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
	|	$x_1^l.\,..\,x_n^l.x^l\ l$		
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
	|	$tnvar_1^l\ ..\ tnvar_n^l$		
$Nexp\_aux$	::=			Numerical expressions for specifying vector lengths and inde
	|	N		
	|	num		
	|	$Nexp_1 * Nexp_2$		
	|	$Nexp_1 + Nexp_2$		
	|	$(Nexp)$		

Nexp	::=		Location-annotated vector lengths
	|	$\mathit{Nexp\_aux}\ l$	
Nexp_constraint_aux	::=		Whether a vector is bounded or fixed size
	|	$\mathit{Nexp} = \mathit{Nexp}'$	
	|	$\mathit{Nexp} \geq \mathit{Nexp}'$	
Nexp_constraint	::=		Location-annotated Nexp range
	|	$\mathit{Nexp\_constraint\_aux}\ l$	
typ_aux	::=		Types
	|	_	Unspecified type
	|	α^l	Type variables
	|	$\mathit{typ}_1 \to \mathit{typ}_2$	Function types
	|	$\mathit{typ}_1 * \ldots. * \mathit{typ}_n$	Tuple types
	|	Nexp	As a typ to permit applications over Nexps, o
	|	$\mathit{id}\ \mathit{typ}_1\ ..\ \mathit{typ}_n$	Type applications
	|	$\mathit{backtick\_string}\ \mathit{typ}_1\ ..\ \mathit{typ}_n$	Backend-Type applications
	|	(typ)	
typ	::=		Location-annotated types
	|	$\mathit{typ\_aux}\ l$	
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	::=		
	|	$\mathit{lit\_aux}\ l$	Location-annotated literal constants
$;^?$	::=		Optional semi-colons
	|		
	|	;	
pat_aux	::=		Patterns
	|	_	Wildcards
	|	$(\mathit{pat}\ \mathbf{as}\ x^l)$	Named patterns
	|	$(\mathit{pat} : \mathit{typ})$	Typed patterns
	|	$\mathit{id}\ \mathit{pat}_1\ ..\ \mathit{pat}_n$	Single variable and constructor patterns

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

	$\vert$	$q\ qbind_1 \ldots qbind_n . exp$	Logical quantifications
	$\vert$	$[exp_1 \vert\ \textbf{forall}\ qbind_1 \mathrel{..} qbind_n \vert exp_2]$	List comprehensions (all binders mu
	$\vert$	$\textbf{do}\ id\ pat_1 \leftarrow exp_1 ; \mathrel{..} pat_n \leftarrow exp_n ;\ \textbf{in}\ exp\ \textbf{end}$	Do notation for monads
exp	$::=$		Location-annotated expressions
	$\vert$	$exp\_aux\ l$	
q	$::=$		Quantifiers
	$\vert$	$\textbf{forall}$	
	$\vert$	$\textbf{exists}$	
$qbind$	$::=$		Bindings for quantifiers
	$\vert$	x^l	
	$\vert$	$(pat\ \textbf{IN}\ exp)$	Restricted quantifications over sets
	$\vert$	$(pat\ \textbf{MEM}\ exp)$	Restricted quantifications over lists
$fexp$	$::=$		Field-expressions
	$\vert$	$id = exp\ l$	
$fexps$	$::=$		Field-expression lists
	$\vert$	$fexp_1 ; \ldots ; fexp_n\ ;^?\ l$	
$pexp$	$::=$		Pattern matches
	$\vert$	$pat \rightarrow exp\ l$	
$psexp$	$::=$		Multi-pattern matches
	$\vert$	$pat_1 \ldots pat_n \rightarrow exp\ l$	
$tannot^?$	$::=$		Optional type annotations
	$\vert$		
	$\vert$	$:\ typ$	
$funcl\_aux$	$::=$		Function clauses
	$\vert$	$x^l\ pat_1 \ldots pat_n\ tannot^? = exp$	
$letbind\_aux$	$::=$		Let bindings
	$\vert$	$pat\ tannot^? = exp$	Value bindings
	$\vert$	$funcl\_aux$	Function bindings
$letbind$	$::=$		Location-annotated let bindings
	$\vert$	$letbind\_aux\ l$	
$funcl$	$::=$		Location-annotated function clauses
	$\vert$	$funcl\_aux\ l$	
$name\_t$	$::=$		Name or name with type for inductive
	$\vert$	x^l	

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

	|	$x^l\, tnvars^l\, name^? = texp$	
	|	$x^l\, tnvars^l\, name^?$	Definitions of opaque types
c	::=		Typeclass constraints
	|	$id\, tnvar^l$	
cs	::=		Typeclass 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
	|		
	|	$\textbf{forall}\, tnvar^l_1 .. tnvar^l_n . cs$	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 on
	|	**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*	
	|	$\langle\vert dexp_1 ; \ldots ; dexp_n\ ;^{?}\ l \vert\rangle$	
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** $string\ exp_1 \ldots exp_n$	

$|$

$\mathit{target\_rep\_lhs}$	$::=$	
	$\mid$	$\mathbf{target\_rep}\ \mathit{component}\ \mathit{id}\ x_1^l\ ..\ x_n^l$
	$\mid$	$\mathbf{target\_rep}\ \mathit{component}\ \mathit{id}\ \mathit{tnvars}^l$
sort	$::=$	
	$\mid$	$\_$
	$\mid$	$\mathit{backtick\_string}$
$\mathit{sorts\_rhs}$	$::=$	
	$\mid$	$\mathit{sort}_1\ ..\ \mathit{sort}_n$
$\mathit{declare\_def}$	$::=$	
	$\mid$	$\mathbf{declare}\ \tau^?\ \mathbf{compile\_message}\ \mathit{id} = \mathit{string}$
	$\mid$	$\mathbf{declare}\ \tau^?\ \mathbf{rename}\ \mathbf{module}\ = x^l$
	$\mid$	$\mathbf{declare}\ \tau^?\ \mathbf{rename}\ \mathit{component}\ \mathit{id} = x^l$
	$\mid$	$\mathbf{declare}\ \tau^?\ \mathbf{ascii\_rep}\ \mathit{component}\ \mathit{id} = \mathit{backtick\_string}$
	$\mid$	$\mathbf{declare}\ \mathit{target}\ \mathbf{target\_rep}\ \mathit{target\_rep\_lhs} = \mathit{target\_rep\_rhs}$
	$\mid$	$\mathbf{declare}\ \mathit{target}\ \mathbf{target\_sorts}\ \mathit{id} = \mathit{sorts\_rhs}$
	$\mid$	$\mathbf{declare}\ \mathbf{set\_flag}\ x_1^l = x_2^l$
	$\mid$	$\mathbf{declare}\ \tau^?\ \mathbf{termination\_argument}\ \mathit{id} = \mathit{termination\_setting}$
	$\mid$	$\mathbf{declare}\ \tau^?\ \mathbf{pattern\_match}\ \mathit{exhaustivity\_setting}\ \mathit{id}\ \mathit{tnvars}^l = [\mathit{id}_1;\ ...\ ;\mathit{id}_n\ ;^?]\ \mathit{elim\_opt}$
$\mathit{val\_def}$	$::=$	
	$\mid$	$\mathbf{let}\ \tau^?\ \mathit{letbind}$
	$\mid$	$\mathbf{let}\ \mathbf{rec}\ \tau^?\ \mathit{funcl}_1\ \mathbf{and}\ ...\ \mathbf{and}\ \mathit{funcl}_n$
	$\mid$	$\mathbf{let}\ \mathbf{inline}\ \tau^?\ \mathit{letbind}$
	$\mid$	$\mathbf{let}\ \mathbf{lem\_transform}\ \tau^?\ \mathit{letbind}$
$\mathit{ascii\_opt}$	$::=$	
	$\mid$	
	$\mid$	$[\mathit{backtick\_string}]$
$\mathit{instance\_decl}$	$::=$	
	$\mid$	$\mathbf{instance}$
	$\mid$	$\mathbf{default\_instance}$
$\mathit{class\_decl}$	$::=$	
	$\mid$	$\mathbf{class}$
	$\mid$	$\mathbf{class}\ \mathbf{inline}$
$\mathit{val\_spec}$	$::=$	
	$\mid$	$\mathbf{val}\ x^l\ \mathit{ascii\_opt} : \mathit{typschm}$
$\mathit{def\_aux}$	$::=$	
	$\mid$	$\mathbf{type}\ \mathit{td}_1\ \mathbf{and}\ ...\ \mathbf{and}\ \mathit{td}_n$

	|	val_def		Valu
	|	$lemma\_decl$		Lemm
	|	$declare\_def$		a dec
	|	$\mathbf{module}\, x^l = \mathbf{struct}\, defs\, \mathbf{end}$		Mod
	|	$\mathbf{module}\, x^l = id$		Mod
	|	$open\_import\ id_1 \ldots id_n$		impo
	|	$open\_import\ \tau^?\ backtick\_string_1 \ldots backtick\_string_n$		impo
	|	$\mathbf{indreln}\, \tau^?\ indreln\_name_1\ \mathbf{and} \ldots \mathbf{and}\ indreln\_name_i\ rule_1\ \mathbf{and} \ldots \mathbf{and}\ rule_n$		Indu
	|	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
	|	$x_1 . \ldots x_n . x$		
	|	$\mathbf{\_\_list}$		
	|	$\mathbf{\_\_bool}$		
	|	$\mathbf{\_\_num}$		
	|	$\mathbf{\_\_set}$		
	|	$\mathbf{\_\_string}$		
	|	$\mathbf{\_\_unit}$		
	|	$\mathbf{\_\_bit}$		
	|	$\mathbf{\_\_vector}$		
σ	::=			Type v
	|	$\{tnv_1 \mapsto t_1 \ldots tnv_n \mapsto t_n\}$		
$t,\ u$	::=			Interna
	|	α		
	|	$t_1 \to t_2$		
	|	$t_1 * \ldots * t_n$		
	|	$p\ t\_args$		
	|	ne		
	|	$\sigma(t)$	M	Mult
	|	$\sigma(tnv)$	M	Singl
	|	$\mathbf{curry}\,(t\_multi, t)$	M	Curr
ne	::=			interna

		|	N		
		|	nat		
		|	$ne_1 * ne_2$		
		|	$ne_1 + ne_2$		
		|	$(-ne)$		
		|	**normalize** (ne)	M	
		|	$ne_1 + \ldots + ne_n$	M	
		|	**bitlength** (bin)	M	
		|	**bitlength** (hex)	M	
		|	**length** $(pat_1 \ldots pat_n)$	M	
		|	**length** $(exp_1 \ldots exp_n)$	M	
t_args	::=				Lists of types
		|	$t_1 \,..\, t_n$		
		|	$\sigma(t\_args)$	M	Multiple substitutions
t_multi	::=				Lists of types
		|	$(t_1 * \,..\, * t_n)$		
		|	$\sigma(t\_multi)$	M	Multiple substitutions
nec	::=				Numeric expression constraints
		|	$ne \langle nec$		
		|	$ne = nec$		
		|	$ne <= nec$		
		|	ne		
$names$	::=				Sets of names
		|	$\{x_1, \,..\,, x_n\}$		
$\mathcal{C}$	::=				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
		|	$\langle$ **forall** $tnvs.t\_multi \rightarrow p, (x$ **of** $names)\rangle$		Constructors
		|	$\langle$ **forall** $tnvs.\mathcal{C} \Rightarrow t, env\_tag\rangle$		Values
f_desc	::=				
		|	$\langle$ **forall** $tnvs.p \rightarrow t, (x$ **of** $names)\rangle$		Fields
xs	::=				
		|	$x_1 \,..\, x_n$		

$\Sigma^{\mathcal{C}}$	$::=$			Typeclass constraints
	$\|$	$\{(p_1\ t_1), \ldots, (p_n\ t_n)\}$		
	$\|$	$\Sigma^{\mathcal{C}}{}_1 \cup \ldots \cup \Sigma^{\mathcal{C}}{}_n$	M	
$\Sigma^{\mathcal{N}}$	$::=$			Nexp constraint lists
	$\|$	$\{nec_1, \ldots, nec_n\}$		
	$\|$	$\Sigma^{\mathcal{N}}{}_1 \cup \ldots \cup \Sigma^{\mathcal{N}}{}_n$	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
	$\|$	$\{x_1 \mapsto v\_desc_1, \ldots, x_n \mapsto v\_desc_n\}$		
	$\|$	$E^{\mathrm{X}}_1 \uplus \ldots \uplus E^{\mathrm{X}}_n$	M	
E^{F}	$::=$			Field environments
	$\|$	$\{x_1 \mapsto f\_desc_1, \ldots, x_n \mapsto f\_desc_n\}$		
	$\|$	$E^{\mathrm{F}}_1 \uplus \ldots \uplus E^{\mathrm{F}}_n$	M	
E^{M}	$::=$			Module environments
	$\|$	$\{x_1 \mapsto E_1, \ldots, x_n \mapsto E_n\}$		
E^{P}	$::=$			Path environments
	$\|$	$\{x_1 \mapsto p_1, \ldots, x_n \mapsto p_n\}$		
	$\|$	$E^{\mathrm{P}}_1 \uplus \ldots \uplus E^{\mathrm{P}}_n$	M	
E^{L}	$::=$			Lexical bindings
	$\|$	$\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$		
	$\|$	$\{x^l_1 \mapsto t_1, \ldots, x^l_n \mapsto t_n\}$		
	$\|$	$E^{\mathrm{L}}_1 \uplus \ldots \uplus E^{\mathrm{L}}_n$	M	
tc_abbrev	$::=$			Type abbreviations
	$\|$	$.t$		
	$\|$			
tc_def	$::=$			Type and class constructor definitions
	$\|$	$tnvs\ tc\_abbrev$		Type constructors
Δ	$::=$			Type constructor definitions
	$\|$	$\{p_1 \mapsto tc\_def_1, \ldots, p_n \mapsto tc\_def_n\}$		
	$\|$	$\Delta_1 \uplus \Delta_2$	M	
δ	$::=$			Typeclass definitions
	$\|$	$\{p_1 \mapsto xs_1, \ldots, p_n \mapsto xs_n\}$		
	$\|$	$\delta_1 \uplus \delta_2$	M	

$inst$ ::= A typeclass instance, t must not contain nested types
  | $\mathcal{C} \Rightarrow (p\ t)$

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

$terminals$ ::=
  | $\geq$ >=
  | $\rightarrow$ ->
  | $\leftarrow$ <-
  | $\Longrightarrow$ ==>
  | $\langle\vert$ <|
  | $\vert\rangle$ |>
  | $\cap$
  | $\cup$
  | $\uplus$
  | $\notin$
  | $\subset$
  | $\neq$
  | $\emptyset$
  | $\langle$
  | $\rangle$
  | $\vdash$
  | $,$
  | $\mapsto$
  | $\triangleright$
  | $\rightsquigarrow$
  | $\Rightarrow$
  | $\_$
  | 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

	$\mid$	$\mathbf{dom}\,(E^{\mathrm{M}}_1) \cap \mathbf{dom}\,(E^{\mathrm{M}}_2) = \emptyset$	
	$\mid$	$\mathbf{dom}\,(E^{\mathrm{X}}_1) \cap \mathbf{dom}\,(E^{\mathrm{X}}_2) = \emptyset$	
	$\mid$	$\mathbf{dom}\,(E^{\mathrm{F}}_1) \cap \mathbf{dom}\,(E^{\mathrm{F}}_2) = \emptyset$	
	$\mid$	$\mathbf{dom}\,(E^{\mathrm{P}}_1) \cap \mathbf{dom}\,(E^{\mathrm{P}}_2) = \emptyset$	
	$\mid$	$\mathbf{disjoint\,doms}\,(E^{\mathrm{L}}_1, \ldots., E^{\mathrm{L}}_n)$	Pairwise disjoint domains
	$\mid$	$\mathbf{disjoint\,doms}\,(E^{\mathrm{X}}_1, \ldots., E^{\mathrm{X}}_n)$	Pairwise disjoint domains
	$\mid$	$\mathbf{compatible\,overlap}\,(x_1 \mapsto t_1, \ldots, x_n \mapsto t_n)$	$(x_i = x_j) \Longrightarrow (t_i = t_j)$
	$\mid$	$\mathbf{duplicates}\,(\mathit{tnvs}) = \emptyset$	
	$\mid$	$\mathbf{duplicates}\,(x_1, \ldots, x_n) = \emptyset$	
	$\mid$	$x \notin \mathbf{dom}\,(E^{\mathrm{L}})$	
	$\mid$	$x \notin \mathbf{dom}\,(E^{\mathrm{X}})$	
	$\mid$	$x \notin \mathbf{dom}\,(E^{\mathrm{F}})$	
	$\mid$	$p \notin \mathbf{dom}\,(\delta)$	
	$\mid$	$p \notin \mathbf{dom}\,(\Delta)$	
	$\mid$	$\mathbf{FV}\,(t) \subset \mathit{tnvs}$	Free type variables
	$\mid$	$\mathbf{FV}\,(\mathit{t\_multi}) \subset \mathit{tnvs}$	Free type variables
	$\mid$	$\mathbf{FV}\,(\mathcal{C}) \subset \mathit{tnvs}$	Free type variables
	$\mid$	$\mathit{inst}\,\mathbf{IN}\,I$	
	$\mid$	$(p\ t) \notin I$	
	$\mid$	$E^{\mathrm{L}}_1 = E^{\mathrm{L}}_2$	
	$\mid$	$E^{\mathrm{X}}_1 = E^{\mathrm{X}}_2$	
	$\mid$	$E^{\mathrm{F}}_1 = E^{\mathrm{F}}_2$	
	$\mid$	$E_1 = E_2$	
	$\mid$	$\Delta_1 = \Delta_2$	
	$\mid$	$\delta_1 = \delta_2$	
	$\mid$	$I_1 = I_2$	
	$\mid$	$\mathit{names}_1 = \mathit{names}_2$	
	$\mid$	$t_1 = t_2$	
	$\mid$	$\sigma_1 = \sigma_2$	
	$\mid$	$p_1 = p_2$	
	$\mid$	$\mathit{xs}_1 = \mathit{xs}_2$	
	$\mid$	$\mathit{tnvs}_1 = \mathit{tnvs}_2$	
$\mathit{convert\_tnvars}$	$::=$		
	$\mid$	$\mathit{tnvars}^l \rightsquigarrow \mathit{tnvs}$	
	$\mid$	$\mathit{tnvar}^l \rightsquigarrow \mathit{tnv}$	
$\mathit{look\_m}$	$::=$		
	$\mid$	$E_1(x^l_1 \,..\, x^l_n) \triangleright E_2$	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
$\mathit{check\_t}$	$::=$		

	|	$\Delta \vdash t\,\mathbf{ok}$	Well-formed types
	|	$\Delta, \mathit{tnv} \vdash t\,\mathbf{ok}$	Well-formed type/Nexps ma
teq	::=		
	|	$\Delta \vdash t_1 = t_2$	Type equality
convert_typ	::=		
	|	$\Delta, E \vdash \mathit{typ} \leadsto t$	Convert source types to int
	|	$\vdash \mathit{Nexp} \leadsto \mathit{ne}$	Convert and normalize num
convert_typs	::=		
	|	$\Delta, E \vdash \mathit{typs} \leadsto \mathit{t\_multi}$	
check_lit	::=		
	|	$\vdash \mathit{lit} : t$	Typing literal constants
inst_field	::=		
	|	$\Delta, E \vdash\ \mathbf{field}\,\mathit{id} : p\ \mathit{t\_args} \to t \rhd (x\ \mathbf{of}\ \mathit{names})$	Field typing (also returns ca
inst_ctor	::=		
	|	$\Delta, E \vdash\ \mathbf{ctor}\,\mathit{id} : \mathit{t\_multi} \to p\ \mathit{t\_args} \rhd (x\ \mathbf{of}\ \mathit{names})$	Data constructor typing (al
inst_val	::=		
	|	$\Delta, E \vdash\ \mathbf{val}\,\mathit{id} : t \rhd \Sigma^{\mathcal{C}}$	Typing top-level bindings, c
not_ctor	::=		
	|	$E, E^{\mathrm{L}} \vdash x\,\mathbf{not\,ctor}$	v is not bound to a data co
not_shadowed	::=		
	|	$E^{\mathrm{L}} \vdash \mathit{id}\,\mathbf{not\,shadowed}$	id is not lexically shadowed
check_pat	::=		
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{pat} : t \rhd E_2^{\mathrm{L}}$	Typing patterns, building th
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{pat\_aux} : t \rhd E_2^{\mathrm{L}}$	Typing patterns, building th
id_field	::=		
	|	$E \vdash \mathit{id}\,\mathbf{field}$	Check that the identifier is
id_value	::=		
	|	$E \vdash \mathit{id}\,\mathbf{value}$	Check that the identifier is
check_exp	::=		
	|	$\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp} : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecti
	|	$\Delta, E, E^{\mathrm{L}} \vdash \mathit{exp\_aux} : t \rhd \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Typing expressions, collecti
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash \mathit{qbind}_1 \,..\, \mathit{qbind}_n \rhd E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for c
	|	$\Delta, E, E_1^{\mathrm{L}} \vdash\ \mathbf{list}\ \mathit{qbind}_1 \,..\, \mathit{qbind}_n \rhd E_2^{\mathrm{L}}, \Sigma^{\mathcal{C}}$	Build the environment for c
	|	$\Delta, E, E^{\mathrm{L}} \vdash \mathit{funcl} \rhd \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	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 let bin
check_rule	::=		
	|	$\Delta, E, E^{\mathrm{L}} \vdash rule \triangleright \{x \mapsto t\}, \Sigma^{\mathcal{C}}, \Sigma^{\mathcal{N}}$	Build the environment for an induc
check_texp_tc	::=		
	|	$xs, \Delta_1, E \vdash \mathbf{tc}\, td \triangleright \Delta_2, E^{\mathrm{P}}$	Extract the type constructor inforn
check_texps_tc	::=		
	|	$xs, \Delta_1, E \vdash \mathbf{tc}\, td_1 \,..\, td_i \triangleright \Delta_2, E^{\mathrm{P}}$	Extract the type constructor inforn
check_texp	::=		
	|	$\Delta, E \vdash tnvs\, p = texp \triangleright \langle E^{\mathrm{F}}, E^{\mathrm{X}} \rangle$	Check a type definition, with its pa
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	::=		
	|	$\Delta, (\alpha_1, \,..\,, \alpha_n) \vdash t\, \mathbf{instance}$	Check that t be a typeclass instanc
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 pa
judgement	::=		
	|	*convert_tnvars*	
	|	*look_m*	
	|	*look_m_id*	
	|	*look_tc*	
	|	*check_t*	
	|	*teq*	
	|	*convert_typ*	
	|	*convert_typs*	
	|	*check_lit*	
	|	*inst_field*	

$$
\begin{array}{lll}
 & | & \mathit{inst\_ctor} \\
 & | & \mathit{inst\_val} \\
 & | & \mathit{not\_ctor} \\
 & | & \mathit{not\_shadowed} \\
 & | & \mathit{check\_pat} \\
 & | & \mathit{id\_field} \\
 & | & \mathit{id\_value} \\
 & | & \mathit{check\_exp} \\
 & | & \mathit{check\_rule} \\
 & | & \mathit{check\_texp\_tc} \\
 & | & \mathit{check\_texps\_tc} \\
 & | & \mathit{check\_texp} \\
 & | & \mathit{check\_texps} \\
 & | & \mathit{convert\_class} \\
 & | & \mathit{solve\_class\_constraint} \\
 & | & \mathit{solve\_class\_constraints} \\
 & | & \mathit{check\_val\_def} \\
 & | & \mathit{check\_t\_instance} \\
 & | & \mathit{check\_defs}
\end{array}
$$

$$
\begin{array}{lll}
\mathit{user\_syntax} & ::= & \\
 & | & n \\
 & | & \mathit{num} \\
 & | & \mathit{nat} \\
 & | & \mathit{hex} \\
 & | & \mathit{bin} \\
 & | & \mathit{string} \\
 & | & \mathit{backtick\_string} \\
 & | & \mathit{regexp} \\
 & | & x \\
 & | & \mathit{ix} \\
 & | & l \\
 & | & x^l \\
 & | & \mathit{ix}^l \\
 & | & \alpha \\
 & | & \alpha^l \\
 & | & N \\
 & | & N^l \\
 & | & \mathit{id} \\
 & | & \mathit{tnv} \\
 & | & \mathit{tnvar}^l \\
 & | & \mathit{tnvs} \\
 & | & \mathit{tnvars}^l \\
 & | & \mathit{Nexp\_aux} \\
 & | & \mathit{Nexp} \\
 & | & \mathit{Nexp\_constraint\_aux}
\end{array}
$$

$|$ $\mathit{Nexp\_constraint}$
$|$ $\mathit{typ\_aux}$
$|$ typ
$|$ $\mathit{lit\_aux}$
$|$ lit
$|$ $;^?$
$|$ $\mathit{pat\_aux}$
$|$ pat
$|$ fpat
$|$ $|^?$
$|$ $\mathit{exp\_aux}$
$|$ exp
$|$ q
$|$ qbind
$|$ fexp
$|$ fexps
$|$ pexp
$|$ psexp
$|$ $\mathit{tannot}^?$
$|$ $\mathit{funcl\_aux}$
$|$ $\mathit{letbind\_aux}$
$|$ $\mathit{letbind}$
$|$ funcl
$|$ $\mathit{name\_t}$
$|$ $\mathit{name\_ts}$
$|$ $\mathit{rule\_aux}$
$|$ rule
$|$ $\mathit{witness}^?$
$|$ $\mathit{check}^?$
$|$ $\mathit{functions}^?$
$|$ $\mathit{indreln\_name\_aux}$
$|$ $\mathit{indreln\_name}$
$|$ typs
$|$ $\mathit{ctor\_def}$
$|$ texp
$|$ $\mathit{name}^?$
$|$ td
$|$ c
$|$ cs
$|$ $\mathit{c\_pre}$
$|$ $\mathit{typschm}$
$|$ $\mathit{instschm}$
$|$ target
$|$ $\mathit{open\_import}$
$|$ τ
$|$ $\tau^?$

$|$ $lemma\_typ$
$|$ $lemma\_decl$
$|$ $dexp$
$|$ $declare\_arg$
$|$ $component$
$|$ $termination\_setting$
$|$ $exhaustivity\_setting$
$|$ $elim\_opt$
$|$ $fixity\_decl$
$|$ $target\_rep\_rhs$
$|$ $target\_rep\_lhs$
$|$ $sort$
$|$ $sorts\_rhs$
$|$ $declare\_def$
$|$ 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

$| \quad \Delta$
$| \quad \delta$
$| \quad inst$
$| \quad I$
$| \quad D$
$| \quad terminals$
$| \quad formula$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\vdash \mathit{Nexp}_1 \leadsto \mathit{ne}_1 \quad \vdash \mathit{Nexp}_2 \leadsto \mathit{ne}_2}{\vdash \mathit{Nexp}_1 * \mathit{Nexp}_2\ l \leadsto \mathit{ne}_1 * \mathit{ne}_2} \quad \text{CONVERT\_NEXP\_MULT}$$

$$\frac{\vdash \mathit{Nexp}_1 \leadsto \mathit{ne}_1 \quad \vdash \mathit{Nexp}_2 \leadsto \mathit{ne}_2}{\vdash \mathit{Nexp}_1 + \mathit{Nexp}_2\ l \leadsto \mathit{ne}_1 + \mathit{ne}_2} \quad \text{CONVERT\_NEXP\_ADD}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

CHECK_PAT_AUX_RECORD

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash pat_1 : t \vartriangleright E^{\mathrm{L}}_1 \quad \dots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t \vartriangleright E^{\mathrm{L}}_n\\ \textbf{disjoint doms}\,(E^{\mathrm{L}}_1,\ \dots, E^{\mathrm{L}}_n)\\ \textbf{length}\,(pat_1\ \dots\ pat_n) = nat\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash [|pat_1;\ \dots; pat_n\ ;^{?}|] : \_\_\textbf{vector}\ nat\ t \vartriangleright E^{\mathrm{L}}_1 \uplus \dots \uplus E^{\mathrm{L}}_n}$$

CHECK_PAT_AUX_VECTOR

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash pat_1 : \_\_\textbf{vector}\ ne_1\ t \vartriangleright E^{\mathrm{L}}_1 \quad \dots \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : \_\_\textbf{vector}\ ne_n\ t \vartriangleright E^{\mathrm{L}}_n\\ \textbf{disjoint doms}\,(E^{\mathrm{L}}_1,\ \dots, E^{\mathrm{L}}_n)\\ ne' = ne_1 +\ \dots\ + ne_n\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash [|pat_1\ \dots\ pat_n|] : \_\_\textbf{vector}\ ne'\ t \vartriangleright E^{\mathrm{L}}_1 \uplus \dots \uplus E^{\mathrm{L}}_n}$$

CHECK_PAT_AUX_VECTORC

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}} \vdash pat_1 : t_1 \vartriangleright E^{\mathrm{L}}_1 \quad \dots. \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t_n \vartriangleright E^{\mathrm{L}}_n\\ \textbf{disjoint doms}\,(E^{\mathrm{L}}_1,\ \dots., E^{\mathrm{L}}_n)\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash (pat_1,\ \dots., pat_n) : t_1 *\ \dots.\ * t_n \vartriangleright E^{\mathrm{L}}_1 \uplus \dots. \uplus E^{\mathrm{L}}_n}$$

CHECK_PAT_AUX_TUP

$$\frac{\begin{array}{l}\Delta \vdash t\ \textbf{ok}\\ \Delta, E, E^{\mathrm{L}} \vdash pat_1 : t \vartriangleright E^{\mathrm{L}}_1 \quad .. \quad \Delta, E, E^{\mathrm{L}} \vdash pat_n : t \vartriangleright E^{\mathrm{L}}_n\\ \textbf{disjoint doms}\,(E^{\mathrm{L}}_1,\ .., E^{\mathrm{L}}_n)\end{array}}{\Delta, E, E^{\mathrm{L}} \vdash [pat_1;\ ..; pat_n\ ;^{?}] : \_\_\textbf{list}\ t \vartriangleright E^{\mathrm{L}}_1 \uplus .. \uplus E^{\mathrm{L}}_n}$$

CHECK_PAT_AUX_LIST

$$\frac{\Delta, E, E^{\mathrm{L}}_1 \vdash pat : t \vartriangleright E^{\mathrm{L}}_2}{\Delta, E, E^{\mathrm{L}}_1 \vdash (pat) : t \vartriangleright E^{\mathrm{L}}_2}$$

CHECK_PAT_AUX_PAREN

$$\frac{\begin{array}{l}\Delta, E, E^{\mathrm{L}}_1 \vdash pat_1 : t \vartriangleright E^{\mathrm{L}}_2\\ \Delta, E, E^{\mathrm{L}}_1 \vdash pat_2 : \_\_\textbf{list}\ t \vartriangleright E^{\mathrm{L}}_3\\ \textbf{disjoint doms}\,(E^{\mathrm{L}}_2, E^{\mathrm{L}}_3)\end{array}}{\Delta, E, E^{\mathrm{L}}_1 \vdash pat_1 :: pat_2 : \_\_\textbf{list}\ t \vartriangleright E^{\mathrm{L}}_2 \uplus E^{\mathrm{L}}_3}$$

CHECK_PAT_AUX_CONS

$$\frac{\vdash lit : t}{\Delta, E, E^{\mathrm{L}} \vdash lit : t \vartriangleright \{\,\}}$$

CHECK_PAT_AUX_LIT

$$\frac{E, E^{\mathrm{L}} \vdash x\ \textbf{not ctor}}{\Delta, E, E^{\mathrm{L}} \vdash x\ l + num : \_\_\textbf{num}\ \ \vartriangleright \{x \mapsto \_\_\textbf{num}\ \}}$$

CHECK_PAT_AUX_NUM_ADD

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

$$\frac{E^{\mathrm{F}}(x) \vartriangleright f\_desc}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \vdash\ x\ l_1\ l_2\ \textbf{field}}$$

ID_FIELD_EMPTY

$$\frac{\begin{array}{l}E^{\mathrm{M}}(x) \vartriangleright E\\ x \notin \textbf{dom}\,(E^{\mathrm{F}})\\ E \vdash \overline{y^{l}_{i}.}^{\,i}\ z^{l}\ l_2\ \textbf{field}\end{array}}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \vdash x\ l_1 .\ \overline{y^{l}_{i}.}^{\,i}\ z^{l}\ l_2\ \textbf{field}}$$

ID_FIELD_CONS

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

$$\frac{E^{\mathrm{X}}(x) \vartriangleright v\_desc}{\langle E^{\mathrm{M}}, E^{\mathrm{P}}, E^{\mathrm{F}}, E^{\mathrm{X}}\rangle \vdash\ x\ l_1\ l_2\ \textbf{value}}$$

ID_VALUE_EMPTY

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\begin{array}{l}tnvars^l \leadsto tnvs \\ \Delta, E \vdash typ \leadsto t \\ \textbf{duplicates}\,(tnvs) = \emptyset \\ \textbf{FV}\,(t) \subset tnvs \\ \overline{y_i.}^{\,i}\, x \not\in \textbf{dom}\,(\Delta)\end{array}}{\overline{y_i}^{\,i}, \Delta, E \vdash\; \textbf{tc}\; x\; l\; tnvars^l \;= typ \triangleright \{\overline{y_i.}^{\,i}\, x \mapsto tnvs\,.t\}, \{x \mapsto \overline{y_i.}^{\,i}\, x\}} \quad \text{CHECK\_TEXP\_TC\_ABBREV}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Definition rules:        141 good    0 bad
Definition rule clauses: 425 good    0 bad
```