\,\mathbf{ok} \end{array}}{\Delta \vdash p\; t_1 \,..\, t_n\,\mathbf{ok}}\quad \textsc{check\_t\_app}$$

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

$$\frac{\Delta \vdash t\,\mathbf{ok}}{\Delta, \alpha \vdash t\,\mathbf{ok}}\quad \textsc{check\_tlen\_t}$$

$$\frac{}{\Delta, N \vdash \mathit{ne}\,\mathbf{ok}}\quad \textsc{check\_tlen\_len}$$

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

$$\frac{\Delta \vdash t\,\mathbf{ok}}{\Delta \vdash t = t}\quad \textsc{teq\_refl}$$

$$\frac{\Delta \vdash t_2 = t_1}{\Delta \vdash t_1 = t_2}\quad \textsc{teq\_sym}$$

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

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

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

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

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

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

$\boxed{\Delta, E \vdash typ \leadsto t}$ 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

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

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

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

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

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

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

$\boxed{\vdash lit : t}$ 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)

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

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

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

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

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

$\boxed{E, E^{\mathrm{L}} \vdash x\ \mathbf{not\ ctor}}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\boxed{\Delta, E, E^{\text{L}} \vdash exp : t \triangleright \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\boxed{\Delta, E, E_1^{\text{L}} \vdash letbind \triangleright E_2^{\text{L}}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}}$ 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

$$\frac{\begin{array}{l}\overline{\Delta \vdash t_i\,\mathbf{ok}}^{\,i} \\ E_2^{\text{L}} = \{\,\overline{name\_t_i \to x \mapsto t_i}^{\,i}\,\} \\ \Delta, E, E_1^{\text{L}} \uplus E_2^{\text{L}} \vdash exp' : \_\_\mathbf{bool} \triangleright \Sigma^{\mathcal{C}\prime}, \Sigma^{\mathcal{N}\prime} \\ \Delta, E, E_1^{\text{L}} \uplus E_2^{\text{L}} \vdash exp_1 : u_1 \triangleright \Sigma^{\mathcal{C}}{}_1, \Sigma^{\mathcal{N}}{}_1 \quad .. \quad \Delta, E, E_1^{\text{L}} \uplus E_2^{\text{L}} \vdash exp_n : u_n \triangleright \Sigma^{\mathcal{C}}{}_n, \Sigma^{\mathcal{N}}\end{array}}{\Delta, E, E_1^{\text{L}} \vdash x_1^l : \mathbf{forall}\,\overline{name\_t_i}^{\,i}\,.exp' \Longrightarrow x\, l\, exp_1 \,..\, exp_n\, l' \triangleright \{x \mapsto \mathbf{curry}\,((u_1 * .. * u_n), \_\_\mathbf{bool}\,)\}, \Sigma^{\mathcal{C}\prime} \cup \Sigma^{\mathcal{C}}{}_1 \cup}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\boxed{\overline{z_j}^{\,j}, D_1, E_1 \vdash \mathit{defs} \triangleright D_2, E_2}$ 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:        141 good    4 bad
Definition rule clauses: 435 good    4 bad
```