

## Constraint Based Typing

$\Gamma \vdash e:T \mid \{C\}$

It means “expression  $e$  has type  $T$  under assumptions whenever the constraints  $C$  are satisfied”.

$\Gamma \vdash s \mid \{C\}$

It means “statement  $s$  typechecks under assumptions whenever the constraints  $C$  are satisfied”.

### Note:

- 1) PICOInfer assumes the input program is a well-typed in Freedom-Before-Commitment(FBC) type system and initialization qualifiers are given already.
- 2) PICOInfer assumes that input program is well-formed in terms of assignability on fields(meaning one and only one assignability qualifier is used on a field; no @RDA is used on static fields) and assignability qualifiers are given already.
- 3) Therefore, PICOInfer **only infers solution in mutability hierarchy**, not the ~~initialization hierarchy or assignability dimension~~.
- 4) In PICOInfer formalization, we only write initialization and assignability qualifiers in assumptions that affect how we generate mutability constraints, but not the in the conclusions(because it doesn't contribute to constraint generation and those two dimensions are assumed to be valid already).

## Typing Rules (Constraint Based)

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

\* **Note** :  $\Gamma_q$  is the type environment that only stores mutability qualifiers of variables.

$$\frac{\Gamma_q(x) = q_x \quad \text{fType}(f) = q_f}{\Gamma_q \vdash x.f : q \mid \{q = q_x \triangleright q_f\}} \quad (\text{T-FLD})$$

**Figure 2 Expression typing**

$$\Gamma_q \vdash e = q_e$$

( T-VARASS )

---


$$\Gamma_q \vdash x = e \mid \{q_e <: \Gamma_q(x)\}$$

$$\Gamma(x) = k_x q_x \quad \Gamma_q(y) = q_y \quad \text{typeof}(f) = q_f a_f$$

( T-FLDASS )

---


$$\Gamma_q \vdash x.f = y \mid \{q_y <: q_x \triangleright q_f, \text{fieldWrite}(k_x, q_x, a_f)\}$$

fieldWrite( $k_x, q_x, a_f$ ) :: =

if  $a_f = \text{assignable} \Rightarrow q_x \neq \text{readonly} \vee q_f \neq \text{receiverdependantmutable}$

else if  $k_x = \text{underinitialized} \Rightarrow q_x = \text{mutable} \vee q_x = \text{immutable} \vee q_x = \text{receiverdependantmutable}$

else  $\Rightarrow q_x = \text{mutable}$

*Red  $\vee$  means CFI currently doesn't support this constraint(disjunction)*

$$\Gamma_q(x) = q_x \quad \Gamma_q(y) = q_y \quad \Gamma_q(\bar{z}) = \bar{q}_z \quad \text{typeof}(m) = q_{\text{this}}, \bar{q}_p \rightarrow q_{\text{ret}}$$

---


$$\Gamma_q \vdash x = y.m(\bar{z}) \mid \{q_{\text{this-vp}} = q_y \triangleright q_{\text{this}}, \bar{q}_{p\text{-vp}} = q_y \triangleright \bar{q}_p, q_{\text{ret-vp}} = q_y \triangleright q_{\text{ret}}, q_y <: q_{\text{this-vp}}, \bar{q}_z <: \bar{q}_{p\text{-vp}}, q_{\text{ret-vp}} <: q_x\}$$

( T-CALL )

**TODO** : inference of polymutable methods will be implemented later.

$$\begin{array}{l} kd \text{ in } C \quad C <: D \quad \text{typeof}(D) = \bar{q}_{p\text{-D}} \rightarrow q_{\text{ret-D}} \quad \text{typeof}(kd) = \bar{\_} \rightarrow q_{\text{ret-C}} \\ \Gamma_q(z) = q_z \end{array}$$

---


$$\Gamma_q \vdash \text{super}(\bar{z}) \text{ in } kd \mid \{q_{\text{ret-C}} <: q_{\text{ret-C}} \triangleright q_{\text{ret-D}}, \bar{q}_z <: q_{\text{ret-C}} \triangleright \bar{q}_{p\text{-D}}\}$$

( T-SUPER )

( T-THIS ) (omitted)

\* *Note:* As the type rule in the typechecking side, the constraint based type rule T-THIS is very much the same as T-SUPER except that the constructor invoked by “this(..., ...)” comes from the same class.

$$\begin{array}{c}
 \Gamma_q(x) = q_x \quad \Gamma_q(\bar{y}) = \bar{q}_y \quad \text{typeof}(C) = \bar{q}_p \rightarrow q_{\text{ret}} \\
 \hline
 \Gamma_q \vdash x = \text{new } q \ C(\bar{y}) \mid \{ \bar{q}_y <: q \triangleright \bar{q}_p, q <: q \triangleright q_{\text{ret}}, q <: q_x, q \neq \text{readonly} \} \\
 \text{(T-NEW)}
 \end{array}$$
  

$$\begin{array}{c}
 \Gamma_q \vdash s_1 \mid \{C1\} \quad \Gamma_q \vdash s_2 \mid \{C2\} \\
 \hline
 \Gamma_q \vdash s_1; s_2 \mid \{C1, C2\} \\
 \text{(T-SEQ)}
 \end{array}$$

**Figure 3 Statement typing**

### Well-formdness Rules

$$\begin{array}{c}
 \text{fType}(\text{fd}) = q \quad C <: D \quad \text{fd} \notin \text{fields}(D) \\
 \hline
 \vdash_c \text{fd is OK} \mid \{q \neq \text{polymutable}\} \\
 \text{(WF-FLD)}
 \end{array}$$
  

$$\begin{array}{c}
 \hline
 \vdash_{\text{object}} \text{kd is OK} \mid \{\} \\
 \text{(WF-CONS-OBJECT)}
 \end{array}$$

$$\begin{array}{l}
\text{cBody}(\underline{\text{kd}}) = \text{super}(\underline{\text{g}}); \text{this.f} = \underline{\text{f}} \quad \text{typeof}(\underline{\text{kd}}) = \underline{\text{q}}_{\text{p}} \rightarrow \underline{\text{q}}_{\text{ret}} \\
\Gamma = (\text{this} : \text{underinitialized } \underline{\text{q}}_{\text{ret}}, \underline{\text{p}} : \underline{\text{q}}_{\text{p}}, \underline{\text{y}} : \underline{\text{q}}_{\text{local}}) \\
\Gamma_{\text{q}} \vdash \text{super}(\underline{\text{y}}) \text{ in } \underline{\text{kd}} \mid \{\text{C1}\} \quad \Gamma_{\text{q}} \vdash \text{this.f} = \underline{\text{f}} \mid \{\text{C2}\}
\end{array}$$

(WF-CONS)

---


$$\vdash_{\text{C}} \underline{\text{kd}} \text{ is OK} \mid \{\underline{\text{q}}_{\text{ret}} \neq \text{readonly}, \underline{\text{q}}_{\text{ret}} \neq \text{polymutable}, \text{C1}, \text{C2}\}$$

*Note:*  $\vdash_{\text{C}} \underline{\text{kd}}$  reads “constructor  $\underline{\text{kd}}$  in class  $\text{C}$  is well-formed”.

$$\begin{array}{l}
\text{mBody}(\underline{\text{md}}) = \underline{\text{s}}; \text{return } \underline{\text{z}} \quad \text{typeof}(\underline{\text{md}}) = \underline{\text{q}}_{\text{this}}, \underline{\text{q}}_{\text{p}} \rightarrow \underline{\text{q}}_{\text{ret}} \\
\Gamma_{\text{q}} = (\text{this} : \underline{\text{q}}_{\text{this}}, \underline{\text{p}} : \underline{\text{q}}_{\text{p}}, \underline{\text{y}} : \underline{\text{q}}_{\text{local}}) \quad \Gamma_{\text{q}} \vdash \underline{\text{s}} \mid \{\text{C1}\} \quad \Gamma_{\text{q}}(\underline{\text{z}}) <: \underline{\text{q}}_{\text{ret}} \mid \{\text{C2}\} \\
\text{Standard method overriding rule holds} \mid \{\text{C3}\}
\end{array}$$

(WF-METH)

---


$$\vdash_{\text{C}} \underline{\text{md}} \text{ is OK} \mid \{\text{C1}, \text{C2}, \text{C3}\}$$

$$\vdash_{\text{C}} \underline{\text{fd}} \text{ is OK} \mid \{\text{C1}\} \quad \vdash_{\text{C}} \underline{\text{kd}} \text{ is OK} \mid \{\text{C2}\} \quad \vdash_{\text{C}} \underline{\text{md}} \text{ is OK} \mid \{\text{C3}\}$$

(WF-CLASS)

---


$$\vdash \text{C is OK} \mid \{\text{C1}, \text{C2}, \text{C3}\}$$

**Figure 4 Well-formdness typing**

## Extension to real Java

In real Java, there are static fields, static methods, initialization blocks.

## Helper Method

usedQualifiers( $\bar{s}$ ) returns all mutability qualifiers used in  $\bar{s}$  recursively

$cd ::= \text{class } C \text{ extends } D \{ \bar{sfd} \bar{fd}; \bar{sib} \bar{ib} \bar{kd} \bar{smd} \bar{md} \}$	class
$sfd ::= \text{static } q \ C \ sf$	static field
$smd ::= \text{static } t \ C \ sm \ ( \ t \ C \ x \ ) \{ \bar{t} \ C \ \bar{y} \ \bar{s}; \text{return } z; \}$	static method
$sib ::= \text{static}\{\bar{s};\}$	static initialization block
$ib ::= \{\bar{s};\}$	initialization block

$$fType(sfd) = q$$

(WF-STATIC-FLD)

---


$$\vdash sfd \text{ is OK} \mid \{q \neq \text{polymutable}, q \neq \text{receiverdependantmutable}\}$$

$$\begin{array}{lll} mBody(smd) = \bar{s}; \text{return } z & \text{typeof}(smd) = \bar{q}_p \rightarrow q_{ret} & \\ \Gamma_q = (\bar{p} : \bar{q}_p, \bar{y} : \bar{q}_{local}) & \Gamma_q \vdash \bar{s} \mid \{C1\} & \Gamma_q(z) <: q_{ret} \mid \{C2\} \end{array}$$


---


$$\vdash smd \text{ is OK} \mid \{C1, C2, \bar{q}_p \neq \text{receiverdependantmutable}, q_{ret} \neq \text{receiverdependantmutable}, \text{foreach } q \text{ in usedQualifiers}(\bar{s}; \text{return } z) : q \neq \text{receiverdependantmutable}\}$$

(WF-STATIC-METH)

$$\Gamma_q \vdash \bar{s} \mid \{C\}$$

(WF-STATIC-BLK)

---


$$\vdash sib \text{ is OK} \mid \{C, \text{foreach } q \text{ in usedQualifiers}(\bar{s}) : q \neq \text{receiverdependantmutable}\}$$

$$\Gamma_q \vdash \bar{s} \mid \{C\}$$

(WF-BLK)

---


$$\vdash_c ib \text{ is OK} \mid \{C\}$$

$$\begin{array}{l} \vdash \bar{sfd} \text{ is OK} \mid \{C1\} \quad \vdash_c \bar{fd} \text{ is OK} \mid \{C2\} \quad \vdash_c \bar{kd} \text{ is OK} \mid \{C3\} \quad \vdash \bar{smd} \text{ is OK} \mid \{C4\} \\ \vdash_c \bar{md} \text{ is OK} \mid \{C5\} \quad \vdash \bar{sib} \text{ is OK} \mid \{C6\} \quad \vdash_c \bar{ib} \text{ is OK} \mid \{C7\} \end{array}$$

(WF-CLASS)

---


$$\vdash C \text{ is OK} \mid \{C1, C2, C3, C4, C5, C6, C7\}$$