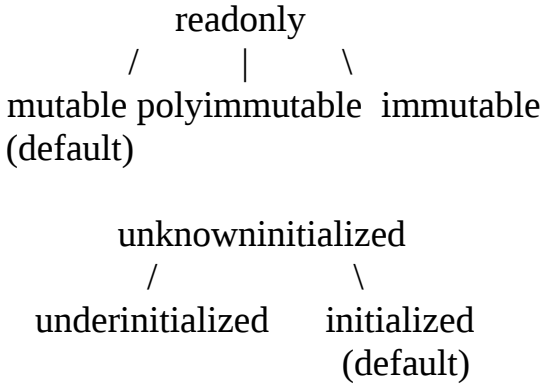
cd ::= class C extends D { $\overline{fd}$; kd $\overline{md}$ }	class
fd ::= t C f	field
kd ::= q C (t C g, t C f) { super(g); this.f = f; };	constructor
md ::= t C m (t C this, t C x) { $\overline{t\ C\ y}$ s; return z }	instance method
e ::= x | x.f	expression
s ::= x = e | x.f = y | x = y.m(z) | super(g) | x = new t() | s;s	statement
t ::= k q	qualifier type
k ::= initiazlized | underinitialized | unknowninitialized	initializatioin qualifier
q ::= readonly | polyimmutable | mutable | immutable	immutability qualifier

Each class has only one constructor. But it doesn’t affect the generality.

Type Hierarchy



	Mutable	Immutable	PolyImmutable	Readonly
UnderInitialized	√	√	√	X
Initialized	√	X	X	X
UnknownInitializaed	√	X	X	X

√ means allowing assigning fields

Figure 1 Combination of qulifiers. Two qualifier hierarchies are orthogonal. If an object is under initialization, its immutability guarantee is not satisfied. So even immutable and polyimmutable objects can also be modifed when under initialization. We don’t have readonly objects, so there is no need to initialize readonly objects. Therefore, readonly doesn’t have such exception when under initialization.

Subtype relations:
$k_1\ q_1 <: k_2\ q_2 <=> k_1 <: k_2 \wedge q_1 <: q_2$

Helper Functions

$$\frac{q\ C\ f}{fType(f) = q}$$ *Note:* No initialization modifier on field declarations

cBody(c) returns constructor body of c.
mBody(m) returns method body of m.

Viewpoint Adaptation Rules

_ ▷ mutable = mutable
_ ▷ readonly = readonly
_ ▷ immutable = immutable
q ▷ polyimmutable = q

Special Rules

- Forbid mutable and readonly on fields
- Forbid readonly constructor return type
- In constructor, $q_{this} = q_{ret}$
- Does not allow initialization modifier on fields and constructor return types

Typing Rules

$$\frac{x \in \Gamma}{\Gamma \vdash x : \Gamma(x)}\ (\text{T-VAR})$$

$$\frac{\Gamma(x) = k_x\ q_x \quad fType(f) = q_f \quad q = q_x \triangleright q_f \quad k = \begin{cases} initialized & \textit{if } k_x = \textit{initialized} \\ \textit{unknown}initialized & \textit{otherwise} \end{cases}}{\Gamma \vdash x.f : k\ q}\ (\text{T-FLD})$$

Figure 2 Expression typing

$$\frac{\Gamma \vdash e = t_e \quad t_e <: \Gamma(x)}{\Gamma \vdash x = e} \text{ (T-VARASS)}$$

$$\frac{\begin{array}{l}\Gamma(x) = k_x\, q_x \quad \Gamma(y) = k_y\, q_y \quad \text{typeof}(f) = q_f \\ q_x = \text{mutable} \\ \quad \vee (k_x = \text{underinitialized} \wedge q_x = \text{immutable}) \text{ // Didn't restrict x = "this"} \\ \quad \vee (k_x = \text{underinitialized} \wedge q_x = \text{polyimmutable}) \text{ // Didn't restrict x = "this"} \\ q_y <: q_x \triangleright q_f \\ k_x = \text{underinitialized} \vee k_y = \text{initialized}\end{array}}{\Gamma \vdash x.f = y} \text{ (T-FLDASS)}$$

$$\frac{\begin{array}{l}\Gamma(x) = k_x\, q_x \quad \Gamma(y) = k_y\, q_y \quad \Gamma(\bar{z}) = \bar{k}_z\, \bar{q}_z \quad \text{typeof}(m) = k_{this}\, q_{this}, \bar{k}_p\, \bar{q}_p \rightarrow k_{ret}\, q_{ret} \\ k_y <: k_{this} \quad \bar{k}_z <: \bar{k}_p \quad k_{ret} <: k_x \\ q_y <: q_x \triangleright q_{this} \quad \bar{q}_z <: q_x \triangleright \bar{q}_p \quad q_x \triangleright q_{ret} <: q_x\end{array}}{\Gamma \vdash x = y.m(\bar{z})} \text{ (T-CALL)}$$

$$\frac{\begin{array}{l}c \text{ in } C \quad C <: D \quad \text{typeof}(D) = \bar{k}_{p\text{-}D}\, \bar{q}_{p\text{-}D} \rightarrow q_{ret\text{-}D} \quad \text{typeof}(C) = \overline{\_\ \_} \rightarrow q_{ret\text{-}C} \\ \Gamma(z) = k_z\, q_z \quad q_{ret\text{-}C} <: q_{ret\text{-}C} \triangleright q_{ret\text{-}D} \quad \bar{q}_z <: q_{ret\text{-}C} \triangleright \bar{q}_{p\text{-}D} \quad \bar{k}_z <: \bar{k}_{p\text{-}D}\end{array}}{\Gamma \vdash \text{super}(\bar{z}) \text{ in } c} \text{ (T-SUPER)}$$

$$\frac{\Gamma(x) = k_x\ q_x \quad \Gamma(\bar{y}) = \bar{k}_y\ \bar{q}_y \quad \text{typeof}(C) = \bar{k}_p\ \bar{q}_p \rightarrow q_{ret} \quad \bar{q}_y <: q \triangleright \bar{q}_p \quad q <: q \triangleright q_{ret} \quad q = \text{mutable} \vee q = \text{immutable} \quad k_y <: k_p \quad q <: q_x \quad k <: k_x \quad k = \begin{cases} \text{initialized} & \textit{if } \bar{k}_p = \textit{initialized} \\ \text{underinitialized} & \textit{otherwise} \end{cases}}{\Gamma \vdash x = \text{new } q\ C(\bar{y})} \text{ (T-NEW)}$$

$$\frac{\Gamma \vdash s_1 \quad \Gamma \vdash s_2}{\Gamma \vdash s_1; s_2} \text{ (T-SEQ)}$$

Figure 3 Statement typing

Well-formdness Rules

$$\frac{}{\vdash_{\text{object}} \_ \text{ is OK}} \text{ (WF-CONS-OBJECT)}$$

$$\frac{\text{cBody}(c) = \text{super}(g);s \quad \text{typeof}(c) = \bar{k}_p\ \bar{q}_p \rightarrow q_{ret} \quad \Gamma = (\text{this} : \text{underinitialized } q_{ret},\ \bar{p} : \bar{k}_p\ \bar{q}_p,\ y : \bar{k}_{local}\ \bar{q}_{local}) \quad \Gamma \vdash \text{super}(\bar{y}) \text{ in } c \quad \Gamma \vdash s \quad \bar{q}_p = \begin{cases} \text{polyimmutable or immutable} & \textit{if } q_{ret} = \textit{polyimmutable or } q_{ret} = \textit{immutable} \\ \_ & \textit{otherwise} \end{cases}}{\vdash_C c \text{ is OK}} \text{ (WF-CONS)}$$

Note: $\vdash_C c$ reads “constructor c in class C is well-formed”.

Only allowing polyimmutable and immutable constructor parameter types in polyimmutable and immutable constructor allows readonly field to be safe, i.e., no aliased mutable objects will be captured by readonly fields of an immutable object and break the immutability contract.

$$\frac{\text{mBody}(m) = s;\text{return } z \quad \text{typeof}(m) = k_{this}\ q_{this},\ \overline{k}_p\ \overline{q}_p \rightarrow t_{ret} \qquad \Gamma = (\text{this} : k_{this}\ q_{this},\ \overline{p} : \overline{k}_p\ \overline{q}_p,\ \overline{y} : \overline{k}_{local}\ \overline{q}_{local}) \quad \Gamma \vdash s \quad \Gamma(z) <: t_{ret}}{\vdash m \text{ is OK}} \text{ (WF-METH)}$$

Figure 4 Well-formdness typing