cd ::= class C extends D { $\overline{\text{fd}}$; kd $\overline{\text{md}}$ }	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) { $\overline{\text{t C y}}$ 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 ::= initilized | underinitialized | anyinitialized	initializatioin qualifier
q ::= readonly | polyimmutable | mutable | immutable	immutability qualifier

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

Type Hierarchy

```
                readonly
           /        |        \
mutable polyimmutable  immutable
(default)
```

```
          anyinitialized
           /              \
underinitialized     initialized
                      (default)
```

	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

$$\frac{q\ C\ f}{fType(f) = q}$$ 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

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

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

Figure 2 Expression typing

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

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

$$\frac{\begin{array}{llll} \Gamma(x) = k_x\ q_x & \Gamma(y) = k_y\ q_y & \Gamma(\overline{z}) = \overline{k}_z\, \overline{q}_z & \text{typeof}(m) = k_{this}\ q_{this},\ \overline{k}_p\ \overline{q}_p \to k_{ret}\ q_{ret} \\ k_y <: k_{this} & \overline{k}_z <: \overline{k}_p & k_{ret} <: k_x & \\ q_y <: q_x \rhd q_{this} & \overline{q}_z <: q_x \rhd \overline{q}_p & q_x \rhd q_{ret} <: q_x & \end{array}}{\Gamma \vdash x = y.m(\overline{z})} \qquad (\text{ T-CALL })$$

$$\frac{\begin{array}{llll} c \text{ in } C & C <: D & \text{typeof}(D) = \overline{k}_{p\text{-}D}\ \overline{q}_{p\text{-}D} \to q_{ret\text{-}D} & \text{typeof}(C) = \overline{\_\ \_} \to q_{ret\text{-}C} \\ \Gamma(z) = k_z\ q_z & q_{ret\text{-}C} <: q_{ret\text{-}C} \rhd q_{ret\text{-}D} & \overline{q}_z <: q_{ret\text{-}C} \rhd \overline{q}_{p\text{-}D} & \overline{k}_z <: \overline{k}_{p\text{-}D} \end{array}}{\Gamma \vdash \text{super}(\overline{z}) \text{ in } c} \qquad (\text{ T-SUPER})$$

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

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

Figure 3 Statement typing

Well-formdness Rules

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

$$
\frac{
\begin{array}{l}
\mathit{cBody}(c) = \mathit{super}(\underline{g});s \qquad \mathit{typeof}(c) = \overline{k}_p\ \overline{q}_p \rightarrow q_{ret} \\
\Gamma = (\mathit{this} : \mathit{underinitialized}\ q_{ret},\ \overline{p} : \overline{k}_p\ \overline{q}_p,\ \overline{y} : \overline{k}_{local}\ \overline{q}_{local}) \quad \Gamma \vdash \mathit{super}(\overline{y}) \text{ in } c \quad \Gamma \vdash s
\end{array}
}{
\vdash c \text{ is OK}
} \quad \text{(WF-CONS)}
$$

Note: $\vdash$ c reads “constructor c is well-formed”.

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

Figure 4 Well-formdness typing