Mechanization and Overhaul of Feature Featherweight Java with Coq

Pedro da C. Abreu Jr.

Universidade de Brasilia

August 08, 2017

Objetivo

Mecanizar Feature Featherweight Java

Objetivo

Mecanizar Feature Featherweight Java

Objetivo

Mecanizar Feature Featherweight Java

Por que mecanizar?

- Aplicações de criticamente seguras: Bugs são inaceitáveis
  - Controladores de Avião
  - Equipamentos Médicos
  - Carros
- Bugs encontrados tardiamente são caros
- A medida que o software e o hardware crescem dificulta encontrar bugs através de testes.

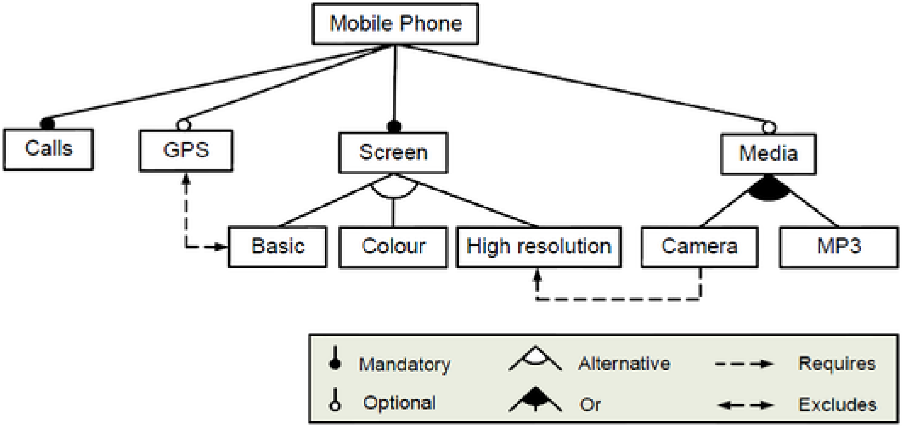
O que é feature?

Feature

'A feature is a unit of functionality of a software system that satisfies a requirement, represents a design decision, and provides a potential configuration option.' - S. Apel & K. Kastner

Feature Oriented Programming

Software Product Line



$\mathtt{CD}$ ::= *class declarations:*

$\mathtt{class\ C\ extends\ D\ \{\bar{C}\ \bar{f};\ K\ \bar{M}\}}$

$\mathtt{K}$ ::= *constructor declarations:*

$\mathtt{C(\bar{C}\ \bar{f})\{super(\bar{f});\ this.\bar{f}=\bar{f};\}}$

$\mathtt{KD}$ ::= *constructor refinements:*

$\mathtt{refines\ C(\bar{E}\ \bar{h},\ \bar{C}\ \bar{f})\{original(\bar{f});\ this.\bar{f}=\bar{f};\}}$

$\mathtt{M}$ ::= *method declarations:*

$\mathtt{C\ m\ (\bar{C}\ \bar{x})\ \{return\ e;\}}$

```
e ::=            expressions:
      x
      e.f
      e.m(ē)
      new C(ē)
      (C)e

v ::=            values:
      new C(ē)
```

$$\frac{}{\texttt{C <: C}} \qquad \frac{\texttt{C <: D} \qquad \texttt{D <: E}}{\texttt{C <: E}} \qquad \frac{\texttt{class C extends D \{ ... \}}}{\texttt{C <: D}}$$

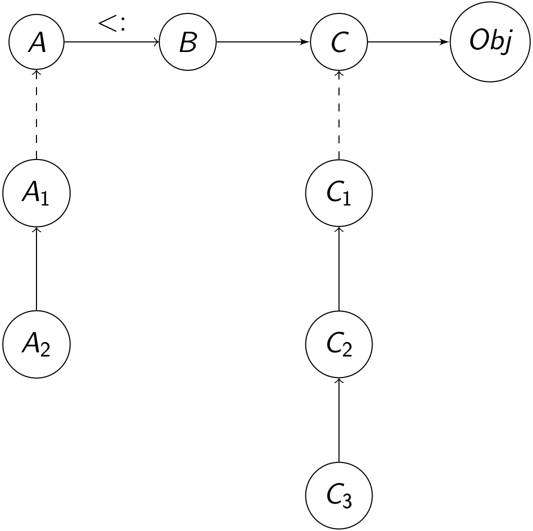
Featherweight Java

Relação de Subtipo

$$A \xrightarrow{<:} B \longrightarrow C \longrightarrow Obj$$

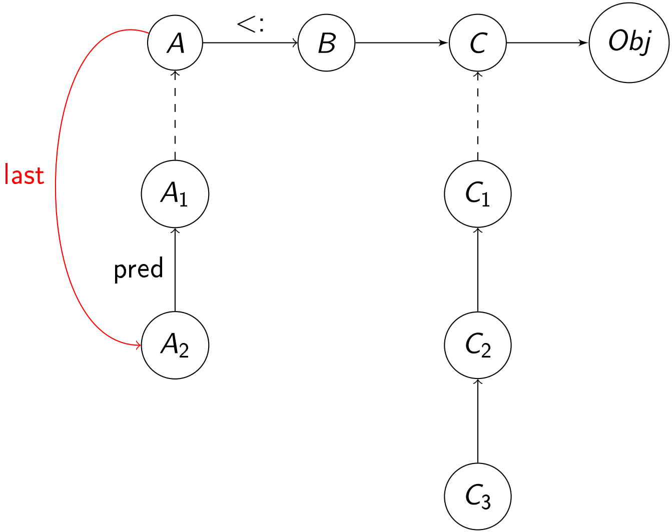
Featherweight Java

Relação de Subtipo



Featherweight Java

Relação de Subtipo



Sintaxe Abstrata

R ::= *refinement names:*

`C@feat`

CR ::= *class refinements:*

$\texttt{refines class R}\ \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{KD}\ \bar{\texttt{M}}\ \overline{\texttt{MR}}\}$

KD ::= *constructor refinements:*

$\texttt{refines C}(\bar{\texttt{E}}\ \bar{\texttt{h}},\ \bar{\texttt{C}}\ \bar{\texttt{f}})\{\texttt{original}(\bar{\texttt{f}});\ \texttt{this}.\bar{\texttt{f}}=\bar{\texttt{f}};\}$

MR ::= *method refinements:*

$\texttt{refines C m}\ (\bar{\texttt{R}}\ \bar{\texttt{x}})\ \{\texttt{return e};\}$

Field Lookup

$$\textit{fields}\ \texttt{Object} = \bullet$$

$$\frac{\texttt{class C extends D } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{K}\ \bar{\texttt{M}}\} \qquad \neg \textit{last}\ \texttt{C}}{\textit{fields}\ \texttt{C} = \textit{fields}\ \texttt{D},\ \bar{\texttt{C}}\ \bar{\texttt{f}}}$$

$$\frac{\texttt{class C extends D } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{K}\ \bar{\texttt{M}}\}}{\textit{fields}\ \texttt{C} = \textit{fields}\ \texttt{D},\ \bar{\texttt{C}}\ \bar{\texttt{f}},\ \textit{fields}_R\ (\textit{last}\ \texttt{C})}$$

Field Lookup (Refinement)

$$\frac{\texttt{refines R \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; KR }\bar{\texttt{M}}\ \bar{\texttt{M}}\texttt{R\}} \qquad \neg \mathit{pred}\ \texttt{R}}{\mathit{fields}_R\ \texttt{R}\ =\ \bar{\texttt{C}}\ \bar{\texttt{f}}}$$

$$\frac{\texttt{refines R \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; KR }\bar{\texttt{M}}\ \bar{\texttt{M}}\texttt{R\}}}{\mathit{fields}_R\ \texttt{R} = \mathit{fields}_R\ (\mathit{pred}\ \texttt{P}), \bar{\texttt{C}}\ \bar{\texttt{f}}}$$

Method Type Lookup

$$\frac{\texttt{class C extends D \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; K}\ \bar{\texttt{M}}\texttt{\}} \qquad \texttt{B m (}\bar{\texttt{B}}\ \bar{\texttt{x}}\texttt{) \{return e;\}} \in \bar{\texttt{M}} \qquad \neg \mathit{mtype}_R\ (\texttt{m}, \mathit{last}\ \texttt{C})}{\mathit{mtype}\ (\texttt{m}, \texttt{C}) \ = \ \bar{\texttt{B}} \ \rightarrow \ \texttt{B}}$$

$$\frac{\texttt{class C extends D \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; K}\ \bar{\texttt{M}}\texttt{\}} \qquad \texttt{m} \notin \bar{\texttt{M}} \qquad \neg \mathit{mtype}_R\ (\texttt{m}, \mathit{last}\ \texttt{C})}{\mathit{mtype}\ (\texttt{m}, \texttt{C}) \ = \ \mathit{mtype}\ (\texttt{m}, \texttt{D})}$$

$$\frac{\texttt{class C extends D \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; K}\ \bar{\texttt{M}}\texttt{\}}}{\mathit{mtype}\ (\texttt{m}, \texttt{C}) \ = \ \mathit{mtype}_R\ (\texttt{m}, \mathit{last}\ \texttt{C})}$$

Method Type Lookup (Refinement)

$$\frac{\texttt{refines class R } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{KR}\ \bar{\texttt{M}}\ \overline{\texttt{MR}}\} \qquad \texttt{B m } (\bar{\texttt{B}}\ \bar{\texttt{x}})\ \{\texttt{return e;}\} \in \bar{\texttt{M}}}{\mathit{mtype}_R\,(\texttt{m}, \texttt{R}) = \bar{\texttt{B}} \rightarrow \texttt{B}}$$

$$\frac{\begin{array}{c}\texttt{refines class R } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{KR}\ \bar{\texttt{M}}\ \overline{\texttt{MR}}\} \qquad \texttt{m} \notin \bar{\texttt{M}} \\ \texttt{refines B m } (\bar{\texttt{B}}\ \bar{\texttt{x}})\ \{\texttt{return e;}\} \in \overline{\texttt{MR}}\end{array}}{\mathit{mtype}_R\,(\texttt{m}, \texttt{R}) = \bar{\texttt{B}} \rightarrow \texttt{B}}$$

$$\frac{\begin{array}{c}\texttt{refines class R } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{KR}\ \bar{\texttt{M}}\ \overline{\texttt{MR}}\} \\ \texttt{m} \notin \bar{\texttt{M}} \quad \texttt{m} \notin \overline{\texttt{MR}}\end{array}}{\mathit{mtype}_R\,(\texttt{m}, \texttt{R}) = \mathit{mtype}_R\,(\texttt{m}, \mathit{pred}\ \texttt{P})}$$

Method Body Lookup

$$\frac{\texttt{class C extends D } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{K}\ \bar{\texttt{M}}\} \qquad \texttt{B m } (\bar{\texttt{B}}\ \bar{\texttt{x}})\ \{\texttt{return e;}\} \in \bar{\texttt{M}} \qquad \neg \mathit{mbody}_R\ (\texttt{m}, \mathit{last}\ \texttt{C})}{\mathit{mbody}\ (\texttt{m}, \texttt{C}) \ = \ \bar{\texttt{x}}.\texttt{e}}$$

$$\frac{\texttt{class C extends D } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{K}\ \bar{\texttt{M}}\} \qquad \texttt{m} \notin \bar{\texttt{M}} \qquad \neg \mathit{mbody}_R\ (\texttt{m}, \mathit{last}\ \texttt{C})}{\mathit{mbody}\ (\texttt{m}, \texttt{C}) \ = \ \mathit{mbody}\ (\texttt{m}, \texttt{D})}$$

$$\frac{\texttt{class C extends D } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{K}\ \bar{\texttt{M}}\}}{\mathit{mbody}\ (\texttt{m}, \texttt{C}) \ = \ \mathit{mbody}_R\ (\texttt{m}, \mathit{last}\ \texttt{C})}$$

Method Body Lookup (Refinement)

$$\frac{\texttt{refines class R \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; KR}\ \bar{\texttt{M}}\ \overline{\texttt{MR}}\texttt{\}} \qquad \texttt{B m (}\bar{\texttt{B}}\ \bar{\texttt{x}}\texttt{) \{return e;\}} \in \bar{\texttt{M}}}{mbody_R\,(\texttt{m},\texttt{R}) \;=\; \bar{\texttt{x}}.\texttt{e}}$$

$$\frac{\begin{array}{c}\texttt{refines class R \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; KR}\ \bar{\texttt{M}}\ \overline{\texttt{MR}}\texttt{\}} \qquad \texttt{m} \notin \bar{\texttt{M}} \\ \texttt{refines B m (}\bar{\texttt{B}}\ \bar{\texttt{x}}\texttt{) \{return e;\}} \in \overline{\texttt{MR}}\end{array}}{mbody_R\,(\texttt{m},\texttt{R}) \;=\; \bar{\texttt{x}}.\texttt{e}}$$

$$\frac{\begin{array}{c}\texttt{refines class R \{}\bar{\texttt{C}}\ \bar{\texttt{f}}\texttt{; KR}\ \bar{\texttt{M}}\ \overline{\texttt{MR}}\texttt{\}} \\ \texttt{m} \notin \bar{\texttt{M}} \quad \texttt{m} \notin \overline{\texttt{MR}}\end{array}}{mbody_R\,(\texttt{m},\texttt{R}) \;=\; mbody_R\,(\texttt{m}, last\ \texttt{P})}$$

$$E ::= \Box \mid E.f_i \mid E.m(\overline{e}) \mid e.m(\overline{e_l}, E, \overline{e_r}) \mid (C)\ E \mid new\ \ C(\overline{e_l}, E, \overline{e_r})$$

Table: Evaluation Context

Lemmas

Lemma (Typed method has body)

If $mtype(m, C) = B \to \overline{B}$
then $\exists \overline{x}\ \exists e$ *such that* $mbody(m, C) = \overline{x}.e$

Lemmas

Lemma (Typed method has body - Refinement)

If $mtype_R(m, R) = \mathtt{B} \rightarrow \mathtt{Bs}$
then $\exists \overline{x}\ \exists e$ *such that* $mbody_R(m, R) = \overline{x}.e$

Lemma (Body method has type - Refinement)

If $mbody_R(m, R) = \overline{x}.e$
then $\exists \overline{B}\ \exists B$ *such that* $mtype_R(m, R) = \mathtt{B} \rightarrow \mathtt{Bs}$

Lemmas

Lemma (Subtype respects method types)

If $\texttt{class C extends D } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{K}\ \bar{\texttt{M}}\}$
then $mtype(m, C) = mtype(m, D)$

Lemmas

Lemma (Refinement respects method types)

If $\texttt{class C extends D } \{\bar{\texttt{C}}\ \bar{\texttt{f}};\ \texttt{K}\ \bar{\texttt{M}}\}$
then $\forall feat,\ mtype_R(m, C@feat) = mtype(m, D)$

Lemmas

Lemma (A1.4 - Method Body is Typable)

If $mtype\ (m, C) = \overline{D} \rightarrow D$
and $mbody\ (\mathtt{m}, \mathtt{C}) = \overline{x}.e$
then $\exists C <: D, \exists C_0 <: D_0,\ this : D, \overline{x} : \overline{D} \vdash e : C_0$

Lemmas

Lemma (Method Body is Typable - Refinement)

If $mtype_R$ $(\mathtt{m}, \mathtt{C@feat}) = \overline{D} \rightarrow D$
and $mbody_R$ $(\mathtt{m}, \mathtt{C@feat}) = \overline{x}.e$
then $\exists C <: D, \exists C_0 <: D_0,$ $this : D, \overline{x} : \overline{D} \vdash e : C_0$

Theorems

Theorem (Preservation)

If $\Gamma \vdash e : C$ *and* $e \rightarrow e'$, *then* $\Gamma \vdash e' : C'$ *for some* $C' <: C$.

Theorems

Theorem (Progress)

Suppose e is closed, well-typed normal form.
Then either (1) e is a value, or (2) for some evaluation context E, we can express e as $e = E[(C)(newD(\overline{e}))]$, with $D \not<: C$.

References I

Harper, Robert.
Practical foundations for programming languages.
Cambridge University Press, 2012.

Igarashi, Atsushi, Benjamin C. Pierce, and Philip Wadler.
Featherweight Java: a minimal core calculus for Java and GJ.
ACM Transactions on Programming Languages and Systems (TOPLAS) 2001.

Apel, Sven, Christian Kästner, and Christian Lengauer.
Feature Featherweight Java: A calculus for feature-oriented programming and stepwise refinement
Proceedings of the 7th international conference on Generative programming and component engineering. ACM, 2008.

Class Name

$$\frac{\mathtt{R} = \mathtt{C@feat}}{\mathit{class\_name}\ \mathtt{R} = \mathtt{C}}$$

Refinements of a class

$$\frac{\mathit{filter}\ (\lambda R \cdot \mathit{class\_name}\ \mathtt{R} == \mathtt{C})\ \mathrm{RT} = \bar{\mathtt{R}}}{\mathit{refinements\_of}\ \mathtt{C} = \bar{\mathtt{R}}}$$

Predecessor

$$\frac{\begin{array}{c}\mathit{refinements\_of}\ (\mathit{class\_name}\ \mathtt{R}) = \bar{\mathtt{R}} \\ \mathit{index}\ \mathtt{R}\ \bar{\mathtt{R}} \ = \ n \qquad \mathit{get}\ (n-1)\ \bar{\mathtt{R}} \ = \ \mathtt{P}\end{array}}{\mathit{pred}\ \mathtt{R} \ = \mathtt{P}}$$

Last

$$\frac{\mathit{refinements\_of}\ \mathtt{C} = \bar{\mathtt{R}} \qquad \mathit{tail}\ \bar{\mathtt{R}} \ = \ \mathtt{R}}{\mathit{last}\ \mathtt{C} \ = \mathtt{R}}$$

FJ Typing Rules

$$\Gamma \vdash x : \Gamma(x) \qquad \text{(T-Var)}$$

$$\frac{\Gamma \vdash e_0 : C_0 \qquad \mathit{fields}\ (C_0) = \overline{C}\ \overline{f}}{\Gamma \vdash e_0.f_i : C_i} \qquad \text{(T-Field)}$$

$$\frac{\Gamma \vdash e_0 : C_0 \qquad \mathit{mtypes}\ (m,\ C_0) = \overline{D} \to C \qquad \Gamma \vdash \overline{e} : \overline{C} \qquad \overline{C}\ <:\ \overline{D}}{\Gamma \vdash e_0.m(\overline{e}) : C} \qquad \text{(T-Invk)}$$

$$\frac{\mathit{fields}(C) = \overline{D}\ \overline{f} \qquad \Gamma \vdash \overline{e} : \overline{C} \qquad \overline{C}\ <:\ \overline{D}}{\Gamma \vdash \mathit{new}\ C(\overline{e}) : C} \qquad \text{(T-New)}$$

$$\frac{\Gamma \vdash e_0 : D \qquad D\ <:\ C}{\Gamma \vdash (C)\ e_0 : C} \qquad \text{(T-UCast)}$$

$$\frac{\Gamma \vdash e_0 : D \qquad C\ <:\ D \qquad C \neq D}{\Gamma \vdash (C)\ e_0 : C} \qquad \text{(T-DCast)}$$

$$\frac{\Gamma \vdash e_0 : D \qquad C\ \not<:\ D \qquad D\ \not<:\ C \qquad \mathit{stupid\ warning}}{\Gamma \vdash (C)\ e_0 : C} \qquad \text{(T-SCast)}$$

FJ Computation Rules

$$\frac{fields\ (C) = \overline{C\, f}}{(new\ \ C(\bar{e})).f_i \rightarrow e_i} \qquad \text{(R-Field)}$$

$$\frac{mbody\ (m, C) = \overline{x}.e_0}{(new\ \ C\ (\overline{e})).m\ (\overline{d}) \rightarrow [\overline{d}/\overline{x}, new\ \ C\ (\bar{e})/this]e_0} \qquad \text{(R-Invk)}$$

$$\frac{C <: D}{(D)(new\ \ C\ (\overline{e})) \rightarrow new\ \ C\ (\overline{e})} \qquad \text{(R-Cast)}$$

FJ Congruence Rules

$$\frac{e_0 \to e_0'}{e_0.f \to e_0'.f} \quad \text{(RC-Field)}$$

$$\frac{e_0 \to e_0'}{e_0.m\ (\overline{e}) \to e_0'.m\ (\overline{e})} \quad \text{(RC-Invk-Recv)}$$

$$\frac{e_i \to e_i'}{e_0.m\ (\ldots, e_i, \ldots) \to e_0'.m\ (\ldots, e_i, \ldots)} \quad \text{(RC-Invk-Arg)}$$

$$\frac{e_i \to e_i'}{new\ \ C\ (\ldots, e_i, \ldots) \to new\ \ C\ (\ldots, e_i', \ldots)} \quad \text{(RC-New-Arg)}$$

$$\frac{e_0 \to e_0'}{(C)e_0 \to (C)e_0'} \quad \text{(RC-Cast)}$$

$$\frac{\mathit{mtype}\ (\mathtt{m}, \mathtt{D}) = \bar{\mathtt{D}} \to \mathtt{D}\ \mathit{implies}\ \bar{\mathtt{C}} = \bar{\mathtt{D}}\ \mathit{and}\ \mathtt{C}_0 = \mathtt{D}}{\mathit{override}\ \mathtt{m}\ \mathtt{D}\ \bar{\mathtt{C}}\ \mathtt{C}_0}$$

$$\frac{\mathtt{class\ C\ extends\ D\ \{}\bar{\mathtt{C}}\ \bar{\mathtt{f}}\mathtt{;\ K}\ \bar{\mathtt{M}}\mathtt{\}} \qquad \mathtt{C}_0\ \mathtt{m}\ (\bar{\mathtt{C}}\ \bar{\mathtt{x}})\ \mathtt{\{return\ e;\}} \in \bar{\mathtt{M}} \qquad \neg\ \mathit{pred}\ \mathtt{R} \qquad \mathtt{R} = \mathtt{C@feat}}{\mathit{override}_R\ \mathtt{m}\ \mathtt{R}\ \bar{\mathtt{C}}\ \mathtt{C}_0}$$

$$\frac{\mathtt{refines\ class\ P\ \{}\bar{\mathtt{C}}\ \bar{\mathtt{f}}\mathtt{;\ KR}\ \bar{\mathtt{M}}\ \overline{\mathtt{MR}}\mathtt{\}} \qquad \mathtt{C}_0\ \mathtt{m}\ (\bar{\mathtt{C}}\ \bar{\mathtt{x}})\ \mathtt{\{return\ e;\}} \in \bar{\mathtt{M}} \qquad \mathit{pred}\ \mathtt{R} = \mathtt{P}}{\mathit{override}_R\ \mathtt{m}\ \mathtt{R}\ \bar{\mathtt{C}}\ \mathtt{C}_0}$$

$$\frac{\mathtt{refines\ class\ P\ \{}\bar{\mathtt{C}}\ \bar{\mathtt{f}}\mathtt{;\ KR}\ \bar{\mathtt{M}}\ \overline{\mathtt{MR}}\mathtt{\}} \qquad \mathtt{m} \notin \bar{\mathtt{M}} \qquad \mathit{pred}\ \mathtt{R} = \mathtt{P} \qquad \mathit{override}_R\ \mathtt{m}\ \mathtt{P}\ \bar{\mathtt{C}}\ \mathtt{C}_0}{\mathit{override}_R\ \mathtt{m}\ \mathtt{R}\ \bar{\mathtt{C}}\ \mathtt{C}_0}$$

$$\frac{\mathit{pred}\ \mathtt{R}\ =\ \mathtt{S} \qquad \neg\ \mathit{mtype}_R\ (\mathtt{m},\mathtt{S})}{\mathit{introduce}\ \mathtt{m}\ \ \mathtt{R}}$$

$$\frac{\begin{array}{c}\neg\ \mathit{pred}\ \mathtt{R} \qquad \mathtt{R}\ =\ \mathtt{C@feat} \\ \mathtt{class\ C\ extends\ D\ \{\bar{C}\ \bar{f};\ K\ \bar{M}\}} \qquad \mathtt{m} \notin \bar{\mathtt{M}}\end{array}}{\mathit{introduce}\ \mathtt{m}\ \ \mathtt{R}}$$

$$\frac{\begin{array}{c}\overline{\mathtt{x}} : \overline{\mathtt{C}}, \mathtt{this} : \mathtt{C} \vdash \mathtt{t}_0 : \mathtt{E}_0 \qquad \mathtt{E}_0 <: \mathtt{C}_0 \\ CT(\mathtt{C}) = \mathtt{class\ C\ extends\ D}\ \{\ldots\} \qquad \mathit{override}(\mathtt{m}, \mathtt{D}, \overline{\mathtt{C}} \to \mathtt{C})\end{array}}{\mathtt{C}_0\ \mathtt{m}\ (\overline{\mathtt{C}}\ \overline{\mathtt{x}})\{\mathtt{return}\ \mathtt{t}_0;\}\ \mathtt{OK\ in\ C}}$$

$$\frac{\begin{array}{c}\overline{\mathtt{x}} : \overline{\mathtt{C}}, \mathtt{this} : \mathtt{C} \vdash \mathtt{t}_0 : \mathtt{E}_0 \qquad \mathtt{E}_0 <: \mathtt{C}_0 \qquad \mathtt{R} = \mathtt{C@feat} \\ CT(\mathtt{C}) = \mathtt{class\ C\ extends\ D}\ \{\ldots\} \qquad RT(\mathtt{R}) = \mathtt{refines\ R}\ \{\ldots\ \overline{\mathtt{M}}\ \ldots\} \\ \mathit{override}(\mathtt{m}, \mathtt{D}, \overline{\mathtt{C}} \to \mathtt{C}) \qquad \mathit{introduce}\ \mathtt{m}\ \mathtt{R} \qquad \mathtt{m} \in \overline{\mathtt{M}}\end{array}}{\mathtt{C}_0\ \mathtt{m}\ (\overline{\mathtt{C}}\ \overline{\mathtt{x}})\{\mathtt{return}\ \mathtt{t}_0;\}\ \mathtt{OK\ in\ R}}$$

$$\frac{\begin{array}{c}\overline{\mathtt{x}} : \overline{\mathtt{C}}, \mathtt{this} : \mathtt{C} \vdash \mathtt{t}_0 : \mathtt{E}_0 \qquad \mathtt{E}_0 <: \mathtt{C}_0 \qquad \mathtt{R} = \mathtt{C@feat} \\ RT(\mathtt{R}) = \mathtt{refines\ R}\ \{\ldots\ \overline{\mathtt{M}},\ \overline{\mathtt{MR}}\ \ldots\} \qquad \mathtt{m} \notin \overline{\mathtt{M}} \qquad \mathtt{m} \in \overline{\mathtt{MR}} \\ \mathit{override}_R(\mathtt{m}, \mathtt{R}, \overline{\mathtt{C}} \to \mathtt{C}) \qquad \mathit{introduce}\ \mathtt{m}\ \mathtt{R}\end{array}}{\mathtt{refines}\ \mathtt{C}_0\ \mathtt{m}\ (\overline{\mathtt{C}}\ \overline{\mathtt{x}})\{\mathtt{return}\ \mathtt{t}_0;\}\ \mathtt{OK\ in\ R}}$$

$$\frac{\texttt{K} \;=\; \texttt{C}\ (\overline{\texttt{D}}\ \overline{\texttt{g}},\ \overline{\texttt{C}}\ \overline{\texttt{f}})\{\texttt{super}(\overline{\texttt{g}});\ \texttt{this}.\overline{\texttt{f}} = \overline{\texttt{f}}\} \qquad \mathit{fields}(\texttt{D}) \;=\; \overline{\texttt{D}}\ \overline{\texttt{g}} \qquad \overline{\texttt{M}}\ \texttt{OK in C}}{\texttt{class C extends D}\ \{\overline{\texttt{C}}\ \overline{\texttt{f}};\ \texttt{K}\ \overline{\texttt{M}}\}\ \texttt{OK}}$$

$$\frac{\overline{\texttt{M}}\ \texttt{OK in R} \qquad \overline{\texttt{MR}}\ \texttt{OK in R}}{\texttt{refines class R}\ \{\overline{\texttt{C}}\ \overline{\texttt{f}};\ \texttt{KR}\ \overline{\texttt{M}}\ \overline{\texttt{MR}}\}\ \texttt{OK}}$$