k	index variable for class type variables and arguments
l	index variable for method type variables and arguments
q	index variable for method parameters and arguments
i, j	index variables as arbitrary elements
n	index variable as upper limit
f	field identifier
mid	method identifier
pid	parameter identifier
X	type variable identifier
Cid	derived class identifier
$RAId$	raw address identifier

terminals	::=		
	|	`class`	keyword: class declaration
	|	`extends`	keyword: super type declaration
	|	`new`	keyword: object creation
	|	`this`	keyword: current object
	|	`null`	keyword: null value
	|	`pure`	keyword: pure method
	|	`impure`	keyword: non-pure method
	|	`<`	syntax: start generics
	|	`>`	syntax: end generics
	|	$\{$	syntax: start block
	|	$\}$	syntax: end block
	|	`(`	syntax: start parameters
	|	`)`	syntax: end parameters
	|	`;`	syntax: separator
	|	`.`	syntax: selector
	|	`=`	syntax: assignment
	|	`Object`	name of root class
	|	$\in$	containment judgement
	|	$\notin$	non-containment jdugement
	|	$\vdash$	single element judgement
	|	$\vdash$	multiple element judgement
	|	$:$	separator
	|	$:_s$	strict separator
	|	$\mapsto$	maps-to
	|	OK	well-formedness judgement
	|	strictly OK	strict well-formedness judgement
	|	pure	purity judgement
	|	strictly pure	strict purity judgement
	|	enc	encapsulation judgement
	|	prg OK	programmer well-formedness judgement
	|	$\sqsubseteq$	subclassing
	|	$<:$	subtyping of single type
	|	$<:_u$	ordering of ownership modifiers
	|	$<:_l$	argument ordering
	|	$<:_s$	strict ordering
	|	$<:_i$	invariant ordering
	|	$=$	alias
	|	$\neq$	not alias
	|	$=$	multiple alias
	|	$=_o$	option alias
	|	$\neq_o$	option not alias
	|	$=_o$	optional, multiple alias
	|	$\subseteq$	subset relation
	|	$\cup$	append two lists
	|	$\vee$	logical or

	|	$\wedge$		logical and
	|	**AND**		top-level logical and
	|	$\Longrightarrow$		logical implication
	|	$\texttt{null}_a$		special null address
	|	$\texttt{any}_a$		special any address
	|	$\texttt{root}_a$		special root address
	|	$\texttt{lost}_a$		special lost address

formula	::=			formulas
	|	otherwise		none of the previous rules applied
	|	*judgement*		judgement
	|	$formula_1 \quad , \quad .. \quad , \quad formula_k$		sequence
	|	$(formula)$		bracketed
	|	$!formula$		negation
	|	$formula \ \vee \ formula'$		logical or
	|	$formula \ \wedge \ formula'$		logical and
	|	$formula \qquad formula'$		top-level logical and
	|	$formula \Longrightarrow formula'$		implies
	|	$\overline{{}^s fml}$		static formulas
	|	$\overline{{}^r fml}$		runtime formulas
	|	$\forall f \in \overline{f}.\ formula$		for all f in $\overline{f}$ holds *formula*
	|	$\forall i \in \{1 \ldots k\}.\ formula$		for all i in $\{1 \ldots k\}$ holds *formula*
	|	$\exists\, \overline{{}^s T}.\ formula$		exists $\overline{{}^s T}$ such that *formula*
	|	$\forall\, \overline{{}^s T}.\ formula$		for all $\overline{{}^s T}$ holds *formula*
	|	$\forall {}^s CT.\ formula$		for all ${}^s CT$ holds *formula*
	|	$\forall C, C'.\ formula$		for all C and C' holds *formula*
	|	$\forall {}^r T.\ formula$		for all ${}^r T$ holds *formula*
	|	$\forall {}^o\iota \in \overline{{}^o\iota}.\ formula$		for all ${}^o\iota$ in $\overline{{}^o\iota}$ holds *formula*
	|	$\forall \iota \in \overline{\iota}, f \in \overline{fv}.\ formula$		for all ι in $\overline{\iota}$ and field f in $\overline{fv}$ holds *formula*
	|	$\forall \overline{\iota} \in \mathcal{P}(\overline{{}^o\iota}).\ formula$		for all $\overline{\iota}$ from $\overline{{}^o\iota}$ holds *formula*
	|	$\forall f \in \overline{fv}.\ formula$		for all f in $\overline{fv}$ holds *formula*

C	::=			class name
	|	*Cid*		derived class identifier
	|	`Object`		name of base class
	|	_	M	some class name
	|	$\mathrm{ClassOf}({}^s N)$	M	class of a static non-variable type
	|	$\mathrm{ClassOf}({}^r T)$	M	class of a runtime type

u	::=			ownership modifier
	|	`self`		current object
	|	`peer`		same context
	|	`rep`		representation context
	|	`any`		any context
	|	`lost`		lost context
	|	_	M	some ownership modifier
	|	$\mathrm{om}({}^s N)$	M	ownership modifier of non-variable type

	|	$\mathrm{om}({}^sT, {}^s\Gamma)$	M	ownership modifier of static type
$\overline{u}$	::=			ownership modifiers
	|	$u_1, .., u_n$		ownership modifier list
	|	$\{\overline{u}\}$	M	notation
${}^sCT$	::=			class type
	|	$C<\overline{{}^sT}>$		generic class type
${}^sT$	::=			static type
	|	${}^sN$		non-variable type
	|	X		type variable
	|	$\_$	M	some type
	|	${}^sT\left[\overline{{}^sT}/\overline{X}\right]$	M	substitution of $\overline{{}^sT}$ for $\overline{X}$ in ${}^sT$
$\overline{{}^sT}$	::=			static types
	|	${}^sT_1, .., {}^sT_n$		static type list
	|	$\overline{{}^sT}_1,\ \overline{{}^sT}_2$		two static type lists
	|	$\overline{{}^sN}$		non-variable type list
	|	$\overline{X}$		variable type list
	|	$\emptyset$		no static types
	|	$\_$	M	some static types
	|	$\overline{{}^sT}'\left[\overline{{}^sT}/\overline{X}\right]$	M	substitution of $\overline{{}^sT}$ for $\overline{X}$ in each type in $\overline{{}^sT}'$
${}^sT_o$	::=			static type option
	|	${}^sT$		lifted static type
	|	${}^sN_o$		non-variable type option
	|	${}^s\Gamma(x)$	M	look up parameter type
$\overline{{}^sT}_o$	::=			static types option
	|	$\overline{{}^sT}$		lifted static types
${}^sN$	::=			non-variable type
	|	$u\ C<\overline{{}^sT}>$		definition
	|	$\_$	M	some non-variable type
$\overline{{}^sN}$	::=			non-variable types
	|	${}^sN$		one non-variable type
	|	$\overline{{}^sN}_1, .., \overline{{}^sN}_n$		non-variable type list
	|	$\emptyset$		no non-variable types
	|	$\_$	M	some non-variable types
${}^sN_o$	::=			non-variable type option
	|	${}^sN$		lifted non-variable type
	|	${}^s\Gamma(X)$	M	look up upper bound of type variable
$\overline{{}^sN}_o$	::=			non-variable types option

	|	$\overline{{}^{s}N}$		lifted non-variable types
$\overline{X}$	::=			type variables
	|	X		one type variable
	|	$\overline{X}_1, .., \overline{X}_n$		type variable list
	|	$\emptyset$		no type variables
	|	$\_$	M	some type variables
	|	$\mathrm{free}(\overline{{}^{s}T})$	M	type variables in $\overline{{}^{s}T}$
$\overline{X}_o$	::=			type variables option
	|	$\overline{X}$		lifted type variables
P	::=			program
	|	$\overline{Cls},\ C,\ e$		
Cls	::=			class declaration
	|	`class` $ClsHd$ { $\overline{fd}$ $\overline{md}$ }		class declaration
	|	`class Object` {}		declaration of base class
$\overline{Cls}$	::=			class declarations
	|	$Cls_1 .. Cls_n$		class declaration list
$ClsHd$	::=			class header
	|	Cid`<`$\overline{TP}$`> extends` C`<`$\overline{{}^{s}T}$`>`		generic class header declaration
$\overline{TP}$	::=			type parameters
	|	X `extends` ${}^{s}N$		type variable X has upper bound ${}^{s}N$
	|	$\overline{TP}_1, .., \overline{TP}_k$		type parameter list
	|	$\_$	M	some unspecified type parameters
$\overline{fd}$	::=			field declarations
	|	${}^{s}T\ f$`;`		type ${}^{s}T$ and field name f
	|	$\overline{fd}_1 .. \overline{fd}_n$		field declaration list
	|	$\_$	M	some field declarations
$\overline{f}$	::=			list of field identifiers
	|	$f_1 .. f_n$		field identifier list
	|	$\mathrm{fields}(C)$	M	recursive fields look-up
e	::=			expression
	|	`null`		null expression
	|	x		variable read
	|	`new` ${}^{s}T$`()`		object construction
	|	e`.`f		field read
	|	e_0`.`f `=` e_1		field write
	|	e_0`.`m`<`$\overline{{}^{s}T}$`>(`$\overline{e}$`)`		method call
	|	`(`${}^{s}T$`)` e		cast

e_o	::=			expression option
	|	e		lifted expression
$\overline{e}$	::=			expressions
	|	$e_1, \ldots, e_k$		list of expressions
	|	$\emptyset$		empty list
md	::=			method declaration
	|	$ms\ \{\ e\ \}$		method signature and method body
$\overline{md}$	::=			method declarations
	|	md		method declaration
	|	$\overline{md}_1 \ldots \overline{md}_n$		method declaration list
	|	$\_$	M	some method declarations
ms	::=			method signature
	|	$p\ \texttt{<}\overline{TP}\texttt{>}\ {}^sT\ m\texttt{(}\overline{mpd}\texttt{)}$		method signature definition
	|	$ms[\overline{{}^sT}/\overline{X}]$	M	substitution of $\overline{{}^sT}$ for $\overline{X}$ in ms
ms_o	::=			method signature option
	|	ms		lifted method signature
	|	$None$		no method signature defined
p	::=			method purity modifiers
	|	`pure`		side-effect free
	|	`impure`		side effects possible
	|	$\_$	M	some purity modifier
m	::=			method name
	|	mid		method identifier
	|	$\mathrm{MName}(ms)$	M	extract method name from signature
$\overline{mpd}$	::=			method parameter declarations
	|	${}^sT\ pid$		type and parameter name
	|	$\overline{mpd}_1, \ldots, \overline{mpd}_q$		list
	|	$\_$	M	some method parameter declarations
x	::=			parameter name
	|	pid		parameter identifier
	|	`this`		name of current object
${}^s\Gamma$	::=			static environment
	|	$\{{}^s\gamma\ ;\ {}^s\delta\}$		composition
${}^s\gamma_e$	::=			static type environment entry
	|	$X \mapsto {}^sN$		type variable X has upper bound ${}^sN$

$^s\gamma$	$::=$			static type environment
	$\mid$	$^s\gamma_{e1}, .., {}^s\gamma_{en}$		mapping list
	$\mid$	$\emptyset$		empty type environment
$^s\delta_p$	$::=$			static variable parameter environment
	$\mid$	$pid \mapsto {}^sT$		variable pid has static type ^sT
$^s\delta_t$	$::=$			static variable environment for `this`
	$\mid$	`this` $\mapsto {}^sN$		variable `this` has static type ^sN
$^s\delta$	$::=$			static variable environment
	$\mid$	$^s\delta_t$		mapping for `this`
	$\mid$	$^s\delta_t, \_$		mapping for `this` and some others
	$\mid$	$^s\delta_t, {}^s\delta_{p1}, .., {}^s\delta_{pq}$		mappings list
$\overline{^sfml}$	$::=$			static formulas
	$\mid$	$^sT{=}^sT'$		type alias
	$\mid$	$^sT_o{=}_o{}^sT$		type option alias
	$\mid$	$^sT{\neq}^sT'$		type not alias
	$\mid$	$\overline{^sT}{=}\overline{^sT}'$		types alias
	$\mid$	$\overline{X} \subseteq \overline{X}'$		type variables $\overline{X}$ subset of $\overline{X}'$
	$\mid$	$^sT \in \overline{^sT}$		type ^sT contained in list $\overline{^sT}$
	$\mid$	$C{=}C'$		class name alias
	$\mid$	$C{\neq}C'$		class name not alias
	$\mid$	$^s\Gamma{=}^s\Gamma'$		static environment alias
	$\mid$	$ms{=}ms'$		method signature alias
	$\mid$	$ms_o{=}_o ms'_o$		method signature option alias
	$\mid$	$m{=}m'$		method name alias
	$\mid$	$m{\neq}m'$		method name not alias
	$\mid$	$e{=}e'$		expression alias
	$\mid$	$Cls \in P$		class definition in program
	$\mid$	`class` C`<`$\overline{TP}$`>`$\ldots \in P$		partial class definition in program
	$\mid$	$X \in {}^s\Gamma$		type variable in static environment
	$\mid$	$x \in {}^s\Gamma$		parameter in static environment
	$\mid$	$u{=}u'$		ownership modifier alias
	$\mid$	$u{\neq}u'$		ownership modifier not alias
	$\mid$	$\overline{u} \in \overline{^sT}$		ownership modifiers in types (ignores Xs)
	$\mid$	$\overline{u} \notin \overline{^sT}$		ownership modifiers not in types (ignores Xs)
	$\mid$	$u \in \overline{u}$		ownership modifier in set of ownership modifiers
	$\mid$	$p{=}p'$		purity alias
ι	$::=$			address identifier
	$\mid$	$RAId$		raw address identifier
	$\mid$	$^r\Gamma($`this`$)$	M	currently active object look-up
	$\mid$	$\_$	M	some address identifier
$\overline{\iota}$	$::=$			address identifiers

	|	$\iota_1, .., \iota_n$		address identifier list
	|	$\emptyset$		empty list
	|	$\_$	M	some address identifier list
	|	$\mathrm{dom}(h)$	M	domain of heap
v	::=			value
	|	ι		address identifier
	|	$\mathtt{null}_a$		null value
	|	$\_$	M	some value
v_o	::=			value option
	|	v		lifted value
	|	$h(\iota . f)$	M	field value look-up
	|	${}^{r}\Gamma(x)$	M	argument value look-up
	|	$\overline{fv}(f)$	M	field value look-up
$\overline{v}$	::=			values
	|	$v_1, .., v_n$		value list
	|	$\emptyset$		empty list
${}^{o}\iota$	::=			owner adress
	|	ι		address of the owner
	|	$\mathtt{root}_a$		root owner
	|	$\mathtt{any}_a$		special any address
	|	$\_$	M	some owner address
${}^{o}\iota_o$	::=			owner adress option
	|	${}^{o}\iota$		lifted owner address
	|	$\mathrm{owner}(h, \iota)$	M	look up owner in heap
	|	$u\,[\sigma]$	M	substitute u using σ
$\overline{{}^{o}\iota}$	::=			owner addresses
	|	${}^{o}\iota_1, .., {}^{o}\iota_n$		owner address list
	|	$\overline{{}^{o}\iota}_1 \cup .. \cup \overline{{}^{o}\iota}_n$		union of owner address lists
	|	$\overline{\iota}$		list of address identifiers
	|	$\emptyset$		empty list
	|	$\_$	M	some list of owner addresses
	|	$\{\overline{{}^{o}\iota}\}$		notation
$\overline{{}^{o}\iota}_o$	::=			owner addresses option
	|	$\overline{{}^{o}\iota}$		lifted owner addresses
	|	$\mathrm{owners}(h, \iota)$	M	look up transitive owners in heap
$\overline{\overline{{}^{o}\iota}}$	::=			list of owner address identifiers
	|	$\overline{{}^{o}\iota}_1 ; .. ; \overline{{}^{o}\iota}_n$		owner address identifiers list
σ_e	::=			runtime substitution

	|	${}^{o}\iota/u$		substitute ${}^{o}\iota$ for u
	|	$\overline{{}^{r}T}/\overline{X}$		substitute $\overline{{}^{r}T}$ for $\overline{X}$
σ	::=			runtime substitutions
	|	$\sigma_{e1}, .., \sigma_{en}$		list of substitutions
${}^{r}T$	::=			runtime type
	|	${}^{o}\iota\ C\texttt{<}\overline{{}^{r}T}\texttt{>}$		definition
${}^{r}T_{o}$	::=			runtime type option
	|	${}^{r}T$		lifted runtime type
	|	$h(\iota)\downarrow_{1}$	M	look up type in heap
	|	${}^{r}\Gamma(X)$	M	look up runtime type of type variable
$\overline{{}^{r}T}$	::=			runtime types
	|	${}^{r}T_{1}, .., {}^{r}T_{n}$		runtime type list
	|	$\emptyset$		no runtime types
	|	$\_$	M	some runtime types
$\overline{{}^{r}T}_{o}$	::=			runtime types option
	|	$\overline{{}^{r}T}$		lifted runtime types
	|	$\overline{{}^{s}T}\,[\sigma]$	M	apply substitutions σ to $\overline{{}^{s}T}$
$\overline{fv}$	::=			field values
	|	$f \mapsto v$		field f has value v
	|	$\overline{fv}_{1}, .., \overline{fv}_{n}$		field value list
	|	$\_$	M	some field values
	|	$h(\iota)\downarrow_{2}$	M	look up field values in heap
	|	$\overline{fv}[f \mapsto v]$	M	update existing field f to v
o	::=			object
	|	$\left({}^{r}T, \overline{fv}\right)$		runtime type ${}^{r}T$ and field values $\overline{fv}$
o_{o}	::=			object option
	|	o		lifted object
	|	$h(\iota)$	M	look up object in heap
he	::=			heap entry
	|	$(\iota \mapsto o)$		address ι maps to object o
h	::=			heap
	|	$\emptyset$		empty heap
	|	$h + he$		add he to h, overwriting existing mappings
${}^{r}\Gamma$	::=			runtime environment
	|	$\{{}^{r}\gamma\ ;\ {}^{r}\delta\}$		composition

${}^r\gamma$	::=		runtime type environment
	|	$X \mapsto {}^rT$	type variable X has runtime type ${}^rT$
	|	${}^r\gamma_1, .., {}^r\gamma_n$	list of mappings
	|	$\emptyset$	empty type environment
${}^r\delta_p$	::=		runtime variable environment parameter entry
	|	$pid \mapsto v$	variable pid has value v
${}^r\delta_t$	::=		runtime variable environment entry for `this`
	|	$\texttt{this} \mapsto \iota$	variable `this` has address ι
${}^r\delta$	::=		runtime variable environment
	|	${}^r\delta_t$	mapping for `this`
	|	${}^r\delta_t, \_$	mapping for `this` and some others
	|	${}^r\delta_t, {}^r\delta_{p_1}, .., {}^r\delta_{p_q}$	mappings list
$\overline{{}^rfml}$	::=		runtime formulas
	|	$h{=}h'$	heap alias
	|	${}^rT{=}{}^rT'$	runtime type alias
	|	${}^rT_o{=}_o{}^rT'_o$	type option alias
	|	$\overline{{}^rT}{=}\overline{{}^rT}'$	runtime types alias
	|	$\overline{{}^rT}_o{=}_o\overline{{}^rT}'$	runtime types option alias
	|	${}^rT \in \overline{{}^rT}$	type ${}^rT$ contained in list $\overline{{}^rT}$
	|	$v{=}v'$	value alias
	|	$v{\neq}v'$	value not alias
	|	$v_o{=}_o v'_o$	value option alias
	|	$v_o{\neq}_o v'_o$	value option not alias
	|	$\iota \in \overline{\iota}$	address in addresses
	|	$\overline{\iota}{\neq}\overline{\iota}'$	addresses not aliased
	|	$\iota \notin \overline{\iota}$	address not in addresses
	|	${}^o\iota{=}{}^o\iota'$	owner address alias
	|	${}^o\iota_o{=}_o{}^o\iota_o{}'$	owner address option alias
	|	${}^o\iota{\neq}{}^o\iota'$	owner address not alias
	|	$\overline{{}^o\iota}{=}\overline{{}^o\iota}'$	owner addresses alias
	|	$\overline{\overline{{}^o\iota}}{=}\overline{\overline{{}^o\iota}}'$	owner addresses alias
	|	$\overline{{}^o\iota}_o{=}_o\overline{{}^o\iota}'_o$	owner addresses option alias
	|	${}^o\iota_o \in \overline{{}^o\iota}_o$	owner address in owner addresses
	|	$\overline{{}^o\iota}_o \subseteq \overline{{}^o\iota}'$	owners option $\overline{{}^o\iota}_o$ contained in $\overline{{}^o\iota}'$
	|	$o{=}o'$	object alias
	|	$o_o{=}_o o'$	object option alias
	|	${}^r\Gamma{=}{}^r\Gamma'$	runtime environment alias
	|	$\overline{fv}{=}\overline{fv}'$	fields alias
	|	$X \in {}^r\Gamma$	type variable in runtime environment
	|	$x \in {}^r\Gamma$	parameter in runtime environment
	|	$f \in \mathrm{dom}(\overline{fv})$	field identifier f contained in domain of $\overline{fv}$
$st\_helpers$	::=		

	|	$\mathrm{FType}(C, f) =_o {}^{s}T_o$	look up field f in class C
	|	$\mathrm{FType}({}^{s}N, f) =_o {}^{s}T_o$	look up field f in type ${}^{s}N$
	|	$\mathrm{MSig}(C, m) =_o ms_o$	look up signature of method m in class C
	|	$\mathrm{MSig}\big({}^{s}N, m, \overline{{}^{s}T}\big) =_o ms_o$	m in ${}^{s}N$ with method type arguments $\overline{{}^{s}T}$ substituted
	|	$\mathrm{ClassDom}(C) =_o \overline{X}_o$	look up type variables of class C
	|	$\mathrm{ClassBnds}(C) =_o \overline{{}^{s}N}_o$	look up bounds of class C
	|	$\mathrm{ClassBnds}({}^{s}N) =_o \overline{{}^{s}N}_o$	look up bounds of type ${}^{s}N$
ucombdef	::=		
	|	$u \rhd u' = u''$	combining two ownership modifiers
	|	$u \rhd {}^{s}T = {}^{s}T'$	ownership modifier - type combination
	|	$u \rhd \overline{{}^{s}T} = \overline{{}^{s}T}'$	ownership modifier - types combination
tcombdef	::=		
	|	${}^{s}N \rhd {}^{s}T =_o {}^{s}T'_o$	type - type combination
	|	${}^{s}N \rhd \overline{{}^{s}T} =_o \overline{{}^{s}T}'_o$	type - types combination
	|	$\big({}^{s}N \rhd \overline{{}^{s}T}\big)\Big[\overline{{}^{s}T}' / \overline{X}'\Big] =_o \overline{{}^{s}T}''_o$	type - types combination and substitution
stsubxing	::=		
	|	$u <:_u u'$	ordering of ownership modifiers
	|	${}^{s}CT \sqsubseteq {}^{s}CT'$	subclassing
	|	${}^{s}\Gamma \vdash {}^{s}T <: {}^{s}T'$	static subtyping
	|	$\vdash {}^{s}T <:_l {}^{s}T'$	type argument subtyping
	|	${}^{s}\Gamma \vdash {}^{s}T <:_s {}^{s}T'$	strict static subtyping
	|	$\vdash \overline{{}^{s}T} <:_l \overline{{}^{s}T}'$	type argument subtypings
	|	${}^{s}\Gamma \vdash \overline{{}^{s}T} <: \overline{{}^{s}T}'$	static subtypings
	|	${}^{s}\Gamma \vdash \overline{{}^{s}T} <:_s \overline{{}^{s}T}'$	strict static subtypings
typerules	::=		
	|	${}^{s}\Gamma \vdash e : {}^{s}T$	expression typing
	|	${}^{s}\Gamma \vdash \overline{e} : \overline{{}^{s}T}$	expression typings
	|	${}^{s}\Gamma \vdash e :_s {}^{s}T$	strict expression typing
	|	${}^{s}\Gamma \vdash \overline{e} :_s \overline{{}^{s}T}$	strict expression typings
wfstatic	::=		
	|	${}^{s}\Gamma \vdash {}^{s}T$ OK	well-formed static type
	|	${}^{s}\Gamma \vdash \overline{{}^{s}T}$ OK	well-formed static types
	|	${}^{s}\Gamma \vdash {}^{s}T$ strictly OK	strictly well-formed static type
	|	${}^{s}\Gamma \vdash \overline{{}^{s}T}$ strictly OK	strictly well-formed static types
	|	Cls OK	well-formed class declaration
	|	${}^{s}\Gamma \vdash {}^{s}T\ f$; OK	well-formed field declaration
	|	${}^{s}\Gamma \vdash \overline{fd}$ OK	well-formed field declarations
	|	${}^{s}\Gamma, C \vdash md$ OK	well-formed method declaration
	|	${}^{s}\Gamma, C \vdash \overline{md}$ OK	well-formed method declarations
	|	${}^{s}CT \vdash m$ OK	method overriding OK
	|	${}^{s}CT, C\langle\overline{{}^{s}T}, \overline{X}\rangle \vdash m$ OK	method overriding OK auxiliary
	|	${}^{s}\Gamma$ OK	well-formed static environment

	|	$\vdash P$ OK	well-formed program
encapsulation	::=		
	|	${}^s\Gamma \vdash e$ enc	encapsulated expression
	|	${}^s\Gamma \vdash \overline{e}$ enc	encapsulated expressions
	|	${}^s\Gamma, C \vdash md$ enc	encapsulated method declaration
	|	${}^s\Gamma, C \vdash \overline{md}$ enc	encapsulated method declarations
	|	Cls enc	encapsulated class declaration
	|	$\vdash P$ enc	encapsulated program
purity	::=		
	|	${}^s\Gamma \vdash e$ pure	pure expression
	|	${}^s\Gamma \vdash e$ strictly pure	strictly pure expression
	|	${}^s\Gamma \vdash \overline{e}$ strictly pure	pure expressions
prgok	::=		
	|	${}^s\Gamma \vdash {}^sT$ prg OK	reasonable static type
	|	${}^s\Gamma, {}^sN'' \vdash {}^sT <: {}^sN, {}^sN'$	reasonable static type argument
	|	${}^s\Gamma, {}^sN'' \vdash \overline{{}^sT} <: \overline{{}^sN}, \overline{{}^sN}'$	reasonable static type arguments
	|	${}^s\Gamma \vdash \overline{{}^sT}$ prg OK	reasonable static types
	|	Cls prg OK	reasonable class declaration
	|	${}^s\Gamma \vdash {}^sT\ f$; prg OK	reasonable field declaration
	|	${}^s\Gamma \vdash \overline{fd}$ prg OK	reasonable field declarations
	|	${}^s\Gamma, C \vdash md$ prg OK	reasonable method declaration
	|	${}^s\Gamma, C \vdash \overline{md}$ prg OK	reasonable method declarations
	|	$\vdash P$ prg OK	reasonable program
	|	${}^s\Gamma \vdash e$ prg OK	reasonable expression
	|	${}^s\Gamma \vdash \overline{e}$ prg OK	reasonable expressions
rt_helpers	::=		
	|	$h + o = (h', \iota)$	add object o to heap h resulting in heap h' and fr
	|	$h[\iota . f = v] =_o h'$	field update in heap
	|	$\mathrm{FType}(h, \iota, f) =_o {}^sT_o$	look up type of field in heap
	|	$\mathrm{MSig}(h, \iota, m) =_o ms_o$	look up method signature of method m at ι
	|	$\mathrm{MBody}(C, m) =_o e_o$	look up most-concrete body of m in class C or a s
	|	$\mathrm{MBody}(h, \iota, m) =_o e_o$	look up most-concrete body of method m at ι
	|	$\mathrm{sdyn}(\overline{{}^sT}, h, \iota, {}^rT, \overline{{}^o\iota}) =_o \overline{{}^rT}$	simple dynamization of types $\overline{{}^sT}$
	|	$\mathrm{ClassBnds}(h, \iota, {}^rT, \overline{{}^o\iota}) =_o \overline{{}^rT}$	upper bounds of type ${}^rT$ from viewpoint ι
rtsubxing	::=		
	|	$h \vdash {}^rT <: {}^rT'$	type ${}^rT$ is a subtype of ${}^rT'$
	|	$h \vdash \overline{{}^rT} <: \overline{{}^rT}'$	runtime subtypings
	|	$h \vdash v_o : {}^rT$	runtime type ${}^rT$ assignable to value v_o
	|	$h, {}^r\Gamma \vdash v : {}^sT$	static type ${}^sT$ assignable to value v (relative to ${}^r\Gamma$
	|	$h, {}^r\Gamma \vdash \overline{v} : \overline{{}^sT}$	static types $\overline{{}^sT}$ assignable to values $\overline{v}$ (relative to
	|	$h, \iota \vdash v_o : {}^sT$	static type ${}^sT$ assignable to value v_o (relative to ι
	|	$\mathrm{dyn}({}^sT, h, {}^r\Gamma, \overline{{}^o\iota}) =_o {}^rT_o$	dynamization of static type (relative to ${}^r\Gamma$)

| $\text{dyn}\left(\overline{{}^{s}T}, h, {}^{r}\Gamma, \overline{\overline{o}_{\iota}}\right) =_o \overline{{}^{r}T_o}$ — dynamization of static types (relative to ${}^{r}\Gamma$)
| $h, {}^{r}\Gamma \vdash {}^{s}N, {}^{s}T; \left(\overline{{}^{s}T}/\overline{X}, \iota\right) =_o {}^{r}\Gamma'$ — validate and create new viewpoint ${}^{r}\Gamma'$

semantics ::=
| ${}^{r}\Gamma \vdash h, e \rightsquigarrow h', v$ — big-step operational semantics
| ${}^{r}\Gamma \vdash h, \overline{e} \rightsquigarrow h', \overline{v}$ — sequential big-step operational semantics
| $\vdash P \rightsquigarrow h, v$ — big-step operational semantics of a program

wfruntime ::=
| $h,\ \iota \vdash {}^{r}T_o$ $\textsf{strictly OK}$ — strictly well-formed runtime type ${}^{r}T_o$
| $h,\ \iota \vdash \overline{{}^{r}T}$ $\textsf{strictly OK}$ — strictly well-formed runtime types
| $h,\ \overline{\iota} \vdash \overline{{}^{r}T}$ $\textsf{strictly OK}$ — strictly well-formed runtime types
| $h \vdash \iota$ $\textsf{OK}$ — well-formed object at an address
| h $\textsf{OK}$ — well-formed heap
| $h, {}^{r}\Gamma : {}^{s}\Gamma$ $\textsf{OK}$ — runtime and static environments correspond

judgement ::=
| *st_helpers*
| *ucombdef*
| *tcombdef*
| *stsubxing*
| *typerules*
| *wfstatic*
| *encapsulation*
| *purity*
| *prgok*
| *rt_helpers*
| *rtsubxing*
| *semantics*
| *wfruntime*

user_syntax ::=
| k
| l
| q
| i
| n
| f
| mid
| pid
| X
| Cid
| $RAId$
| *terminals*
| *formula*
| C
| u

| $\overline{u}$
| ${}^{s}CT$
| ${}^{s}T$
| $\overline{{}^{s}T}$
| ${}^{s}T_o$
| $\overline{{}^{s}T}_o$
| ${}^{s}N$
| $\overline{{}^{s}N}$
| ${}^{s}N_o$
| $\overline{{}^{s}N}_o$
| $\overline{X}$
| $\overline{X}_o$
| P
| Cls
| $\overline{Cls}$
| $ClsHd$
| $\overline{TP}$
| $\overline{fd}$
| $\overline{f}$
| e
| e_o
| $\overline{e}$
| md
| $\overline{md}$
| ms
| ms_o
| p
| m
| $\overline{mpd}$
| x
| ${}^{s}\mathit{\Gamma}$
| ${}^{s}\gamma_e$
| ${}^{s}\gamma$
| ${}^{s}\delta_p$
| ${}^{s}\delta_t$
| ${}^{s}\delta$
| $\overline{{}^{s}fml}$
| ι
| $\overline{\iota}$
| v
| v_o
| $\overline{v}$
| ${}^{o}\iota$
| ${}^{o}\iota_o$
| $\overline{{}^{o}\iota}$
| $\overline{{}^{o}\iota}_o$

$|\ \overline{\overline{\sigma}}_l$
$|\ \sigma_e$
$|\ \sigma$
$|\ {}^rT$
$|\ {}^rT_o$
$|\ \overline{{}^rT}$
$|\ \overline{{}^rT}_o$
$|\ \overline{fv}$
$|\ o$
$|\ o_o$
$|\ he$
$|\ h$
$|\ {}^r\Gamma$
$|\ {}^r\gamma$
$|\ {}^r\delta_p$
$|\ {}^r\delta_t$
$|\ {}^r\delta$
$|\ \overline{{}^rfml}$

$\boxed{\text{FType}(C, f)\ =_o\ {}^sT_o}$ look up field f in class C

$$\frac{\texttt{class}\ Cid\texttt{<\_>}\ \texttt{extends}\ \_\texttt{<\_>}\ \{\ \_\ {}^sT\ f\texttt{;}\ \_\ \_\ \}\in P}{\text{FType}(Cid, f)\ =_o\ {}^sT}\ \text{SFTC\_DEF}$$

$\boxed{\text{FType}({}^sN, f)\ =_o\ {}^sT_o}$ look up field f in type ${}^sN$

$$\frac{\begin{array}{l}\text{FType}(\text{ClassOf}({}^sN), f)\ =_o\ {}^sT_1\\ {}^sN\ \rhd\ {}^sT_1\ =_o\ {}^sT\end{array}}{\text{FType}({}^sN, f)\ =_o\ {}^sT}\ \text{SFTN\_DEF}$$

$\boxed{\text{MSig}(C, m)\ =_o\ ms_o}$ look up signature of method m in class C

$$\frac{\begin{array}{l}\texttt{class}\ Cid\texttt{<\_>}\ \texttt{extends}\ \_\texttt{<\_>}\ \{\ \_\ \_\ ms\ \{\ e\ \}\ \_\ \}\in P\\ \text{MName}(ms) = m\end{array}}{\text{MSig}(Cid, m)\ =_o\ ms}\ \text{SMSC\_DEF}$$

$\boxed{\text{MSig}\left({}^sN, m, \overline{{}^sT}\right)\ =_o\ ms_o}$ m in ${}^sN$ with method type arguments $\overline{{}^sT}$ substituted

$$\frac{\begin{array}{l}\text{MSig}(\text{ClassOf}({}^sN), m)\ =_o\ p\ \texttt{<}\overline{X_l\ \texttt{extends}\ {}^sN_l}^{\,l}\texttt{>}\ {}^sT\ m\texttt{(}\overline{{}^sT'_q\ pid}^{\,q}\texttt{)}\\ \left({}^sN\ \rhd\ \overline{{}^sN_l}^{\,l}\right)\left[\overline{{}^sT_l}^{\,l}/\overline{X_l}^{\,l}\right]\ =_o\ \overline{{}^sN'_l}^{\,l}\qquad ({}^sN\ \rhd\ {}^sT)\left[\overline{{}^sT_l}^{\,l}/\overline{X_l}^{\,l}\right]\ =_o\ {}^sT'\\ \left({}^sN\ \rhd\ \overline{{}^sT'_q}^{\,q}\right)\left[\overline{{}^sT_l}^{\,l}/\overline{X_l}^{\,l}\right]\ =_o\ \overline{{}^sT''_q}^{\,q}\end{array}}{\text{MSig}\left({}^sN, m, \overline{{}^sT_l}^{\,l}\right)\ =_o\ p\ \texttt{<}\overline{X_l\ \texttt{extends}\ {}^sN'_l}^{\,l}\texttt{>}\ {}^sT'\ m\texttt{(}\overline{{}^sT''_q\ pid}^{\,q}\texttt{)}}\ \text{SMSN\_DEF}$$

$\boxed{\text{ClassDom}(C)\ =_o\ \overline{X}_o}$ look up type variables of class C

$$\frac{\texttt{class}\ Cid\texttt{<}\overline{X_k\ \texttt{extends}\ \_}^{\,k}\texttt{>}\ \texttt{extends}\ \_\texttt{<\_>}\ \{\ \_\ \_\ \}\in P}{\text{ClassDom}(Cid)\ =_o\ \overline{X_k}^{\,k}}\ \text{SCD\_NVAR}$$

$$\frac{}{\text{ClassDom}(\texttt{Object})\ =_o\ \emptyset}\ \text{SCD\_OBJECT}$$

$\boxed{\text{ClassBnds}(C)\ =_o\ \overline{{}^sN}_o}$ look up bounds of class C

$$\frac{\texttt{class}\ Cid\texttt{<}\overline{X_k\ \texttt{extends}\ {}^sN_k}^k\texttt{>}\ \texttt{extends}\ \_\texttt{<}\_\texttt{>}\ \{\ \_\ \_\ \} \in P}{\text{ClassBnds}(Cid)\ =_o\ \overline{{}^sN_k}^k}\quad \text{SCBC\_NVAR}$$

$$\frac{}{\text{ClassBnds}(\texttt{Object})\ =_o\ \emptyset}\quad \text{SCBC\_OBJECT}$$

$\boxed{\text{ClassBnds}({}^sN)\ =_o\ \overline{{}^sN}_o}$ look up bounds of type ${}^sN$

$$\frac{\begin{array}{l}\text{ClassBnds}(\text{ClassOf}({}^sN))\ =_o\ \overline{{}^sN}_1\\ {}^sN\ \triangleright\ \overline{{}^sN}_1\ =_o\ \overline{{}^sN}\end{array}}{\text{ClassBnds}({}^sN)\ =_o\ \overline{{}^sN}}\quad \text{SCBN\_DEF}$$

$\boxed{u\ \triangleright\ u'\ =\ u''}$ combining two ownership modifiers

$$\frac{}{\texttt{self}\ \triangleright\ u\ =\ u}\quad \text{UCU\_SELF}$$

$$\frac{}{\texttt{peer}\ \triangleright\ \texttt{peer}\ =\ \texttt{peer}}\quad \text{UCU\_PEER}$$

$$\frac{}{\texttt{rep}\ \triangleright\ \texttt{peer}\ =\ \texttt{rep}}\quad \text{UCU\_REP}$$

$$\frac{}{u\ \triangleright\ \texttt{any}\ =\ \texttt{any}}\quad \text{UCU\_ANY}$$

$$\frac{\text{otherwise}}{u\ \triangleright\ u'\ =\ \texttt{lost}}\quad \text{UCU\_LOST}$$

$\boxed{u\ \triangleright\ {}^sT\ =\ {}^sT'}$ ownership modifier - type combination

$$\frac{}{u\ \triangleright\ X\ =\ X}\quad \text{UCT\_VAR}$$

$$\frac{\begin{array}{l}u\ \triangleright\ u'\ =\ u''\\ u\ \triangleright\ \overline{{}^sT}\ =\ \overline{{}^sT}'\end{array}}{u\ \triangleright\ u'\ C\texttt{<}\overline{{}^sT}\texttt{>}\ =\ u''\ C\texttt{<}\overline{{}^sT}'\texttt{>}}\quad \text{UCT\_NVAR}$$

$\boxed{u\ \triangleright\ \overline{{}^sT}\ =\ \overline{{}^sT}'}$ ownership modifier - types combination

$$\frac{\overline{u\ \triangleright\ {}^sT_k\ =\ {}^sT'_k}^k}{u\ \triangleright\ \overline{{}^sT_k}^k\ =\ \overline{{}^sT'_k}^k}\quad \text{UCTS\_DEF}$$

$\boxed{{}^sN\ \triangleright\ {}^sT\ =_o\ {}^sT'_o}$ type - type combination

$$\frac{\begin{array}{l}u\ \triangleright\ {}^sT\ =\ {}^sT_1\\ {}^sT_1\left[\overline{{}^sT}/\overline{X}\right] = {}^sT' \qquad \text{ClassDom}(C)\ =_o\ \overline{X}\end{array}}{u\ C\texttt{<}\overline{{}^sT}\texttt{>}\ \triangleright\ {}^sT\ =_o\ {}^sT'}\quad \text{TCT\_DEF}$$

$\boxed{{}^sN\ \triangleright\ \overline{{}^sT}\ =_o\ \overline{{}^sT}'_o}$ type - types combination

$$\frac{\overline{{}^sN\ \triangleright\ {}^sT_k\ =_o\ {}^sT'_k}^k}{{}^sN\ \triangleright\ \overline{{}^sT_k}^k\ =_o\ \overline{{}^sT'_k}^k}\quad \text{TCTS\_DEF}$$

$\boxed{\left({}^sN\ \triangleright\ \overline{{}^sT}\right)\left[\overline{{}^sT}'/\overline{X}'\right]\ =_o\ \overline{{}^sT}''_o}$ type - types combination and substitution

$$\frac{{}^{s}N \rhd \overline{{}^{s}T} \;=_o\; \overline{{}^{s}T}_1 \qquad \overline{{}^{s}T}_1\left[\overline{{}^{s}T}'/\overline{X}'\right] = \overline{{}^{s}T}''}{\left({}^{s}N \rhd \overline{{}^{s}T}\right)\left[\overline{{}^{s}T}'/\overline{X}'\right] \;=_o\; \overline{{}^{s}T}''} \quad \text{TCTSSUBSTS\_DEF}$$

$\boxed{u <:_u u'}$ ordering of ownership modifiers

$$\frac{}{\texttt{self} <:_u \texttt{peer}} \quad \text{OMO\_TP}$$

$$\frac{}{\texttt{peer} <:_u \texttt{lost}} \quad \text{OMO\_PL}$$

$$\frac{}{\texttt{rep} <:_u \texttt{lost}} \quad \text{OMO\_RL}$$

$$\frac{}{u <:_u \texttt{any}} \quad \text{OMO\_UA}$$

$$\frac{}{u <:_u u} \quad \text{OMO\_REFL}$$

$\boxed{{}^{s}CT \sqsubseteq {}^{s}CT'}$ subclassing

$$\frac{\texttt{class}\ Cid\texttt{<}\overline{X_k\ \texttt{extends}\ \_}^{k}\texttt{>}\ \texttt{extends}\ C'\texttt{<}\overline{{}^{s}T}\texttt{>}\ \{\ \_\ \_\ \} \in P}{Cid\texttt{<}\overline{X_k}^{k}\texttt{>} \sqsubseteq C'\texttt{<}\overline{{}^{s}T}\texttt{>}} \quad \text{SC1}$$

$$\frac{\texttt{class}\ C\texttt{<}\overline{X_k\ \texttt{extends}\ \_}^{k}\texttt{>}\ldots \in P}{C\texttt{<}\overline{X_k}^{k}\texttt{>} \sqsubseteq C\texttt{<}\overline{X_k}^{k}\texttt{>}} \quad \text{SC2}$$

$$\frac{C\texttt{<}\overline{X}\texttt{>} \sqsubseteq C_1\texttt{<}\overline{{}^{s}T}_1\texttt{>} \qquad C_1\texttt{<}\overline{X}_1\texttt{>} \sqsubseteq C'\texttt{<}\overline{{}^{s}T}'\texttt{>}}{C\texttt{<}\overline{X}\texttt{>} \sqsubseteq C'\texttt{<}\overline{{}^{s}T}'\left[\overline{{}^{s}T}_1/\overline{X}_1\right]\texttt{>}} \quad \text{SC3}$$

$\boxed{{}^{s}\Gamma \vdash {}^{s}T <: {}^{s}T'}$ static subtyping

$$\frac{C\texttt{<}\overline{X}\texttt{>} \sqsubseteq C'\texttt{<}\overline{{}^{s}T}_1\texttt{>} \qquad u\ C\texttt{<}\overline{{}^{s}T}\texttt{>} \rhd \overline{{}^{s}T}_1 \;=_o\; \overline{{}^{s}T}'}{{}^{s}\Gamma \vdash u\ C\texttt{<}\overline{{}^{s}T}\texttt{>} <: u\ C'\texttt{<}\overline{{}^{s}T}'\texttt{>}} \quad \text{ST1}$$

$$\frac{u <:_u u' \qquad \vdash \overline{{}^{s}T} <:_l \overline{{}^{s}T}'}{{}^{s}\Gamma \vdash u\ C\texttt{<}\overline{{}^{s}T}\texttt{>} <: u'\ C\texttt{<}\overline{{}^{s}T}'\texttt{>}} \quad \text{ST2}$$

$$\frac{{}^{s}T = X \;\vee\; {}^{s}\Gamma(X) =_o {}^{s}T}{{}^{s}\Gamma \vdash X <: {}^{s}T} \quad \text{ST3}$$

$$\frac{{}^{s}\Gamma \vdash {}^{s}T <: {}^{s}T_1 \qquad {}^{s}\Gamma \vdash {}^{s}T_1 <: {}^{s}T'}{{}^{s}\Gamma \vdash {}^{s}T <: {}^{s}T'} \quad \text{ST4}$$

$\boxed{\vdash {}^{s}T <:_l {}^{s}T'}$ type argument subtyping

$$\frac{u' \in \{u, \texttt{lost}\} \qquad \vdash \overline{{}^{s}T} <:_l \overline{{}^{s}T}'}{\vdash u\ C\texttt{<}\overline{{}^{s}T}\texttt{>} <:_l u'\ C\texttt{<}\overline{{}^{s}T}'\texttt{>}} \quad \text{AST1}$$

$$\frac{}{\vdash X <:_l X} \quad \text{AST2}$$

$\boxed{{}^s\Gamma \vdash {}^sT <:_s {}^sT'}$ strict static subtyping

$$\frac{\begin{array}{c} {}^s\Gamma \vdash {}^sT <: {}^sT' \\ \texttt{lost} \notin {}^sT' \end{array}}{{}^s\Gamma \vdash {}^sT <:_s {}^sT'} \quad \text{SSTDEF}$$

$\boxed{\vdash \overline{{}^sT} <:_l \overline{{}^sT'}}$ type argument subtypings

$$\frac{\overline{\vdash {}^sT_k <:_l {}^sT'_k}^{\,k}}{\vdash \overline{{}^sT_k}^{\,k} <:_l \overline{{}^sT'_k}^{\,k}} \quad \text{ASTS\_DEF}$$

$\boxed{{}^s\Gamma \vdash \overline{{}^sT} <: \overline{{}^sT'}}$ static subtypings

$$\frac{\overline{{}^s\Gamma \vdash {}^sT_k <: {}^sT'_k}^{\,k}}{{}^s\Gamma \vdash \overline{{}^sT_k}^{\,k} <: \overline{{}^sT'_k}^{\,k}} \quad \text{STS\_DEF}$$

$\boxed{{}^s\Gamma \vdash \overline{{}^sT} <:_s \overline{{}^sT'}}$ strict static subtypings

$$\frac{\overline{{}^s\Gamma \vdash {}^sT_k <:_s {}^sT'_k}^{\,k}}{{}^s\Gamma \vdash \overline{{}^sT_k}^{\,k} <:_s \overline{{}^sT'_k}^{\,k}} \quad \text{SSTS\_DEF}$$

$\boxed{{}^s\Gamma \vdash e : {}^sT}$ expression typing

$$\frac{\begin{array}{c} {}^s\Gamma \vdash e : {}^sT_1 \\ {}^s\Gamma \vdash {}^sT_1 <: {}^sT \\ {}^s\Gamma \vdash {}^sT \ \mathsf{OK} \end{array}}{{}^s\Gamma \vdash e : {}^sT} \quad \text{TR\_SUBSUM}$$

$$\frac{\begin{array}{c} \texttt{self} \notin {}^sT \\ {}^s\Gamma \vdash {}^sT \ \mathsf{OK} \end{array}}{{}^s\Gamma \vdash \texttt{null} : {}^sT} \quad \text{TR\_NULL}$$

$$\frac{{}^s\Gamma(x) =_o {}^sT}{{}^s\Gamma \vdash x : {}^sT} \quad \text{TR\_VAR}$$

$$\frac{\begin{array}{c} {}^s\Gamma \vdash {}^sT \ \mathsf{strictly\ OK} \\ \mathrm{om}({}^sT, {}^s\Gamma) \in \{\texttt{peer}, \texttt{rep}\} \end{array}}{{}^s\Gamma \vdash \texttt{new}\ {}^sT\texttt{()} : {}^sT} \quad \text{TR\_NEW}$$

$$\frac{\begin{array}{c} {}^s\Gamma \vdash e_0 : {}^sN_0 \\ \mathrm{FType}({}^sN_0, f) =_o {}^sT \end{array}}{{}^s\Gamma \vdash e_0\texttt{.}f : {}^sT} \quad \text{TR\_READ}$$

$$\frac{\begin{array}{ll} {}^s\Gamma \vdash e_0 : {}^sN_0 & \mathrm{FType}({}^sN_0, f) =_o {}^sT \\ {}^s\Gamma \vdash e_1 :_s {}^sT & \end{array}}{{}^s\Gamma \vdash e_0\texttt{.}f \texttt{ = } e_1 : {}^sT} \quad \text{TR\_WRITE}$$

$$\frac{\begin{array}{cc} {}^s\Gamma \vdash e_0 : {}^sN_0 & {}^s\Gamma \vdash \overline{{}^sT_l}^{\,l} \ \mathsf{strictly\ OK} \\ \multicolumn{2}{c}{\mathrm{MSig}\left({}^sN_0, m, \overline{{}^sT_l}^{\,l}\right) =_o \_ \texttt{<}\overline{X_l \texttt{ extends } {}^sN_l}^{\,l}\texttt{>}\ {}^sT\ m\texttt{(}\overline{{}^sT'_q\ pid}^{\,q}\texttt{)}} \\ {}^s\Gamma \vdash \overline{e_q}^{\,q} :_s \overline{{}^sT'_q}^{\,q} & {}^s\Gamma \vdash \overline{{}^sT_l}^{\,l} <:_s \overline{{}^sN_l}^{\,l} \end{array}}{{}^s\Gamma \vdash e_0\texttt{.}m\texttt{<}\overline{{}^sT_l}^{\,l}\texttt{>(}\overline{e_q}^{\,q}\texttt{)} : {}^sT} \quad \text{TR\_CALL}$$

$$\frac{\begin{array}{c} {}^s\Gamma \vdash e : \_ \\ {}^s\Gamma \vdash {}^sT \ \mathsf{OK} \end{array}}{{}^s\Gamma \vdash \texttt{(}{}^sT\texttt{)}\ e : {}^sT} \quad \text{TR\_CAST}$$

$\boxed{{}^s\Gamma \vdash \overline{e} : \overline{{}^sT}}$ expression typings

$$\frac{\overline{{}^s\Gamma \vdash e_k : {}^sT_k}^{k}}{{}^s\Gamma \vdash \overline{e_k}^{k} : \overline{{}^sT_k}^{k}} \quad \text{TRM\_DEF}$$

$\boxed{{}^s\Gamma \vdash e :_s {}^sT}$ strict expression typing

$$\frac{\begin{array}{c}{}^s\Gamma \vdash e : {}^sT\\ \texttt{lost} \notin {}^sT\end{array}}{{}^s\Gamma \vdash e :_s {}^sT} \quad \text{STR\_DEF}$$

$\boxed{{}^s\Gamma \vdash \overline{e} :_s \overline{{}^sT}}$ strict expression typings

$$\frac{\overline{{}^s\Gamma \vdash e_k :_s {}^sT_k}^{k}}{{}^s\Gamma \vdash \overline{e_k}^{k} :_s \overline{{}^sT_k}^{k}} \quad \text{STRM\_DEF}$$

$\boxed{{}^s\Gamma \vdash {}^sT \ \ \mathsf{OK}}$ well-formed static type

$$\frac{X \in {}^s\Gamma}{{}^s\Gamma \vdash X \ \ \mathsf{OK}} \quad \text{WFT\_VAR}$$

$$\frac{\begin{array}{cc}{}^s\Gamma \vdash \overline{{}^sT} \ \ \mathsf{OK} & \texttt{self} \notin \overline{{}^sT}\\ \mathrm{ClassBnds}\big(u\ C\texttt{<}\overline{{}^sT}\texttt{>}\big) =_o \overline{{}^sN} & {}^s\Gamma \vdash \overline{{}^sT} <: \overline{{}^sN}\end{array}}{{}^s\Gamma \vdash u\ C\texttt{<}\overline{{}^sT}\texttt{>} \ \ \mathsf{OK}} \quad \text{WFT\_NVAR}$$

$\boxed{{}^s\Gamma \vdash \overline{{}^sT} \ \ \mathsf{OK}}$ well-formed static types

$$\frac{\overline{{}^s\Gamma \vdash {}^sT_k \ \ \mathsf{OK}}^{k}}{{}^s\Gamma \vdash \overline{{}^sT_k}^{k} \ \ \mathsf{OK}} \quad \text{WFTS\_DEF}$$

$\boxed{{}^s\Gamma \vdash {}^sT \ \ \mathsf{strictly\ OK}}$ strictly well-formed static type

$$\frac{X \in {}^s\Gamma}{{}^s\Gamma \vdash X \ \ \mathsf{strictly\ OK}} \quad \text{SWFT\_VAR}$$

$$\frac{\begin{array}{cc}{}^s\Gamma \vdash \overline{{}^sT} \ \ \mathsf{strictly\ OK} & \{\texttt{self}, \texttt{lost}\} \notin u\ C\texttt{<}\overline{{}^sT}\texttt{>}\\ \mathrm{ClassBnds}\big(u\ C\texttt{<}\overline{{}^sT}\texttt{>}\big) =_o \overline{{}^sN} & {}^s\Gamma \vdash \overline{{}^sT} <:_s \overline{{}^sN}\end{array}}{{}^s\Gamma \vdash u\ C\texttt{<}\overline{{}^sT}\texttt{>} \ \ \mathsf{strictly\ OK}} \quad \text{SWFT\_NVAR}$$

$\boxed{{}^s\Gamma \vdash \overline{{}^sT} \ \ \mathsf{strictly\ OK}}$ strictly well-formed static types

$$\frac{\overline{{}^s\Gamma \vdash {}^sT_k \ \ \mathsf{strictly\ OK}}^{k}}{{}^s\Gamma \vdash \overline{{}^sT_k}^{k} \ \ \mathsf{strictly\ OK}} \quad \text{SWFTS\_DEF}$$

$\boxed{Cls \ \ \mathsf{OK}}$ well-formed class declaration

$$\frac{\begin{array}{l}{}^s\Gamma = \Big\{\overline{X_k \mapsto {}^sN_k}^{k} \ ; \ \texttt{this} \mapsto \texttt{self}\ Cid\texttt{<}\overline{X_k}^{k}\texttt{>}, \_\Big\}\\ \begin{array}{ccc} {}^s\Gamma \vdash \overline{{}^sN_k}^{k} \ \ \mathsf{OK} & & \texttt{self} \notin \overline{{}^sN_k}^{k}\\ {}^s\Gamma \vdash \overline{{}^sT} \ \ \mathsf{strictly\ OK} & \mathrm{ClassBnds}\big(\texttt{self}\ C\texttt{<}\overline{{}^sT}\texttt{>}\big) =_o \overline{{}^sN}' & {}^s\Gamma \vdash \overline{{}^sT} <:_s \overline{{}^sN}'\\ {}^s\Gamma \vdash \overline{fd} \ \ \mathsf{OK} & & {}^s\Gamma, Cid \vdash \overline{md} \ \ \mathsf{OK}\end{array}\end{array}}{\texttt{class}\ Cid\texttt{<}\overline{X_k\ \texttt{extends}\ {}^sN_k}^{k}\texttt{>}\ \texttt{extends}\ C\texttt{<}\overline{{}^sT}\texttt{>}\ \{\ \overline{fd}\ \ \overline{md}\ \} \ \ \mathsf{OK}} \quad \text{WFC\_DEF}$$

$$\frac{}{\texttt{class Object}\ \{\} \ \ \mathsf{OK}} \quad \text{WFC\_OBJECT}$$

$\boxed{{}^s\Gamma \vdash {}^sT\ f;\ \ \mathsf{OK}}$ well-formed field declaration

$$\frac{{}^s\Gamma \vdash {}^sT\ \ \mathsf{OK}}{{}^s\Gamma \vdash {}^sT\ f;\ \ \mathsf{OK}}\quad \text{WFFD\_DEF}$$

$\boxed{{}^s\Gamma \vdash \overline{fd}\ \ \mathsf{OK}}$ well-formed field declarations

$$\frac{\overline{{}^s\Gamma \vdash {}^sT_i\ f_i;\ \ \mathsf{OK}}^{\,i}}{{}^s\Gamma \vdash \overline{{}^sT_i\ f_i;}^{\,i}\ \ \mathsf{OK}}\quad \text{WFFDS\_DEF}$$

$\boxed{{}^s\Gamma, C \vdash md\ \ \mathsf{OK}}$ well-formed method declaration

$$\frac{\begin{array}{l} {}^s\Gamma = \left\{\overline{X'_k \mapsto {}^sN'_k}^{\,k}\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\mathtt{<}\overline{X'_k}^{\,k}\mathtt{>}, \_\right\} \\ {}^s\Gamma' = \left\{\overline{X'_k \mapsto {}^sN'_k}^{\,k}, \overline{X_l \mapsto {}^sN_l}^{\,l}\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\mathtt{<}\overline{X'_k}^{\,k}\mathtt{>}, \overline{pid \mapsto {}^sT_q}^{\,q}\right\} \\ \quad {}^s\Gamma' \vdash \overline{{}^sN_l}^{\,l}, {}^sT, \overline{{}^sT_q}^{\,q}\ \ \mathsf{OK} \qquad\qquad \mathtt{self} \notin \overline{{}^sN_l}^{\,l} \\ \quad {}^s\Gamma' \vdash e : {}^sT \qquad\qquad C\mathtt{<}\overline{X'_k}^{\,k}\mathtt{>} \vdash m\ \ \mathsf{OK} \end{array}}{{}^s\Gamma, C \vdash \_\ \mathtt{<}\overline{X_l\ \mathtt{extends}\ {}^sN_l}^{\,l}\mathtt{>}\ {}^sT\ m(\overline{{}^sT_q\ pid}^{\,q})\ \{\ e\ \}\ \ \mathsf{OK}}\quad \text{WFMD\_DEF}$$

$\boxed{{}^s\Gamma, C \vdash \overline{md}\ \ \mathsf{OK}}$ well-formed method declarations

$$\frac{\overline{{}^s\Gamma, C \vdash md_k\ \ \mathsf{OK}}^{\,k}}{{}^s\Gamma, C \vdash \overline{md_k}^{\,k}\ \ \mathsf{OK}}\quad \text{WFMDS\_DEF}$$

$\boxed{{}^sCT \vdash m\ \ \mathsf{OK}}$ method overriding OK

$$\frac{\forall C'\mathtt{<}\overline{X}'\mathtt{>}.\ \forall \overline{{}^sT}.\ \left(C\mathtt{<}\overline{X}\mathtt{>} \sqsubseteq C'\mathtt{<}\overline{{}^sT}\mathtt{>} \Longrightarrow C\mathtt{<}\overline{X}\mathtt{>}, C'\mathtt{<}\overline{{}^sT},\ \overline{X}'\mathtt{>} \vdash m\ \ \mathsf{OK}\right)}{C\mathtt{<}\overline{X}\mathtt{>} \vdash m\ \ \mathsf{OK}}\quad \text{OVR\_DEF}$$

$\boxed{{}^sCT, C\mathtt{<}\overline{{}^sT},\ \overline{X}\mathtt{>} \vdash m\ \ \mathsf{OK}}$ method overriding OK auxiliary

$$\frac{\begin{array}{c} \mathrm{MSig}(C, m)\ =_o ms \qquad \mathrm{MSig}(C', m)\ =_o ms'_o \\ ms'_o{=_o}None\ \vee\ \left(ms'_o{=_o}ms'\ \wedge\ ms'[\overline{{}^sT}/\overline{X}']{=}ms\right) \end{array}}{C\mathtt{<}\overline{X}\mathtt{>}, C'\mathtt{<}\overline{{}^sT},\ \overline{X}'\mathtt{>} \vdash m\ \ \mathsf{OK}}\quad \text{OVRA\_DEF}$$

$\boxed{{}^s\Gamma\ \ \mathsf{OK}}$ well-formed static environment

$$\frac{\begin{array}{l} {}^s\Gamma = \left\{\overline{X_k \mapsto {}^sN_k}^{\,k}, \overline{X'_l \mapsto {}^sN'_l}^{\,l}\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\mathtt{<}\overline{X_k}^{\,k}\mathtt{>}, \overline{pid \mapsto {}^sT_q}^{\,q}\right\} \\ \quad \mathrm{ClassDom}(C)\ =_o \overline{X_k}^{\,k} \qquad\qquad \mathrm{ClassBnds}(C)\ =_o \overline{{}^sN_k}^{\,k} \\ \quad {}^s\Gamma \vdash \overline{{}^sT_q}^{\,q}, \overline{{}^sN_k}^{\,k}, \overline{{}^sN'_l}^{\,l}\ \ \mathsf{OK} \qquad\qquad \mathtt{self} \notin \overline{{}^sN_k}^{\,k}, \overline{{}^sN'_l}^{\,l} \end{array}}{{}^s\Gamma\ \ \mathsf{OK}}\quad \text{SWFE\_DEF}$$

$\boxed{\vdash P\ \ \mathsf{OK}}$ well-formed program

$$\frac{\begin{array}{l} \overline{Cls_i\ \ \mathsf{OK}}^{\,i} \\ \{\emptyset\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\mathtt{<>}\} \vdash \mathtt{self}\ C\mathtt{<>}\ \ \mathsf{OK} \\ \{\emptyset\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\mathtt{<>}\} \vdash e : \_ \\ \forall C', C''.\ ((C'\mathtt{<\_>} \sqsubseteq C''\mathtt{<\_>}\ \wedge\ C''\mathtt{<\_>} \sqsubseteq C'\mathtt{<\_>}) \Longrightarrow C'{=}C'') \end{array}}{\vdash \overline{Cls_i}^{\,i},\ C,\ e\ \ \mathsf{OK}}\quad \text{WFP\_DEF}$$

$\boxed{{}^s\Gamma \vdash e\ \ \mathsf{enc}}$ encapsulated expression

$$\frac{{}^s\Gamma \vdash \texttt{null} : \_}{{}^s\Gamma \vdash \texttt{null}\ \ \mathsf{enc}} \quad \text{E\_NULL}$$

$$\frac{{}^s\Gamma \vdash x : \_}{{}^s\Gamma \vdash x\ \ \mathsf{enc}} \quad \text{E\_VAR}$$

$$\frac{{}^s\Gamma \vdash \texttt{new}\ {}^sT\texttt{()} : \_}{{}^s\Gamma \vdash \texttt{new}\ {}^sT\texttt{()}\ \ \mathsf{enc}} \quad \text{E\_NEW}$$

$$\frac{\begin{array}{c}{}^s\Gamma \vdash e_0\texttt{.}f : \_ \\ {}^s\Gamma \vdash e_0\ \ \mathsf{enc}\end{array}}{{}^s\Gamma \vdash e_0\texttt{.}f\ \ \mathsf{enc}} \quad \text{E\_READ}$$

$$\frac{\begin{array}{l}{}^s\Gamma \vdash e_0\texttt{.}f \texttt{ = } e_1 : \_ \\ {}^s\Gamma \vdash e_0 : {}^sN_0 \qquad {}^s\Gamma \vdash e_0\ \ \mathsf{enc} \qquad {}^s\Gamma \vdash e_1\ \ \mathsf{enc} \\ \mathrm{om}({}^sN_0) \in \{\texttt{self}, \texttt{peer}, \texttt{rep}\}\end{array}}{{}^s\Gamma \vdash e_0\texttt{.}f \texttt{ = } e_1\ \ \mathsf{enc}} \quad \text{E\_WRITE}$$

$$\frac{\begin{array}{l}{}^s\Gamma \vdash e_0\texttt{.}m\texttt{<}\overline{{}^sT}\texttt{>(}\overline{e}\texttt{)} : \_ \\ \quad {}^s\Gamma \vdash e_0 : {}^sN_0 \qquad\qquad {}^s\Gamma \vdash e_0\ \ \mathsf{enc} \qquad\qquad {}^s\Gamma \vdash \overline{e}\ \ \mathsf{enc} \\ \mathrm{om}({}^sN_0) \in \{\texttt{self}, \texttt{peer}, \texttt{rep}\} \ \vee\ \mathrm{MSig}\big({}^sN_0, m, \overline{{}^sT}\big) =_o \texttt{pure <\_> \_ } m\texttt{(\_)}\end{array}}{{}^s\Gamma \vdash e_0\texttt{.}m\texttt{<}\overline{{}^sT}\texttt{>(}\overline{e}\texttt{)}\ \ \mathsf{enc}} \quad \text{E\_CALL}$$

$$\frac{\begin{array}{l}{}^s\Gamma \vdash \texttt{(}{}^sT\texttt{)}\ e : \_ \\ {}^s\Gamma \vdash e\ \ \mathsf{enc}\end{array}}{{}^s\Gamma \vdash \texttt{(}{}^sT\texttt{)}\ e\ \ \mathsf{enc}} \quad \text{E\_CAST}$$

$\boxed{{}^s\Gamma \vdash \overline{e}\ \ \mathsf{enc}}$ encapsulated expressions

$$\frac{\overline{{}^s\Gamma \vdash e_k\ \ \mathsf{enc}}^{\,k}}{{}^s\Gamma \vdash \overline{e_k}^{\,k}\ \ \mathsf{enc}} \quad \text{EM\_DEF}$$

$\boxed{{}^s\Gamma, C \vdash md\ \ \mathsf{enc}}$ encapsulated method declaration

$$\frac{\begin{array}{l}{}^s\Gamma, C \vdash p\ \texttt{<}\overline{X_l\ \texttt{extends}\ {}^sN_l}^{\,l}\texttt{>}\ {}^sT\ m\texttt{(}\overline{{}^sT_q\ pid}^{\,q}\texttt{)}\ \{\ e\ \}\quad \mathsf{OK} \\ {}^s\Gamma {=} \left\{\overline{X'_k \mapsto {}^sN'_k}^{\,k}\ ;\ \texttt{this} \mapsto \texttt{self}\ C\texttt{<}\overline{X'_k}^{\,k}\texttt{>}, \_\right\} \\ {}^s\Gamma' {=} \left\{\overline{X'_k \mapsto {}^sN'_k}^{\,k}, \overline{X_l \mapsto {}^sN_l}^{\,l}\ ;\ \texttt{this} \mapsto \texttt{self}\ C\texttt{<}\overline{X'_k}^{\,k}\texttt{>}, \overline{pid \mapsto {}^sT_q}^{\,q}\right\} \\ \quad (p{=}\texttt{pure} \Longrightarrow {}^s\Gamma' \vdash e\ \ \mathsf{pure}) \qquad\qquad (p{=}\texttt{impure} \Longrightarrow {}^s\Gamma' \vdash e\ \ \mathsf{enc})\end{array}}{{}^s\Gamma, C \vdash p\ \texttt{<}\overline{X_l\ \texttt{extends}\ {}^sN_l}^{\,l}\texttt{>}\ {}^sT\ m\texttt{(}\overline{{}^sT_q\ pid}^{\,q}\texttt{)}\ \{\ e\ \}\ \ \mathsf{enc}} \quad \text{EMD\_DEF}$$

$\boxed{{}^s\Gamma, C \vdash \overline{md}\ \ \mathsf{enc}}$ encapsulated method declarations

$$\frac{\overline{{}^s\Gamma, C \vdash md_i\ \ \mathsf{enc}}^{\,i}}{{}^s\Gamma, C \vdash \overline{md_i}^{\,i}\ \ \mathsf{enc}} \quad \text{EMDS\_DEF}$$

$\boxed{Cls\ \ \mathsf{enc}}$ encapsulated class declaration

$$\frac{\begin{array}{l}\texttt{class}\ Cid\texttt{<}\overline{X_k\ \texttt{extends}\ {}^sN_k}^{\,k}\texttt{>}\ \texttt{extends}\ C\texttt{<}\overline{{}^sT}\texttt{>}\ \{\ \overline{fd}\ \overline{md}\ \}\quad \mathsf{OK} \\ {}^s\Gamma {=} \left\{\overline{X_k \mapsto {}^sN_k}^{\,k}\ ;\ \texttt{this} \mapsto \texttt{self}\ Cid\texttt{<}\overline{X_k}^{\,k}\texttt{>}, \_\right\} \qquad {}^s\Gamma, Cid \vdash \overline{md}\ \ \mathsf{enc}\end{array}}{\texttt{class}\ Cid\texttt{<}\overline{X_k\ \texttt{extends}\ {}^sN_k}^{\,k}\texttt{>}\ \texttt{extends}\ C\texttt{<}\overline{{}^sT}\texttt{>}\ \{\ \overline{fd}\ \overline{md}\ \}\ \ \mathsf{enc}} \quad \text{EC\_DEF}$$

$$\frac{}{\texttt{class Object}\ \{\}\ \ \mathsf{enc}} \quad \text{EC\_OBJECT}$$

$\boxed{\vdash P\ \ \mathsf{enc}}$ encapsulated program

$$\frac{\vdash \overline{Cls},\ C,\ e\ \ \mathsf{OK} \quad \overline{Cls_k\ \ \mathsf{enc}}^{k} \quad \{\emptyset\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\mathtt{<>}\} \vdash e\ \ \mathsf{enc}}{\vdash \overline{Cls},\ C,\ e\ \ \mathsf{enc}} \quad \text{EP\_DEF}$$

$\boxed{{}^{s}\Gamma \vdash e\ \ \mathsf{pure}}$ pure expression

$\boxed{{}^{s}\Gamma \vdash e\ \ \mathsf{strictly\ pure}}$ strictly pure expression

$$\frac{{}^{s}\Gamma \vdash \mathtt{null} : \_}{{}^{s}\Gamma \vdash \mathtt{null}\ \ \mathsf{strictly\ pure}} \quad \text{SP\_NULL}$$

$$\frac{{}^{s}\Gamma \vdash x : \_}{{}^{s}\Gamma \vdash x\ \ \mathsf{strictly\ pure}} \quad \text{SP\_VAR}$$

$$\frac{{}^{s}\Gamma \vdash \mathtt{new}\ {}^{s}T\mathtt{()} : \_}{{}^{s}\Gamma \vdash \mathtt{new}\ {}^{s}T\mathtt{()}\ \ \mathsf{strictly\ pure}} \quad \text{SP\_NEW}$$

$$\frac{{}^{s}\Gamma \vdash e_0\mathtt{.}f : \_ \quad {}^{s}\Gamma \vdash e_0\ \ \mathsf{strictly\ pure}}{{}^{s}\Gamma \vdash e_0\mathtt{.}f\ \ \mathsf{strictly\ pure}} \quad \text{SP\_READ}$$

$$\frac{\begin{array}{l}{}^{s}\Gamma \vdash e_0\mathtt{.}m\mathtt{<}\overline{{}^{s}T}\mathtt{>(}\overline{e}\mathtt{)} : \_ \\ {}^{s}\Gamma \vdash e_0 : {}^{s}N_0 \quad {}^{s}\Gamma \vdash e_0\ \ \mathsf{strictly\ pure} \quad {}^{s}\Gamma \vdash \overline{e}\ \ \mathsf{strictly\ pure} \\ \mathrm{MSig}\left({}^{s}N_0, m, \overline{{}^{s}T}\right) =_o \mathtt{pure}\ \mathtt{<\_>}\ \_\ m\mathtt{(\_)}\end{array}}{{}^{s}\Gamma \vdash e_0\mathtt{.}m\mathtt{<}\overline{{}^{s}T}\mathtt{>(}\overline{e}\mathtt{)}\ \ \mathsf{strictly\ pure}} \quad \text{SP\_CALL}$$

$$\frac{{}^{s}\Gamma \vdash \mathtt{(}{}^{s}T\mathtt{)}\ e : \_ \quad {}^{s}\Gamma \vdash e\ \ \mathsf{strictly\ pure}}{{}^{s}\Gamma \vdash \mathtt{(}{}^{s}T\mathtt{)}\ e\ \ \mathsf{strictly\ pure}} \quad \text{SP\_CAST}$$

$\boxed{{}^{s}\Gamma \vdash \overline{e}\ \ \mathsf{strictly\ pure}}$ pure expressions

$$\frac{\overline{{}^{s}\Gamma \vdash e_k\ \ \mathsf{strictly\ pure}}^{k}}{{}^{s}\Gamma \vdash \overline{e_k}^{k}\ \ \mathsf{strictly\ pure}} \quad \text{PM\_DEF}$$

$\boxed{{}^{s}\Gamma \vdash {}^{s}T\ \ \mathsf{prg\ OK}}$ reasonable static type

$$\frac{X \in {}^{s}\Gamma}{{}^{s}\Gamma \vdash X\ \ \mathsf{prg\ OK}} \quad \text{PWFT\_VAR}$$

$$\frac{\begin{array}{l}{}^{s}\Gamma \vdash \overline{{}^{s}T}\ \ \mathsf{prg\ OK} \qquad \mathtt{self} \notin u\ C\mathtt{<}\overline{{}^{s}T}\mathtt{>} \\ \mathrm{ClassBnds}(C) =_o \overline{{}^{s}N} \quad u\ C\mathtt{<}\overline{{}^{s}T}\mathtt{>} \rhd \overline{{}^{s}N} =_o \overline{{}^{s}N}' \\ {}^{s}\Gamma, u\ C\mathtt{<}\overline{{}^{s}T}\mathtt{>} \vdash \overline{{}^{s}T} <: \overline{{}^{s}N}, \overline{{}^{s}N}'\end{array}}{{}^{s}\Gamma \vdash u\ C\mathtt{<}\overline{{}^{s}T}\mathtt{>}\ \ \mathsf{prg\ OK}} \quad \text{PWFT\_NVAR}$$

$\boxed{{}^{s}\Gamma, {}^{s}N'' \vdash {}^{s}T <: {}^{s}N, {}^{s}N'}$ reasonable static type argument

$$\frac{\begin{array}{l}{}^{s}\Gamma \vdash {}^{s}T <: {}^{s}N' \\ \mathrm{ClassDom}(C) =_o \overline{X_k}^{k} \qquad \mathrm{ClassBnds}(C) =_o \overline{{}^{s}N_k}^{k} \\ {}^{s}\Gamma' = \left\{\overline{X_k \mapsto {}^{s}N_k}^{k}\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\mathtt{<}\overline{X_k}^{k}\mathtt{>}, \_\right\} \\ \exists\, {}^{s}T_0.\ \left(u\ C\mathtt{<}\overline{{}^{s}T}\mathtt{>} \rhd {}^{s}T_0 =_o {}^{s}T \ \wedge\ {}^{s}\Gamma' \vdash {}^{s}T_0 <: {}^{s}N\right)\end{array}}{{}^{s}\Gamma, u\ C\mathtt{<}\overline{{}^{s}T}\mathtt{>} \vdash {}^{s}T <: {}^{s}N, {}^{s}N'} \quad \text{PWFTA\_DEF}$$

$\boxed{{}^{s}\Gamma, {}^{s}N'' \vdash \overline{{}^{s}T} <: \overline{{}^{s}N}, \overline{{}^{s}N}'}$ reasonable static type arguments

$$\frac{\overline{{}^s\Gamma, {}^sN'' \vdash {}^sT_k \;<:\; {}^sN_k, {}^sN'_k}^{k}}{{}^s\Gamma, {}^sN'' \vdash \overline{{}^sT_k}^{k} \;<:\; \overline{{}^sN_k}^{k}, \overline{{}^sN'_k}^{k}} \quad \text{PWFTAS\_DEF}$$

$\boxed{{}^s\Gamma \vdash \overline{{}^sT} \;\; \mathsf{prg\ OK}}$ reasonable static types

$$\frac{\overline{{}^s\Gamma \vdash {}^sT_k \;\; \mathsf{prg\ OK}}^{k}}{{}^s\Gamma \vdash \overline{{}^sT_k}^{k} \;\; \mathsf{prg\ OK}} \quad \text{PTS\_DEF}$$

$\boxed{\mathit{Cls} \;\; \mathsf{prg\ OK}}$ reasonable class declaration

$$\frac{\begin{array}{l} \mathtt{class}\ \mathit{Cid}\texttt{<}\overline{X_k\ \mathtt{extends}\ {}^sN_k}^{k}\texttt{>}\ \mathtt{extends}\ C\texttt{<}\overline{{}^sT}\texttt{>}\ \{\ \overline{\mathit{fd}}\ \overline{\mathit{md}}\ \}\quad \mathsf{OK} \\ {}^s\Gamma = \left\{ \overline{X_k \mapsto {}^sN_k}^{k}\ ;\ \mathtt{this} \mapsto \mathtt{self}\ \mathit{Cid}\texttt{<}\overline{X_k}^{k}\texttt{>}, \_ \right\} \\ \quad {}^s\Gamma \vdash \overline{{}^sN_k}^{k}\ \ \mathsf{prg\ OK} \qquad\qquad \mathtt{lost} \notin \overline{{}^sN_k}^{k} \\ \quad {}^s\Gamma \vdash \overline{\mathit{fd}}\ \ \mathsf{prg\ OK} \qquad\qquad {}^s\Gamma, \mathit{Cid} \vdash \overline{\mathit{md}}\ \ \mathsf{prg\ OK} \end{array}}{\mathtt{class}\ \mathit{Cid}\texttt{<}\overline{X_k\ \mathtt{extends}\ {}^sN_k}^{k}\texttt{>}\ \mathtt{extends}\ C\texttt{<}\overline{{}^sT}\texttt{>}\ \{\ \overline{\mathit{fd}}\ \overline{\mathit{md}}\ \}\quad \mathsf{prg\ OK}} \quad \text{PC\_DEF}$$

$$\frac{}{\mathtt{class\ Object}\ \{\}\quad \mathsf{prg\ OK}} \quad \text{PC\_OBJECT}$$

$\boxed{{}^s\Gamma \vdash {}^sT\ f;\ \ \mathsf{prg\ OK}}$ reasonable field declaration

$$\frac{{}^s\Gamma \vdash {}^sT\ \ \mathsf{prg\ OK} \qquad \mathtt{lost} \notin {}^sT}{{}^s\Gamma \vdash {}^sT\ f;\ \ \mathsf{prg\ OK}} \quad \text{PFD\_DEF}$$

$\boxed{{}^s\Gamma \vdash \overline{\mathit{fd}}\ \ \mathsf{prg\ OK}}$ reasonable field declarations

$$\frac{\overline{{}^s\Gamma \vdash {}^sT_i\ f_i;\ \ \mathsf{prg\ OK}}^{i}}{{}^s\Gamma \vdash \overline{{}^sT_i\ f_i;}^{i}\ \ \mathsf{prg\ OK}} \quad \text{PFDS\_DEF}$$

$\boxed{{}^s\Gamma, C \vdash \mathit{md}\ \ \mathsf{prg\ OK}}$ reasonable method declaration

$$\frac{\begin{array}{l} {}^s\Gamma, C \vdash p\ \texttt{<}\overline{X_l\ \mathtt{extends}\ {}^sN_l}^{l}\texttt{>}\ {}^sT\ m(\overline{{}^sT_q\ \mathit{pid}}^{q})\ \{\ e\ \}\quad \mathsf{OK} \\ {}^s\Gamma = \left\{ \overline{X'_k \mapsto {}^sN'_k}^{k}\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\texttt{<}\overline{X'_k}^{k}\texttt{>}, \_ \right\} \\ {}^s\Gamma' = \left\{ \overline{X'_k \mapsto {}^sN'_k}^{k}, \overline{X_l \mapsto {}^sN_l}^{l}\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\texttt{<}\overline{X'_k}^{k}\texttt{>}, \overline{\mathit{pid} \mapsto {}^sT_q}^{q} \right\} \\ \ {}^s\Gamma' \vdash \overline{{}^sN_l}^{l}, {}^sT, \overline{{}^sT_q}^{q}\ \ \mathsf{prg\ OK} \qquad \mathtt{lost} \notin \overline{{}^sN_l}^{l}, \overline{{}^sT_q}^{q} \qquad {}^s\Gamma' \vdash e\ \ \mathsf{prg\ OK} \\ p{=}\mathtt{pure} \Longrightarrow \left(\mathrm{free}\left(\overline{{}^sN_l}^{l}, \overline{{}^sT_q}^{q}\right) \subseteq \emptyset\ \wedge\ \mathtt{any} \rhd \overline{{}^sN_l}^{l}, \overline{{}^sT_q}^{q}\ =\ \overline{{}^sN_l}^{l}, \overline{{}^sT_q}^{q} \right) \end{array}}{{}^s\Gamma, C \vdash p\ \texttt{<}\overline{X_l\ \mathtt{extends}\ {}^sN_l}^{l}\texttt{>}\ {}^sT\ m(\overline{{}^sT_q\ \mathit{pid}}^{q})\ \{\ e\ \}\quad \mathsf{prg\ OK}} \quad \text{PMD\_DEF}$$

$\boxed{{}^s\Gamma, C \vdash \overline{\mathit{md}}\ \ \mathsf{prg\ OK}}$ reasonable method declarations

$$\frac{\overline{{}^s\Gamma, C \vdash \mathit{md}_i\ \ \mathsf{prg\ OK}}^{i}}{{}^s\Gamma, C \vdash \overline{\mathit{md}_i}^{i}\ \ \mathsf{prg\ OK}} \quad \text{PMDS\_DEF}$$

$\boxed{\vdash P\ \ \mathsf{prg\ OK}}$ reasonable program

$$\frac{\begin{array}{l} \vdash \overline{\mathit{Cls}_i}^{i},\ C,\ e\ \ \mathsf{OK} \\ \overline{\mathit{Cls}_i\ \ \mathsf{prg\ OK}}^{i} \\ \{\emptyset\ ;\ \mathtt{this} \mapsto \mathtt{self}\ C\texttt{<}\emptyset\texttt{>}\} \vdash e\ \ \mathsf{prg\ OK} \end{array}}{\vdash \overline{\mathit{Cls}_i}^{i},\ C,\ e\ \ \mathsf{prg\ OK}} \quad \text{PP\_DEF}$$

$\boxed{{}^s\Gamma \vdash e \ \ \mathsf{prg\ OK}}$ reasonable expression

$$\frac{{}^s\Gamma \vdash \mathtt{null} : \_}{{}^s\Gamma \vdash \mathtt{null} \ \ \mathsf{prg\ OK}} \quad \text{PE\_NULL}$$

$$\frac{{}^s\Gamma \vdash x : \_}{{}^s\Gamma \vdash x \ \ \mathsf{prg\ OK}} \quad \text{PE\_VAR}$$

$$\frac{{}^s\Gamma \vdash \mathtt{new}\ {}^sT\mathtt{()} : \_}{{}^s\Gamma \vdash \mathtt{new}\ {}^sT\mathtt{()} \ \ \mathsf{prg\ OK}} \quad \text{PE\_NEW}$$

$$\frac{\begin{array}{l}{}^s\Gamma \vdash e_0\mathtt{.}f : \_\\ {}^s\Gamma \vdash e_0 \ \ \mathsf{prg\ OK}\end{array}}{{}^s\Gamma \vdash e_0\mathtt{.}f \ \ \mathsf{prg\ OK}} \quad \text{PE\_READ}$$

$$\frac{\begin{array}{l}{}^s\Gamma \vdash e_0\mathtt{.}f\ \mathtt{=}\ e_1 : \_\\ {}^s\Gamma \vdash e_0 \ \ \mathsf{prg\ OK} \qquad {}^s\Gamma \vdash e_1 \ \ \mathsf{prg\ OK}\end{array}}{{}^s\Gamma \vdash e_0\mathtt{.}f\ \mathtt{=}\ e_1 \ \ \mathsf{prg\ OK}} \quad \text{PE\_WRITE}$$

$$\frac{\begin{array}{l}{}^s\Gamma \vdash e_0\mathtt{.}m\mathtt{<}\overline{{}^sT}\mathtt{>(}\overline{e}\mathtt{)} : \_\\ {}^s\Gamma \vdash e_0 \ \ \mathsf{prg\ OK} \qquad {}^s\Gamma \vdash \overline{e} \ \ \mathsf{prg\ OK}\end{array}}{{}^s\Gamma \vdash e_0\mathtt{.}m\mathtt{<}\overline{{}^sT}\mathtt{>(}\overline{e}\mathtt{)} \ \ \mathsf{prg\ OK}} \quad \text{PE\_CALL}$$

$$\frac{\begin{array}{l}{}^s\Gamma \vdash \mathtt{(}{}^sT\mathtt{)}\ e : \_\\ {}^s\Gamma \vdash e \ \ \mathsf{prg\ OK} \qquad {}^s\Gamma \vdash {}^sT \ \ \mathsf{prg\ OK}\end{array}}{{}^s\Gamma \vdash \mathtt{(}{}^sT\mathtt{)}\ e \ \ \mathsf{prg\ OK}} \quad \text{PE\_CAST}$$

$\boxed{{}^s\Gamma \vdash \overline{e} \ \ \mathsf{prg\ OK}}$ reasonable expressions

$$\frac{\overline{{}^s\Gamma \vdash e_i \ \ \mathsf{prg\ OK}}^{\,i}}{{}^s\Gamma \vdash \overline{e_i}^{\,i} \ \ \mathsf{prg\ OK}} \quad \text{PEM\_DEF}$$

$\boxed{h + o \ = \ (h', \iota)}$ add object o to heap h resulting in heap h' and fresh address ι

$$\frac{\iota \notin \mathrm{dom}(h) \qquad h' {=} h + (\iota \mapsto o)}{h + o \ = \ (h', \iota)} \quad \text{HNEW\_DEF}$$

$\boxed{h[\iota\mathtt{.}f\ \mathtt{=}\ v] \ =_o h'}$ field update in heap

$$\frac{\begin{array}{l}v{=}\mathtt{null}_a \ \vee \ (v{=}\iota' \ \wedge \ \iota' \in \mathrm{dom}(h))\\ h(\iota) =_o \left({}^rT, \overline{fv}\right) \qquad f \in \mathrm{dom}\left(\overline{fv}\right) \qquad \overline{fv}'{=}\overline{fv}[f \mapsto v]\\ h'{=}h + \left(\iota \mapsto \left({}^rT, \overline{fv}'\right)\right)\end{array}}{h[\iota\mathtt{.}f\ \mathtt{=}\ v] \ =_o h'} \quad \text{HUP\_DEF}$$

$\boxed{\mathrm{FType}(h, \iota, f) \ =_o {}^sT_o}$ look up type of field in heap

$$\frac{h \vdash \iota : \ \_\ C\mathtt{<}\_\mathtt{>} \qquad \mathrm{FType}(C, f) \ =_o {}^sT}{\mathrm{FType}(h, \iota, f) \ =_o {}^sT} \quad \text{RFT\_DEF}$$

$\boxed{\mathrm{MSig}(h, \iota, m) \ =_o ms_o}$ look up method signature of method m at ι

$$\frac{h \vdash \iota : \ \_\ C\mathtt{<}\_\mathtt{>} \qquad \mathrm{MSig}(C, m) \ =_o ms}{\mathrm{MSig}(h, \iota, m) \ =_o ms} \quad \text{RMS\_DEF}$$

$\boxed{\mathrm{MBody}(C, m) \ =_o e_o}$ look up most-concrete body of m in class C or a superclass

$$\frac{\texttt{class}\ Cid\texttt{<\_>}\ \texttt{extends}\ \_\texttt{<\_>}\ \{\ \_\ \_\ ms\ \{\ e\ \}\ \_\ \}\in P \quad \text{MName}(ms)=m}{\text{MBody}(Cid, m)\ =_o\ e}\ \text{SMBC\_FOUND}$$

$$\frac{\texttt{class}\ Cid\texttt{<\_>}\ \texttt{extends}\ C_1\texttt{<\_>}\ \{\ \_\ \overline{ms_n\ \{\ e_n\ \}}^{n}\ \}\in P \quad \overline{\text{MName}(ms_n)\neq m}^{n} \quad \text{MBody}(C_1, m)\ =_o\ e}{\text{MBody}(Cid, m)\ =_o\ e}\ \text{SMBC\_INH}$$

$\boxed{\text{MBody}(h, \iota, m)\ =_o\ e_o}$ look up most-concrete body of method m at ι

$$\frac{h(\iota){\downarrow_1}\ =_{o\text{-}}\ C\texttt{<\_>} \quad \text{MBody}(C, m)\ =_o\ e}{\text{MBody}(h, \iota, m)\ =_o\ e}\ \text{RMB\_DEF}$$

$\boxed{\text{sdyn}\left(\overline{{}^sT}, h, \iota, {}^rT, \overline{{}^o\iota}\right)\ =_o\ \overline{{}^rT}}$ simple dynamization of types $\overline{{}^sT}$

$$\frac{\begin{array}{c}{}^o\iota' \in \text{dom}(h)\cup\{\texttt{root}_a\} \qquad {}^o\iota\neq\texttt{any}_a \Longrightarrow {}^o\iota'={}^o\iota \\ \text{ClassDom}(C)\ =_o\ \overline{X} \qquad \texttt{rep}\in\overline{{}^sT} \Longrightarrow \text{owner}(h,\iota)=_o{}^o\iota' \\ \overline{{}^sT}\left[{}^o\iota'/\texttt{peer}, \iota/\texttt{rep}, \texttt{any}_a/\texttt{any}, \overline{{}^rT}/\overline{X},\ \overline{{}^o\iota_i/\texttt{lost}}^{i}\right] =_o \overline{{}^rT}'\end{array}}{\text{sdyn}\left(\overline{{}^sT}, h, \iota, {}^o\iota\ C\texttt{<}\overline{{}^rT}\texttt{>}, \overline{{}^o\iota_i}^{i}\right)\ =_o\ \overline{{}^rT}'}\ \text{SDYN}$$

$\boxed{\text{ClassBnds}(h, \iota, {}^rT, \overline{{}^o\iota})\ =_o\ \overline{{}^rT}}$ upper bounds of type ${}^rT$ from viewpoint ι

$$\frac{\text{ClassBnds}(\text{ClassOf}({}^rT))\ =_o\ \overline{{}^sN} \quad \text{sdyn}\left(\overline{{}^sN}, h, \iota, {}^rT, \overline{{}^o\iota}\right)\ =_o\ \overline{{}^rT}}{\text{ClassBnds}(h, \iota, {}^rT, \overline{{}^o\iota})\ =_o\ \overline{{}^rT}}\ \text{RCB\_DEF}$$

$\boxed{h\vdash {}^rT <:\ {}^rT'}$ type ${}^rT$ is a subtype of ${}^rT'$

$$\frac{C\texttt{<}\overline{X}\texttt{>} \sqsubseteq C'\texttt{<}\overline{{}^sT}\texttt{>} \quad {}^o\iota' \in \{{}^o\iota, \texttt{any}_a\} \quad \text{sdyn}\left(\overline{{}^sT}, h, \_, {}^o\iota\ C\texttt{<}\overline{{}^rT}\texttt{>}, \overline{{}^o\iota}\right)\ =_o\ \overline{{}^rT}'}{h\vdash {}^o\iota\ C\texttt{<}\overline{{}^rT}\texttt{>} <:\ {}^o\iota'\ C'\texttt{<}\overline{{}^rT}'\texttt{>}}\ \text{RT\_DEF}$$

$\boxed{h\vdash \overline{{}^rT} <:\ \overline{{}^rT}'}$ runtime subtypings

$$\frac{\overline{h\vdash {}^rT_i <:\ {}^rT'_i}^{i}}{h\vdash \overline{{}^rT_i}^{i} <:\ \overline{{}^rT'_i}^{i}}\ \text{RTS\_DEF}$$

$\boxed{h\vdash v_o :\ {}^rT}$ runtime type ${}^rT$ assignable to value v_o

$$\frac{h(\iota){\downarrow_1} =_o {}^rT_1 \quad h\vdash {}^rT_1 <:\ {}^rT}{h\vdash \iota :\ {}^rT}\ \text{RTT\_ADDR}$$

$$\frac{}{h\vdash \texttt{null}_a :\ {}^rT}\ \text{RTT\_NULL}$$

$\boxed{h, {}^r\Gamma\vdash v :\ {}^sT}$ static type ${}^sT$ assignable to value v (relative to ${}^r\Gamma$)

$$\frac{\text{dyn}({}^sT, h, {}^r\Gamma, \overline{{}^o\iota})\ =_o\ {}^rT \quad h\vdash v :\ {}^rT \quad {}^sT=\texttt{self}\ \_\texttt{<\_>} \Longrightarrow v={}^r\Gamma(\texttt{this})}{h, {}^r\Gamma\vdash v :\ {}^sT}\ \text{RTSTE\_DEF}$$

$\boxed{h, {}^r\Gamma\vdash \overline{v} :\ \overline{{}^sT}}$ static types $\overline{{}^sT}$ assignable to values $\overline{v}$ (relative to ${}^r\Gamma$)

$$\frac{\overline{h, {}^r\Gamma\vdash v_i :\ {}^sT_i}^{i}}{h, {}^r\Gamma\vdash \overline{v_i}^{i} :\ \overline{{}^sT_i}^{i}}\ \text{RTSTSE\_DEF}$$

$\boxed{h, \iota \vdash v_o : {}^sT}$ static type ${}^sT$ assignable to value v_o (relative to ι)

$$\frac{{}^r\Gamma = \{\emptyset\ ;\ \texttt{this} \mapsto \iota\} \qquad h, {}^r\Gamma \vdash v : {}^sT}{h, \iota \vdash v : {}^sT} \quad \text{RTSTA\_DEF}$$

$\boxed{\mathrm{dyn}({}^sT, h, {}^r\Gamma, \overline{{}^o\iota}) =_o {}^rT_o}$ dynamization of static type (relative to ${}^r\Gamma$)

$$\frac{\begin{array}{ll} {}^r\Gamma = \left\{\overline{X_l \mapsto {}^rT_l}^{\,l}\ ;\ \texttt{this} \mapsto \iota, \_\right\} & h \vdash \iota : {}^o\iota\ C\texttt{<}\overline{{}^rT}\texttt{>} \\ {}^o\iota \in \mathrm{dom}(h) \cup \{\texttt{root}_a\} & \mathrm{ClassDom}(C) =_o \overline{X} \\ \multicolumn{2}{c}{{}^sT\left[{}^o\iota/\texttt{self}, {}^o\iota/\texttt{peer}, \iota/\texttt{rep}, \texttt{any}_a/\texttt{any}, \overline{{}^rT}/\overline{X}, \overline{{}^rT_l/X_l}^{\,l}, \overline{{}^o\iota_i/\texttt{lost}}^{\,i}\right] =_o {}^rT'} \end{array}}{\mathrm{dyn}\left({}^sT, h, {}^r\Gamma, \overline{{}^o\iota_i}^{\,i}\right) =_o {}^rT'} \quad \text{DYNE}$$

$\boxed{\mathrm{dyn}\left(\overline{{}^sT}, h, {}^r\Gamma, \overline{\overline{{}^o\iota}}\right) =_o \overline{{}^rT_o}}$ dynamization of static types (relative to ${}^r\Gamma$)

$$\frac{\mathrm{dyn}({}^sT_1, h, {}^r\Gamma, \overline{{}^o\iota_1}) =_o {}^rT_1 \quad , \quad .. \quad , \quad \mathrm{dyn}({}^sT_n, h, {}^r\Gamma, \overline{{}^o\iota_n}) =_o {}^rT_n}{\mathrm{dyn}({}^sT_1, .., {}^sT_n, h, {}^r\Gamma, \overline{{}^o\iota_1}; ..; \overline{{}^o\iota_n}) =_o {}^rT_1, .., {}^rT_n} \quad \text{DYNSE\_DEF}$$

$\boxed{h, {}^r\Gamma \vdash {}^sN, {}^sT; \left(\overline{{}^sT}/\overline{X}, \iota\right) =_o {}^r\Gamma'}$ validate and create new viewpoint ${}^r\Gamma'$

$$\frac{\begin{array}{lll} {}^s\Gamma \vdash {}^sN\ \ \mathsf{OK} & \mathrm{ClassDom}(\mathrm{ClassOf}({}^sN)) =_o \overline{X} & \mathrm{free}({}^sT) \subseteq \overline{X}, \overline{X_l}^{\,l} \\ \mathrm{dyn}\left(\overline{{}^sT_l}^{\,l}, h, {}^r\Gamma, \emptyset\right) =_o \overline{{}^rT_l}^{\,l} & \multicolumn{2}{l}{{}^r\Gamma' = \left\{\overline{X_l \mapsto {}^rT_l}^{\,l}\ ;\ \texttt{this} \mapsto \iota, \_\right\}} \end{array}}{h, {}^r\Gamma \vdash {}^sN, {}^sT; \left(\overline{{}^sT_l}^{\,l}/\overline{X_l}^{\,l}, \iota\right) =_o {}^r\Gamma'} \quad \text{NVP\_DEF}$$

$\boxed{{}^r\Gamma \vdash h, e \rightsquigarrow h', v}$ big-step operational semantics

$$\frac{}{{}^r\Gamma \vdash h, \texttt{null} \rightsquigarrow h, \texttt{null}_a} \quad \text{OS\_NULL}$$

$$\frac{{}^r\Gamma(x) =_o v}{{}^r\Gamma \vdash h, x \rightsquigarrow h, v} \quad \text{OS\_VAR}$$

$$\frac{\begin{array}{ll} \mathrm{dyn}({}^sT, h, {}^r\Gamma, \emptyset) =_o {}^rT & \mathrm{ClassOf}({}^rT) = C \\ \left(\forall f \in \mathrm{fields}(C)\,.\ \overline{fv}(f) =_o \texttt{null}_a\right) & h + \left({}^rT, \overline{fv}\right) = (h', \iota) \end{array}}{{}^r\Gamma \vdash h, \texttt{new}\ {}^sT\texttt{()} \rightsquigarrow h', \iota} \quad \text{OS\_NEW}$$

$$\frac{\begin{array}{l} {}^r\Gamma \vdash h, e_0 \rightsquigarrow h', \iota_0 \\ h'(\iota_0 . f) =_o v \end{array}}{{}^r\Gamma \vdash h, e_0 . f \rightsquigarrow h', v} \quad \text{OS\_READ}$$

$$\frac{\begin{array}{l} {}^r\Gamma \vdash h, e_0 \rightsquigarrow h_0, \iota_0 \\ {}^r\Gamma \vdash h_0, e_1 \rightsquigarrow h_1, v \\ h_1[\iota_0 . f \texttt{ = } v] =_o h' \end{array}}{{}^r\Gamma \vdash h, e_0 . f \texttt{ = } e_1 \rightsquigarrow h', v} \quad \text{OS\_WRITE}$$

$$\frac{\begin{array}{ll} {}^r\Gamma \vdash h, e_0 \rightsquigarrow h_0, \iota_0 & {}^r\Gamma \vdash h_0, \overline{e_q}^{\,q} \rightsquigarrow h_1, \overline{v_q}^{\,q} \\ \mathrm{MBody}(h_0, \iota_0, m) =_o e & \mathrm{MSig}(h_0, \iota_0, m) =_o \_\ \texttt{<}\overline{X_l\ \texttt{extends}\ \_}^{\,l}\texttt{>}\ \_\ m\texttt{(}\overline{\_\ pid}^{\,q}\texttt{)} \\ \mathrm{dyn}\left(\overline{{}^sT_l}^{\,l}, h, {}^r\Gamma, \emptyset\right) =_o \overline{{}^rT_l}^{\,l} & {}^r\Gamma' = \left\{\overline{X_l \mapsto {}^rT_l}^{\,l}\ ;\ \texttt{this} \mapsto \iota_0, \overline{pid \mapsto v_q}^{\,q}\right\} \\ {}^r\Gamma' \vdash h_1, e \rightsquigarrow h', v & \end{array}}{{}^r\Gamma \vdash h, e_0 . m\texttt{<}\overline{{}^sT_l}^{\,l}\texttt{>(}\overline{e_q}^{\,q}\texttt{)} \rightsquigarrow h', v} \quad \text{OS\_CALL}$$

$$\frac{\begin{array}{c}{}^{r}\Gamma \vdash h, e \rightsquigarrow h', v \\ h', {}^{r}\Gamma \vdash v : {}^{s}T\end{array}}{{}^{r}\Gamma \vdash h, (\,{}^{s}T\,)\; e \rightsquigarrow h', v} \quad \text{OS\_CAST}$$

$\boxed{{}^{r}\Gamma \vdash h, \overline{e} \rightsquigarrow h', \overline{v}}$ sequential big-step operational semantics

$$\frac{\begin{array}{c}{}^{r}\Gamma \vdash h, e \rightsquigarrow h_0, v \\ {}^{r}\Gamma \vdash h_0, \overline{e_i}^{\,i} \rightsquigarrow h', \overline{v_i}^{\,i}\end{array}}{{}^{r}\Gamma \vdash h, e, \overline{e_i}^{\,i} \rightsquigarrow h', v, \overline{v_i}^{\,i}} \quad \text{OSS\_DEF}$$

$$\frac{}{{}^{r}\Gamma \vdash h, \emptyset \rightsquigarrow h, \emptyset} \quad \text{OSS\_EMPTY}$$

$\boxed{\vdash P \rightsquigarrow h, v}$ big-step operational semantics of a program

$$\frac{\begin{array}{l}\forall f \in \text{fields}(C)\,.\; \overline{fv}(f) =_o \texttt{null}_a \\ \emptyset + \left(\texttt{root}_a\; C\texttt{<>}, \overline{fv}\right) \;=\; (h_0, \iota_0) \\ {}^{r}\Gamma_0 = \{\emptyset\; ;\; \texttt{this} \mapsto \iota_0\} \qquad {}^{r}\Gamma_0 \vdash h_0, e \rightsquigarrow h, v\end{array}}{\vdash \overline{Cls},\; C,\; e \rightsquigarrow h, v} \quad \text{OSP\_DEF}$$

$\boxed{h,\; \iota \vdash {}^{r}T_o \;\; \mathsf{strictly\ OK}}$ strictly well-formed runtime type ${}^{r}T_o$

$$\frac{\begin{array}{ll}{}^{o}\iota \in \text{dom}(h) \cup \{\texttt{any}_a, \texttt{root}_a\} & \text{ClassBnds}\left(h, \iota, {}^{o}\iota\; C\texttt{<}\overline{{}^{r}T_k}^{\,k}\texttt{>}, \emptyset\right) =_o \overline{{}^{r}T'_k}^{\,k} \\ h,\; \_ \vdash \overline{{}^{r}T_k}^{\,k} \;\; \mathsf{strictly\ OK} & h \vdash \overline{{}^{r}T_k}^{\,k} <: \overline{{}^{r}T'_k}^{\,k}\end{array}}{h,\; \iota \vdash {}^{o}\iota\; C\texttt{<}\overline{{}^{r}T_k}^{\,k}\texttt{>} \;\; \mathsf{strictly\ OK}} \quad \text{SWFRT\_DEF}$$

$\boxed{h,\; \iota \vdash \overline{{}^{r}T} \;\; \mathsf{strictly\ OK}}$ strictly well-formed runtime types

$$\frac{\overline{h,\; \iota \vdash {}^{r}T_i \;\; \mathsf{strictly\ OK}}^{\,i}}{h,\; \iota \vdash \overline{{}^{r}T_i}^{\,i} \;\; \mathsf{strictly\ OK}} \quad \text{SWFRTSO\_DEF}$$

$\boxed{h,\; \overline{\iota} \vdash \overline{{}^{r}T} \;\; \mathsf{strictly\ OK}}$ strictly well-formed runtime types

$$\frac{\overline{h,\; \iota_i \vdash {}^{r}T_i \;\; \mathsf{strictly\ OK}}^{\,i}}{h,\; \overline{\iota_i}^{\,i} \vdash \overline{{}^{r}T_i}^{\,i} \;\; \mathsf{strictly\ OK}} \quad \text{SWFRTS\_DEF}$$

$\boxed{h \vdash \iota \;\; \mathsf{OK}}$ well-formed object at an address

$$\frac{\begin{array}{l}h(\iota)\downarrow_1 =_{o\text{-}} C\texttt{<\_>} \qquad h,\; \iota \vdash h(\iota)\downarrow_1 \;\; \mathsf{strictly\ OK} \qquad \texttt{root}_a \in \text{owners}(h, \iota) \\ \forall f \in \text{fields}(C)\,.\; \exists\, {}^{s}T.\; (\text{FType}(h, \iota, f) =_o {}^{s}T \;\wedge\; h, \iota \vdash h(\iota.f) : {}^{s}T)\end{array}}{h \vdash \iota \;\; \mathsf{OK}} \quad \text{WFA\_DEF}$$

$\boxed{h \;\; \mathsf{OK}}$ well-formed heap

$$\frac{\forall \iota \in \text{dom}(h)\,.\; h \vdash \iota \;\; \mathsf{OK}}{h \;\; \mathsf{OK}} \quad \text{WFH\_DEF}$$

$\boxed{h, {}^{r}\Gamma : {}^{s}\Gamma \;\; \mathsf{OK}}$ runtime and static environments correspond

$$\frac{\begin{array}{l}{}^{r}\Gamma = \left\{\overline{X_l \mapsto {}^{r}T_l}^{\,l}\; ;\; \texttt{this} \mapsto \iota,\; \overline{pid \mapsto v_q}^{\,q}\right\} \\ {}^{s}\Gamma = \left\{\overline{X_l \mapsto {}^{s}N_l}^{\,l}, \overline{X'_k \mapsto \_}^{\,k}\; ;\; \texttt{this} \mapsto \texttt{self}\; C\texttt{<}\overline{X'_k}^{\,k}\texttt{>},\; \overline{pid \mapsto {}^{s}T_q}^{\,q}\right\} \\ \quad h \;\; \mathsf{OK} \qquad {}^{s}\Gamma \;\; \mathsf{OK} \qquad h,\; \iota \vdash \overline{{}^{r}T_l}^{\,l} \;\; \mathsf{strictly\ OK} \\ \quad \text{dyn}\left(\overline{{}^{s}N_l}^{\,l}, h, {}^{r}\Gamma, \emptyset\right) =_o \overline{{}^{r}T'_l}^{\,l} \qquad h \vdash \overline{{}^{r}T_l}^{\,l} <: \overline{{}^{r}T'_l}^{\,l} \\ \quad h, {}^{r}\Gamma \vdash \iota : \texttt{self}\; C\texttt{<}\overline{X'_k}^{\,k}\texttt{>} \qquad h, {}^{r}\Gamma \vdash \overline{v_q}^{\,q} : \overline{{}^{s}T_q}^{\,q}\end{array}}{h, {}^{r}\Gamma : {}^{s}\Gamma \;\; \mathsf{OK}} \quad \text{WFRSE\_DEF}$$

```
Definition rules:        140 good    0 bad
Definition rule clauses: 361 good    0 bad
```