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-Committment(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 assigability qualifiers in assumptions that affect how we generate mutability constraints, but not the in the conclusions(because it doesn’t contribute to contraint 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** : Γ_q is the type environment that only stores mutability qualifiers of variables.

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

Figure 2 Expression typing

$$\frac{\Gamma_q \vdash e = q_e}{\Gamma_q \vdash x = e \mid \{q_e <: \Gamma_q(x)\}} \quad \text{(T-VARASS)}$$

$$\frac{\Gamma(x) = k_x\, q_x \quad \Gamma_q(y) = q_y \quad \text{typeof}(f) = q_f\, a_f}{\Gamma_q \vdash x.f = y \mid \{q_y <: q_x \triangleright q_f,\ \text{fieldWrite}(k_x, q_x, a_f)\}} \quad \text{(T-FLDASS)}$$

fieldWrite(k_x , q_x , a_f) :: =
if a_f = assignable => $q_x \neq$ *readonly* ∨ $q_f \neq$ *receiverdependantmutable*
else if k_x = underinitialized => q_x = mutable ∨ q_x = immutable ∨ q_x = receiverdependantmutable
else => q_x = mutable

Red ∨ means CFI currently doesn't support this constraint(disjunction)

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

TODO : inference of polymutable methods will be implemented later.

$$\frac{kd \text{ in } C \quad C <: D \quad \text{typeof}(D) = \bar{q}_{p\text{-}D} \rightarrow q_{ret\text{-}D} \quad \text{typeof}(kd) = \overline{\_\ \_} \rightarrow q_{ret\text{-}C} \quad \Gamma_q(z) = q_z}{\Gamma_q \vdash \text{super}(\bar{z}) \text{ in } kd \mid \{q_{ret\text{-}C} <: q_{ret\text{-}C} \triangleright q_{ret\text{-}D},\ \bar{q}_z <: q_{ret\text{-}C} \triangleright \bar{q}_{p\text{-}D}\}} \quad \text{(T-SUPER)}$$

(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.

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

$$\frac{\Gamma_q \vdash s_1 \mid \{C1\} \quad \Gamma_q \vdash s_2 \mid \{C2\}}{\Gamma_q \vdash s_1; s_2 \mid \{C1, C2\}} \quad \text{(T-SEQ)}$$

Figure 3 Statement typing

Well-formdness Rules

$$\frac{\text{fType}(fd) = q \qquad C <: D \qquad fd \notin \text{fields}(D)}{\vdash_C fd \text{ is OK} \mid \{q \neq \text{polymutable}\}} \quad \text{(WF-FLD)}$$

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

$$\frac{\text{cBody}(kd) = \text{super}(g);\ \text{this.f} = f \quad \text{typeof}(kd) = \overline{q_p} \rightarrow q_{ret} \quad \Gamma = (\text{this} : \text{underinitialized}\ q_{ret},\ \overline{p} : \overline{q_p},\ \overline{y} : \overline{q_{local}}) \quad \Gamma_q \vdash \text{super}(\overline{y})\ \text{in}\ kd \mid \{C1\} \quad \Gamma_q \vdash \text{this.f} = f \mid \{C2\}}{\vdash_C kd\ \text{is OK} \mid \{q_{ret} \neq \text{readonly},\ q_{ret} \neq \text{polymutable},\ C1,\ C2\}} \quad \text{(WF-CONS)}$$

Note: $\vdash_{C\ kd}$ reads "constructor kd in class C is well-formed".

$$\frac{\text{mBody}(md) = \overline{s}; \text{return}\ z \quad \text{typeof}(md) = q_{this}, \overline{q_p} \rightarrow q_{ret} \quad \Gamma_q = (\text{this} : q_{this},\ \overline{p} : \overline{q_p},\ \overline{y} : \overline{q_{local}}) \quad \Gamma_q \vdash \overline{s} \mid \{C1\} \quad \Gamma_q(z) <: q_{ret} \mid \{C2\} \quad \text{Standard method overriding rule holds} \mid \{C3\}}{\vdash_C md\ \text{is OK} \mid \{C1, C2, C3\}} \quad \text{(WF-METH)}$$

$$\frac{\vdash_C \overline{fd}\ \text{is OK} \mid \{C1\} \qquad \vdash_C kd\ \text{is OK} \mid \{C2\} \qquad \vdash_C \overline{md}\ \text{is OK} \mid \{C3\}}{\vdash C\ \text{is OK} \mid \{C1, C2, C3\}} \quad \text{(WF-CLASS)}$$

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

$$\frac{fType(sfd) = q}{\vdash \text{ sfd is OK} \mid \{q \neq \text{ polymutable}, q \neq \text{receiverdependantmutable}\}} \text{(WF-STATIC-FLD)}$$

$$\frac{mBody(smd) = \bar{s};\text{return z} \quad typeof(smd) = \bar{q}_p \rightarrow q_{ret} \quad \Gamma_q = (\bar{p} : \bar{q}_p,\ \bar{y} : \bar{q}_{local}) \quad \Gamma_q \vdash \bar{s} \mid \{C1\} \quad \Gamma_q(z) <: q_{ret} \mid \{C2\}}{\vdash \text{ smd is OK} \mid \{C1, C2, \bar{q}_p \neq \text{receiverdependantmutable}, q_{ret} \neq \text{receiverdependantmutable}, \text{foreach q in usedQualifiers}(\bar{s};\text{return z}) : q \neq \text{receiverdependantmutable}\}} \text{(WF-STATIC-METH)}$$

$$\frac{\Gamma_q \vdash \bar{s} \mid \{C\}}{\vdash \text{ sib is OK} \mid \{C, \text{foreach q in usedQualifiers}(\bar{s}) : q \neq \text{receiverdependantmutable}\}} \text{(WF-STATIC-BLK)}$$

$$\frac{\Gamma_q \vdash \bar{s} \mid \{C\}}{\vdash_C \text{ ib is OK} \mid \{c\}} \text{(WF-BLK)}$$

$$\frac{\vdash \overline{sfd} \text{ is OK} \mid \{C1\} \quad \vdash_C \overline{fd} \text{ is OK} \mid \{C2\} \quad \vdash_C \text{kd is OK} \mid \{C3\} \quad \vdash \overline{smd} \text{ is OK} \mid \{C4\} \quad \vdash_C \overline{md} \text{ is OK} \mid \{C5\} \quad \vdash \overline{sib} \text{ is OK} \mid \{C6\} \quad \vdash_C \overline{ib} \text{ is OK} \mid \{C7\}}{\vdash \text{C is OK} \mid \{C1, C2, C3, C4, C5, C6, C7\}} \text{(WF-CLASS)}$$