MiniMAO: A Core Aspect Calculus

Curtis Clifton and Gary T. Leavens
Dept. of Computer Science
Iowa State University

MiniMAO: A Core Aspect Calculus

Curtis Clifton and Gary T. Leavens
Dept. of Computer Science
Iowa State University

Supported
by NSF Grant
CCF-0428078

Research Program

Research Program

- Understand the power of aspects
- Use types for separation of concerns
- Reason within separate concerns
- Understand practical implications

Understand the Power of Aspects

- Designed a core language
- Developed a formal semantics
- Developed a sound, static type system

Use Types for Separation of Concerns

Use Types for Separation of Concerns

- Object-oriented programming:
  - Behavioral subtyping
  - Subtypes represent refinement

Use Types for Separation of Concerns

- Object-oriented programming:
  - Behavioral subtyping
  - Subtypes represent refinement
- Aspect-oriented programming:
  - Extend type system
  - Represent orthogonal concerns

Reason Within Separate Concerns

Reason Within Separate Concerns

- Given:
  - Local verification conditions
  - Global system configuration

Reason Within Separate Concerns

- Given:
  - Local verification conditions
  - Global system configuration
- What can be proven about:
  - Behavior of main program
  - Behavior of aspects

Investigate Practical Implications

Investigate Practical Implications

- Design of future languages

Investigate Practical Implications

- Design of future languages
- Engineering of aspect-oriented systems:
  - Design guidelines
  - Tools needed
  - Automatic reasoning support

Investigate Practical Implications

- Design of future languages
- Engineering of aspect-oriented systems:
  - Design guidelines
  - Tools needed
  - Automatic reasoning support
- Teaching about aspect-oriented programming

Research Program

- Understand the power of aspects
- Use types for separation of concerns
- Reason within separate concerns
- Investigate practical implications

Research Program

- Understand the power of aspects
- Use types for separation of concerns
- Reason within separate concerns
- Investigate practical implications

MiniMAO$_1$: A Core Aspect Language

MiniMAO$_1$: A Core Aspect Language

MiniMAO$_1$: A Core Aspect Language

- Object-oriented base, MiniMAO$_0$

MiniMAO$_1$: A Core Aspect Language

- Object-oriented base, MiniMAO$_0$
- Aspect-oriented extensions, MiniMAO$_1$

MiniMAO$_1$: A Core Aspect Language

- Object-oriented base, MiniMAO$_0$
- Aspect-oriented extensions, MiniMAO$_1$
- Imperative

MiniMAO$_1$: A Core Aspect Language

- Object-oriented base, MiniMAO$_0$
- Aspect-oriented extensions, MiniMAO$_1$
- Imperative
- Advice can change targets

MiniMAO$_1$: A Core Aspect Language

- Object-oriented base, MiniMAO$_0$
- Aspect-oriented extensions, MiniMAO$_1$
- Imperative
- Advice can change targets
- Sound, static type system

MiniMAO$_0$ An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^* \ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c \ \{ \ \mathit{field}^* \ \ \mathit{meth}^* \ \} \\
\mathit{field} &::= t \ f \\
\mathit{meth} &::= t \ m(\ \mathit{form}^* \ ) \ \{ \ e \ \} \\
\mathit{form} &::= t \ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \mid \mathit{var} \mid \text{null} \mid e.m(\ e^* \ ) \mid \\
&\qquad e.f \mid e.f = e \mid \text{cast } t \ e \mid e; e
\end{aligned}
$$

MiniMAO$_0$
An Object-Oriented Base

$$P ::= \mathit{decl}^* \; e$$

$$\mathit{decl} ::= \text{class } c \text{ extends } c \; \{ \; \mathit{field}^* \;\; \mathit{meth}^* \; \}$$

$$\mathit{field} ::= t \; f$$

$$\mathit{meth} ::= t \; m(\; \mathit{form}^* \;) \; \{ \; e \; \}$$

$$\mathit{form} ::= t \; \mathit{var}, \text{ where } \mathit{var} \neq \text{this}$$

$$\begin{aligned} e ::= {} & \text{new } c() \;\mid\; \mathit{var} \;\mid\; \text{null} \;\mid\; e.m(\; e^* \;) \;\mid \\ & e.f \;\mid\; e.f = e \;\mid\; \text{cast } t \; e \;\mid\; e; \, e \end{aligned}$$

MiniMAO$_0$
An Object-Oriented Base

$$P ::= decl^* \; e$$

$$decl ::= \text{class } c \text{ extends } c \; \{ \; field^* \; meth^* \; \}$$

$$field ::= t \; f$$

$$meth ::= t \; m(\; form^* \;) \; \{ \; e \; \}$$

$$form ::= t \; var, \text{ where } var \neq \text{this}$$

$$e ::= \text{new } c() \mid var \mid \text{null} \mid e.m(\; e^* \;) \mid$$
$$e.f \mid e.f = e \mid \text{cast } t \; e \mid e; e$$

MiniMAO$_0$
An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^* \; e \\
\mathit{decl} &::= \text{class } c \text{ extends } c \; \{ \; \mathit{field}^* \; \mathit{meth}^* \; \} \\
\mathit{field} &::= t \; f \\
\mathit{meth} &::= t \; m(\; \mathit{form}^* \;) \; \{ \; e \; \} \\
\mathit{form} &::= t \; \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \; | \; \mathit{var} \; | \; \text{null} \; | \; e.m(\; e^* \;) \; | \\
&\qquad e.f \; | \; e.f = e \; | \; \text{cast } t \; e \; | \; e; \, e
\end{aligned}
$$

MiniMAO$_0$ An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^{*}\ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c\ \{\ \mathit{field}^{*}\ \ \mathit{meth}^{*}\ \} \\
\mathit{field} &::= t\ f \\
\mathit{meth} &::= t\ m(\ \mathit{form}^{*}\ )\ \{\ e\ \} \\
\mathit{form} &::= t\ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \mid \mathit{var} \mid \text{null} \mid e.m(\ e^{*}\ ) \mid \\
&\qquad e.f \mid e.f = e \mid \text{cast } t\ e \mid e;\ e
\end{aligned}
$$

MiniMAO$_0$
An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^{*}\ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c\ \{\ \mathit{field}^{*}\ \ \mathit{meth}^{*}\ \} \\
\mathit{field} &::= t\ f \\
\mathit{meth} &::= t\ m(\ \mathit{form}^{*}\ )\ \{\ e\ \} \\
\mathit{form} &::= t\ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \mid \mathit{var} \mid \text{null} \mid e.m(\ e^{*}\ ) \mid \\
&\qquad e.f \mid e.f = e \mid \text{cast } t\ e \mid e;\ e
\end{aligned}
$$

MiniMAO$_0$ An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^{*}\ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c\ \{\ \mathit{field}^{*}\ \ \mathit{meth}^{*}\ \} \\
\mathit{field} &::= t\ f \\
\mathit{meth} &::= t\ m(\ \mathit{form}^{*}\ )\ \{\ e\ \} \\
\mathit{form} &::= t\ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \boxed{\text{new } c()}\ \mid\ \mathit{var}\ \mid\ \text{null}\ \mid\ e.m(\ e^{*}\ )\ \mid \\
&\qquad e.f\ \mid\ e.f = e\ \mid\ \text{cast } t\ e\ \mid\ e;\ e
\end{aligned}
$$

MiniMAO$_0$ An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^* \; e \\
\mathit{decl} &::= \text{class } c \text{ extends } c \; \{ \; \mathit{field}^* \;\; \mathit{meth}^* \; \} \\
\mathit{field} &::= t \; f \\
\mathit{meth} &::= t \; m(\; \mathit{form}^* \;) \; \{ \; e \; \} \\
\mathit{form} &::= t \; \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \;\mid\; \boxed{\mathit{var}} \;\mid\; \text{null} \;\mid\; e.m(\; e^* \;) \;\mid \\
&\qquad e.f \;\mid\; e.f = e \;\mid\; \text{cast } t \; e \;\mid\; e; \, e
\end{aligned}
$$

MiniMAO$_0$
An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^* \ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c \ \{ \ \mathit{field}^* \ \ \mathit{meth}^* \ \} \\
\mathit{field} &::= t \ f \\
\mathit{meth} &::= t \ m(\ \mathit{form}^* \ ) \ \{ \ e \ \} \\
\mathit{form} &::= t \ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \ \mid \ \mathit{var} \ \mid \boxed{\text{null}} \mid \ e.m(\ e^* \ ) \ \mid \\
&\qquad e.f \ \mid \ e.f = e \ \mid \ \text{cast } t \ e \ \mid \ e; \ e
\end{aligned}
$$

MiniMAO$_0$ An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^{*}\ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c\ \{\ \mathit{field}^{*}\ \ \mathit{meth}^{*}\ \} \\
\mathit{field} &::= t\ f \\
\mathit{meth} &::= t\ m(\ \mathit{form}^{*}\ )\ \{\ e\ \} \\
\mathit{form} &::= t\ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \mid \mathit{var} \mid \text{null} \mid \boxed{e.m(\ e^{*}\ )} \mid \\
&\qquad e.f \mid e.f = e \mid \text{cast } t\ e \mid e; e
\end{aligned}
$$

MiniMAO$_0$
An Object-Oriented Base

$$
\begin{aligned}
P &::= decl^{*}\ e \\
decl &::= \text{class } c \text{ extends } c\ \{\ field^{*}\ \ meth^{*}\ \} \\
field &::= t\ f \\
meth &::= t\ m(\ form^{*}\ )\ \{\ e\ \} \\
form &::= t\ var, \text{ where } var \neq \text{this} \\
e &::= \text{new } c() \mid var \mid \text{null} \mid e.m(\ e^{*}\ ) \mid \\
&\quad \boxed{e.f} \mid e.f = e \mid \text{cast } t\ e \mid e; e
\end{aligned}
$$

MiniMAO$_0$
An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^{*}\ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c\ \{\ \mathit{field}^{*}\ \ \mathit{meth}^{*}\ \} \\
\mathit{field} &::= t\ f \\
\mathit{meth} &::= t\ m(\ \mathit{form}^{*}\ )\ \{\ e\ \} \\
\mathit{form} &::= t\ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \ \mid\ \mathit{var} \ \mid\ \text{null} \ \mid\ e.m(\ e^{*}\ ) \ \mid \\
&\qquad e.f \ \mid\ \boxed{e.f = e} \ \mid\ \text{cast } t\ e \ \mid\ e;\ e
\end{aligned}
$$

MiniMAO$_0$ An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^{*}\ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c\ \{\ \mathit{field}^{*}\ \ \mathit{meth}^{*}\ \} \\
\mathit{field} &::= t\ f \\
\mathit{meth} &::= t\ m(\ \mathit{form}^{*}\ )\ \{\ e\ \} \\
\mathit{form} &::= t\ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \mid \mathit{var} \mid \text{null} \mid e.m(\ e^{*}\ ) \mid \\
&\qquad e.f \mid e.f = e \mid \boxed{\text{cast } t\ e} \mid e;\ e
\end{aligned}
$$

MiniMAO$_0$ An Object-Oriented Base

$$
\begin{aligned}
P &::= \mathit{decl}^* \; e \\
\mathit{decl} &::= \text{class } c \text{ extends } c \; \{ \; \mathit{field}^* \;\; \mathit{meth}^* \; \} \\
\mathit{field} &::= t \; f \\
\mathit{meth} &::= t \; m(\; \mathit{form}^* \;) \; \{ \; e \; \} \\
\mathit{form} &::= t \; \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \;\mid\; \mathit{var} \;\mid\; \text{null} \;\mid\; e.m(\; e^* \;) \;\mid \\
&\qquad e.f \;\mid\; e.f = e \;\mid\; \text{cast } t \; e \;\mid\; \boxed{e; \, e}
\end{aligned}
$$

MiniMAO$_0$ An Object-Oriented Base

$$\begin{aligned}
P &::= \mathit{decl}^* \ e \\
\mathit{decl} &::= \text{class } c \text{ extends } c \ \{ \ \mathit{field}^* \ \ \mathit{meth}^* \ \} \\
\mathit{field} &::= t \ f \\
\mathit{meth} &::= t \ m(\ \mathit{form}^* \ ) \ \{ \ e \ \} \\
\mathit{form} &::= t \ \mathit{var}, \text{ where } \mathit{var} \neq \text{this} \\
e &::= \text{new } c() \ | \ \mathit{var} \ | \ \text{null} \ | \ \boxed{e.m(\ e^* \ )} \ | \\
&\quad\ e.f \ | \ e.f = e \ | \ \text{cast } t \ e \ | \ e; \ e
\end{aligned}$$

Operational Semantics: Locations and Functions

$$e ::= \ldots \mid loc \mid (\ \mathsf{fun}\ m\langle var^{*}\rangle.e : \tau\ (\ e \ldots\ )\ )$$

$$\tau ::= t \times \ldots \times t \rightarrow t$$

$$v ::= loc \mid \mathsf{null}$$

Operational Semantics: Locations and Functions

$$e ::= \ldots \mid \boxed{loc} \mid (\ \mathsf{fun}\ m\langle var^* \rangle .e : \tau\ (\ e \ldots\ )\ )$$

$$\tau ::= t \times \ldots \times t \rightarrow t$$

$$v ::= loc \mid \mathsf{null}$$

Operational Semantics: Locations and Functions

$$e ::= \ldots \mid loc \mid \boxed{(\ \mathsf{fun}\ m\langle var^{*}\rangle.e : \tau\ (\ e \ldots\ )\ )}$$

$$\tau ::= t \times \ldots \times t \rightarrow t$$

$$v ::= loc \mid \mathsf{null}$$

Operational Semantics: Locations and Functions

$$e ::= \ldots \mid \mathit{loc} \mid (\ \mathsf{fun}\ m\langle \mathit{var}^* \rangle.e : \tau\ (\ e \ldots\ )\ )$$

$$\tau ::= t \times \ldots \times t \rightarrow t$$

$$\boxed{v ::= \mathit{loc} \mid \mathsf{null}}$$

Operational Semantics: Evaluation Contexts

$$
\begin{aligned}
\mathbb{E} ::= {} & - \\
& \mid \mathbb{E}\,.m(\,e \ldots\,) \\
& \mid v.m(\,v \ldots \mathbb{E}\,e \ldots\,) \\
& \mid (\,l\,(\,v \ldots \mathbb{E}\,e \ldots\,)\,) \\
& \mid \ldots
\end{aligned}
$$

Key Innovation in Operational Semantics

Key Innovation in Operational Semantics

Key Innovation in Operational Semantics

Classic Java		
Call		

Key Innovation in Operational Semantics

Classic Java	MiniMAO$_0$	
Call	Call	
	Exec	

Key Innovation in Operational Semantics

Classic Java	MiniMAO$_0$	
Call	Call	
	Exec	

Key Innovation in Operational Semantics

Classic Java	MiniMAO$_0$	
Call	Call	looks up method
	Exec	

Key Innovation in Operational Semantics

Classic Java	MiniMAO$_0$	
Call	Call	looks up method
	Exec	substitutes for formals

Operational Semantics: Call and Execution

Operational Semantics: Call and Execution

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, S \rangle \hookrightarrow$$

Operational Semantics: Call and Execution

$$\langle \boxed{\mathbb{E}[loc.m(\ \overline{v}\ )]}, J, S\rangle \hookrightarrow$$

Operational Semantics: Call and Execution

$$\langle \mathbb{E}[\![\boxed{loc.m(\,\overline{v}\,)}]\!], J, S\rangle \hookrightarrow$$

Operational Semantics: Call and Execution

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], \boxed{J}, S\rangle \hookrightarrow$$

Operational Semantics: Call and Execution

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, \boxed{S} \rangle \hookrightarrow$$

Operational Semantics: Call and Execution

Call in Classic Java

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\boxed{e}\{\!|loc/\ \mathrm{this}, \overline{v/\ var}|\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call in Classic Java

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[e\{\!\!|\, loc / \text{ this}, \overline{v / \ var}\, |\!\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call in Classic Java

$$\langle \mathbb{E}[\mathit{loc}.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[e\{\!|\mathit{loc}/\ \text{this}, \boxed{\overline{v/\ \mathit{var}}}|\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call in Classic Java

$$\langle \mathbb{E}[\mathit{loc}.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[e\{\!|\mathit{loc}/\ \text{this}, \overline{v/\ \mathit{var}}|\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call

$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow$

$$\langle \mathbb{E}[e\{\!|loc/\ \text{this}, \overline{v/\ var}|\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call

$$\langle \mathbb{E}[loc.m(\,\overline{v}\,)], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[(\ \mathrm{fun}\ \boxed{m}\langle \mathrm{this}, \overline{var}\rangle.e : \tau\ \ (\,loc, \overline{v}\,)\,)], J, S\rangle$$

$$\langle \mathbb{E}[e\{\!|loc/\ \mathrm{this}, \overline{v/\ var}|\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[(\ \text{fun}\ m\boxed{\langle \text{this}, \overline{var}\rangle}.e : \tau\ \ (\ loc, \overline{v}\ )\ )], J, S\rangle$$

$$\langle \mathbb{E}[e\{\!\!|loc/\ \text{this}, \overline{v/\ var}|\!\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call

$$\langle \mathbb{E}[\mathit{loc}.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[(\ \mathrm{fun}\ m\langle \mathrm{this}, \overline{\mathit{var}}\rangle\boxed{.e:\tau}\ (\ \mathit{loc}, \overline{v}\ )\ )], J, S\rangle$$

$$\langle \mathbb{E}[e\{\!|\mathit{loc}/\ \mathrm{this}, \overline{v/\ \mathit{var}}|\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[(\ \text{fun}\ m\langle \text{this}, \overline{var}\rangle.e : \tau\ \boxed{(\ loc, \overline{v}\ )})], J, S\rangle$$

$$\langle \mathbb{E}[e\{\!|loc/\ \text{this}, \overline{v/\ var}|\!\}], J, S\rangle$$

Operational Semantics: Call and Execution

Call

$$\langle \mathbb{E}[loc.m(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[(\ \text{fun}\ m\langle \text{this}, \overline{var}\rangle.e : \tau\ \ (\ loc, \overline{v}\ )\ )], J, S\rangle$$

Exec

$$\langle \mathbb{E}[(\ \text{fun}\ m\langle \text{this}, \overline{var}\rangle.e : \tau\ \ (\ loc, \overline{v}\ )\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[e\{\!|loc/\ \text{this}, \overline{v/\ var}|\!\}], J, S\rangle$$

MiniMAO$_1$
Aspect-Oriented Extension

MiniMAO$_1$
Aspect-Oriented Extension

$$
\begin{aligned}
\mathit{decl} &::= \ldots \mid \text{aspect}\ a\ \{\ \mathit{field}^*\ \mathit{adv}^*\ \} \\
\mathit{adv} &::= t\ \text{around}(\ \mathit{form}^*\ ) : \mathit{pcd}\ \{\ e\ \} \\
\mathit{pcd} &::= \text{call}(\ \mathit{pat}\ ) \mid \text{execution}(\ \mathit{pat}\ ) \mid \\
&\qquad \text{this}(\ \mathit{form}\ ) \mid \text{target}(\ \mathit{form}\ ) \mid \text{args}(\ \mathit{form}^*\ ) \mid \\
&\qquad \mathit{pcd}\ \&\&\ \mathit{pcd} \mid\ !\ \mathit{pcd} \mid \mathit{pcd}\ ||\ \mathit{pcd} \\
\mathit{pat} &::= t\ \mathit{idPat}(..) \\
e &::= \ldots \mid e.\text{proceed}(\ e^*\ )
\end{aligned}
$$

MiniMAO$_1$
Aspect-Oriented Extension

$$
\begin{array}{rcl}
\mathit{decl} & ::= & \ldots \mid \boxed{\text{aspect } a \ \{\ \mathit{field}^{*}\ \mathit{adv}^{*}\ \}} \\
\mathit{adv} & ::= & t \text{ around}(\ \mathit{form}^{*}\ ) : \mathit{pcd}\ \{\ e\ \} \\
\mathit{pcd} & ::= & \text{call}(\ \mathit{pat}\ ) \mid \text{execution}(\ \mathit{pat}\ ) \mid \\
 & & \quad \text{this}(\ \mathit{form}\ ) \mid \text{target}(\ \mathit{form}\ ) \mid \text{args}(\ \mathit{form}^{*}\ ) \mid \\
 & & \quad \mathit{pcd}\ \&\&\ \mathit{pcd} \mid\ !\ \mathit{pcd} \mid \mathit{pcd}\ ||\ \mathit{pcd} \\
\mathit{pat} & ::= & t\ \mathit{idPat}(..) \\
e & ::= & \ldots \mid e.\text{proceed}(\ e^{*}\ )
\end{array}
$$

MiniMAO$_1$ Aspect-Oriented Extension

$$
\begin{aligned}
\mathit{decl} &::= \ldots \mid \text{aspect } a \,\{\, \mathit{field}^{*}\ \mathit{adv}^{*} \,\} \\
\mathit{adv} &::= t \text{ around(} \mathit{form}^{*} \text{) : } \mathit{pcd} \,\{\, e \,\} \\
\mathit{pcd} &::= \text{call(} \mathit{pat} \text{)} \mid \text{execution(} \mathit{pat} \text{)} \mid \\
&\qquad \text{this(} \mathit{form} \text{)} \mid \text{target(} \mathit{form} \text{)} \mid \text{args(} \mathit{form}^{*} \text{)} \mid \\
&\qquad \mathit{pcd} \;\&\&\; \mathit{pcd} \mid\ !\ \mathit{pcd} \mid \mathit{pcd} \mid\mid \mathit{pcd} \\
\mathit{pat} &::= t\ \mathit{idPat}(..) \\
e &::= \ldots \mid e\text{.proceed(} e^{*} \text{)}
\end{aligned}
$$

MiniMAO$_1$
Aspect-Oriented Extension

$$decl ::= \ldots \mid \text{aspect } a \ \{ \, field^* \ adv^* \, \}$$

$$adv ::= t \ \text{around}(\, form^* \,) : pcd \ \{ \, e \, \}$$

$$pcd ::= \boxed{\text{call}(\, pat \,) \mid \text{execution}(\, pat \,)} \mid$$

$$\text{this}(\, form \,) \mid \text{target}(\, form \,) \mid \text{args}(\, form^* \,) \mid$$

$$pcd \ \&\& \ pcd \mid \, ! \ pcd \mid pcd \ || \ pcd$$

$$pat ::= t \ idPat(..)$$

$$e ::= \ldots \mid e.\text{proceed}(\, e^* \,)$$

MiniMAO$_1$
Aspect-Oriented Extension

$$
\begin{aligned}
decl &::= \ldots \mid \text{aspect } a \; \{ \; field^{*} \; adv^{*} \; \} \\
adv &::= t \; \text{around}(\; form^{*} \;) : pcd \; \{ \; e \; \} \\
pcd &::= \text{call}(\; pat \;) \mid \text{execution}(\; pat \;) \mid \\
&\qquad \boxed{\text{this}(\; form \;) \mid \text{target}(\; form \;) \mid \text{args}(\; form^{*} \;)} \mid \\
&\qquad pcd \;\&\&\; pcd \mid \; ! \; pcd \mid pcd \; || \; pcd \\
pat &::= t \; idPat(..) \\
e &::= \ldots \mid e.\text{proceed}(\; e^{*} \;)
\end{aligned}
$$

MiniMAO$_1$ Aspect-Oriented Extension

$$
\begin{aligned}
decl &::= \ldots \mid \text{aspect } a \; \{ \, field^{*} \; adv^{*} \, \} \\
adv &::= t \text{ around}(\, form^{*} \,) : pcd \; \{ \, e \, \} \\
pcd &::= \text{call}(\, pat \,) \mid \text{execution}(\, pat \,) \mid \\
&\qquad \text{this}(\, form \,) \mid \text{target}(\, form \,) \mid \text{args}(\, form^{*} \,) \mid \\
&\qquad \boxed{pcd \;\&\&\; pcd \mid \, ! \; pcd \mid pcd \, || \, pcd} \\
pat &::= t \; idPat(..) \\
e &::= \ldots \mid e.\text{proceed}(\, e^{*} \,)
\end{aligned}
$$

MiniMAO$_1$ Aspect-Oriented Extension

$$
\begin{aligned}
\mathit{decl} &::= \ldots \mid \text{aspect } a \;\{\, \mathit{field}^* \; \mathit{adv}^* \,\} \\
\mathit{adv} &::= t \text{ around}(\, \mathit{form}^* \,) : \mathit{pcd} \;\{\, e \,\} \\
\mathit{pcd} &::= \text{call}(\, \mathit{pat} \,) \mid \text{execution}(\, \mathit{pat} \,) \mid \\
&\qquad \text{this}(\, \mathit{form} \,) \mid \text{target}(\, \mathit{form} \,) \mid \text{args}(\, \mathit{form}^* \,) \mid \\
&\qquad \mathit{pcd} \;\&\&\; \mathit{pcd} \mid \;!\; \mathit{pcd} \mid \mathit{pcd} \;||\; \mathit{pcd} \\
&\boxed{\mathit{pat} ::= t \; \mathit{idPat}(..)} \\
e &::= \ldots \mid e.\text{proceed}(\, e^* \,)
\end{aligned}
$$

MiniMAO$_1$ Aspect-Oriented Extension

$$
\begin{aligned}
\mathit{decl} &::= \ldots \mid \text{aspect } a\ \{\ \mathit{field}^*\ \mathit{adv}^*\ \} \\
\mathit{adv} &::= t \text{ around(}\ \mathit{form}^*\ \text{)} : \mathit{pcd}\ \{\ e\ \} \\
\mathit{pcd} &::= \text{call(}\ \mathit{pat}\ \text{)} \mid \text{execution(}\ \mathit{pat}\ \text{)} \mid \\
&\qquad \text{this(}\ \mathit{form}\ \text{)} \mid \text{target(}\ \mathit{form}\ \text{)} \mid \text{args(}\ \mathit{form}^*\ \text{)} \mid \\
&\qquad \mathit{pcd}\ \&\&\ \mathit{pcd} \mid\ !\ \mathit{pcd} \mid \mathit{pcd}\ \|\ \mathit{pcd} \\
\mathit{pat} &::= t\ \mathit{idPat}(..) \\
e &::= \ldots \mid \boxed{e.\text{proceed(}\ e^*\ \text{)}}
\end{aligned}
$$

Different Form of Proceed

```
void around(Author a,
            Pub p, int amt) :
    call(void royalty(..))
    && target(Author a)
    && this(Pub p)
    && args(int amt)
{
    proceed(a, p, amt);
}
```

```
void around(Author a,
            Pub p, int amt) :
    call(void royalty(..))
    && target(Author a)
    && this(Pub p)
    && args(int amt)
{
    a.proceed(amt);
}
```

Different Form of Proceed

```
void around(Author a,
               Pub p, int amt) :
     call(void royalty(..))
     && target(Author a)
     && this(Pub p)
     && args(int amt)
{
     proceed(a, p, amt);
}
```

AspectJ

```
void around(Author a,
               Pub p, int amt) :
     call(void royalty(..))
     && target(Author a)
     && this(Pub p)
     && args(int amt)
{
     a.proceed(amt);
}
```

Different Form of Proceed

```
void around(Author a,
             Pub p, int amt) :
    call(void royalty(..))
    && target(Author a)
    && this(Pub p)
    && args(int amt)
{
    proceed(a, p, amt);
}
```

AspectJ

```
void around(Author a,
             Pub p, int amt) :
    call(void royalty(..))
    && target(Author a)
    && this(Pub p)
    && args(int amt)
{
    a.proceed(amt);
}
```

Different Form of Proceed

```
void around(Author a,
            Pub p, int amt) :
    call(void royalty(..))
    && target(Author a)
    && this(Pub p)
    && args(int amt)
{
    proceed(a, p, amt);
}
```

AspectJ

```
void around(Author a,
            Pub p, int amt) :
    call(void royalty(..))
    && target(Author a)
    && this(Pub p)
    && args(int amt)
{
    a.proceed(amt);
}
```

MiniMAO_1

Different Form of Proceed

```
void around(Author a,
               Pub p, int amt) :
     call(void royalty(..))
     && target(Author a)
     && this(Pub p)
     && args(int amt)
{
     proceed(a, p, amt);
}
```

AspectJ

```
void around(Author a,
               Pub p, int amt) :
     call(void royalty(..))
     && target(Author a)
     && this(Pub p)
     && args(int amt)
{
     a.proceed(amt);
}
```

MiniMAO$_1$

Operational Semantics: Join Points and Chains

$$
\begin{aligned}
e ::= {} & \ldots \\
& \mid \text{joinpt } j(\ e^{*}\ ) \\
& \mid \text{under } e \\
& \mid \text{chain } \bar{B}, j(\ e^{*}\ )
\end{aligned}
$$

$$
\begin{aligned}
\bar{B} &::= B + \bar{B} \mid \bullet \\
B &::= \llbracket b, loc, e, \tau, \tau \rrbracket \\
b &::= \langle \alpha, \beta, \beta^{*} \rangle \\
\alpha &::= var \mapsto loc \mid - \\
\beta &::= var \mid -
\end{aligned}
$$

Operational Semantics: Join Points and Chains

$$
\begin{aligned}
e ::= {} & \ldots \\
& \mid \boxed{\text{joinpt } j(\; e^* \;)} \\
& \mid \text{under } e \\
& \mid \text{chain } \bar{B}, j(\; e^* \;)
\end{aligned}
$$

$$
\begin{aligned}
\bar{B} &::= B + \bar{B} \mid \bullet \\
B &::= \llbracket b, loc, e, \tau, \tau \rrbracket \\
b &::= \langle \alpha, \beta, \beta^* \rangle \\
\alpha &::= var \mapsto loc \mid - \\
\beta &::= var \mid -
\end{aligned}
$$

Operational Semantics: Join Points and Chains

$$\begin{aligned} e ::= {} & \ldots \\ & \mid \text{joinpt } j(\ e^*\ ) \\ & \mid \text{under } e \\ & \mid \boxed{\text{chain } \bar{B}, j(\ e^*\ )} \end{aligned}$$

$$\begin{aligned} \bar{B} &::= B + \bar{B} \mid \bullet \\ B &::= \llbracket b, loc, e, \tau, \tau \rrbracket \\ b &::= \langle \alpha, \beta, \beta^* \rangle \\ \alpha &::= var \mapsto loc \mid - \\ \beta &::= var \mid - \end{aligned}$$

Operational Semantics: Join Points and Chains

$$
\begin{aligned}
e ::= &\ \ldots \\
&\mid \text{joinpt } j(\ e^*\ ) \\
&\mid \text{under } e \\
&\mid \text{chain } \bar{B}, j(\ e^*\ )
\end{aligned}
$$

$$
\begin{aligned}
\bar{B} &::= B + \bar{B} \mid \bullet \\
B &::= \boxed{\llbracket b, loc, e, \tau, \tau \rrbracket} \\
b &::= \langle \alpha, \beta, \beta^* \rangle \\
\alpha &::= var \mapsto loc \mid - \\
\beta &::= var \mid -
\end{aligned}
$$

Operational Semantics: Join Points and Chains

$$
\begin{aligned}
e ::= {} & \ldots \\
& \mid \text{joinpt } j(\ e^*\ ) \\
& \mid \text{under } e \\
& \mid \text{chain } \bar{B}, j(\ e^*\ )
\end{aligned}
$$

$$
\begin{aligned}
\bar{B} &::= B + \bar{B} \mid \bullet \\
B &::= \boxed{\llbracket b, loc, e, \tau, \tau \rrbracket} \\
b &::= \langle \alpha, \beta, \beta^* \rangle \\
\alpha &::= var \mapsto loc \mid - \\
\beta &::= var \mid -
\end{aligned}
$$

Operational Semantics: Join Points and Chains

$$
\begin{aligned}
e ::= {} & \ldots \\
& \mid \ \text{joinpt } j(\ e^*\ ) \\
& \mid \ \text{under } e \\
& \mid \ \text{chain } \bar{B}, j(\ e^*\ )
\end{aligned}
$$

$$
\begin{aligned}
\bar{B} &::= B + \bar{B} \mid \bullet \\
B &::= \boxed{\llbracket b, loc, e, \tau, \tau \rrbracket} \\
b &::= \langle \alpha, \beta, \beta^* \rangle \\
\alpha &::= var \mapsto loc \mid - \\
\beta &::= var \mid -
\end{aligned}
$$

Operational Semantics: Join Points and Chains

$$
\begin{aligned}
e ::=\ & \ldots \\
\mid\ & \text{joinpt } j(\ e^*\ ) \\
\mid\ & \text{under } e \\
\mid\ & \text{chain } \bar{B}, j(\ e^*\ )
\end{aligned}
$$

$$
\begin{aligned}
\bar{B} &::= B + \bar{B} \mid \bullet \\
B &::= \boxed{\lceil\lfloor b, loc, e, \tau, \tau \rfloor\rceil} \\
b &::= \langle \alpha, \beta, \beta^* \rangle \\
\alpha &::= var \mapsto loc \mid - \\
\beta &::= var \mid -
\end{aligned}
$$

Operational Semantics: Join Points and Chains

$$
\begin{aligned}
e ::= {} & \ldots \\
\mid {} & \text{joinpt } j(\ e^{*}\ ) \\
\mid {} & \text{under } e \\
\mid {} & \text{chain } \bar{B}, j(\ e^{*}\ )
\end{aligned}
$$

$$
\begin{aligned}
\bar{B} &::= B + \bar{B} \mid \bullet \\
B &::= \boxed{\llbracket b, loc, e, \tau, \tau \rrbracket} \\
b &::= \langle \alpha, \beta, \beta^{*} \rangle \\
\alpha &::= var \mapsto loc \mid - \\
\beta &::= var \mid -
\end{aligned}
$$

Key Innovation in Operational Semantics

Key Innovation in Operational Semantics

Key Innovation in Operational Semantics

MiniMAO$_0$		
Call		

Key Innovation in Operational Semantics

MiniMAO$_0$	MiniMAO$_1$	
Call	Call$_A$	
	Bind	
	Advise	
	Call$_B$	

Key Innovation in Operational Semantics

MiniMAO$_0$	MiniMAO$_1$	
Call	Call$_A$	
	Bind	
	Advise	
	Call$_B$	

Key Innovation in Operational Semantics

MiniMAO$_0$	MiniMAO$_1$	
Call	Call$_A$	creates a join point
	Bind	
	Advise	
	Call$_B$	

Key Innovation in Operational Semantics

MiniMAO$_0$	MiniMAO$_1$	
CALL	CALL$_A$	creates a join point
	BIND	looks up advice
	ADVISE	
	CALL$_B$	

Key Innovation in Operational Semantics

MiniMAO$_0$	MiniMAO$_1$	
CALL	CALL$_A$	creates a join point
	BIND	looks up advice
	ADVISE	executes advice
	CALL$_B$	

Key Innovation in Operational Semantics

MiniMAO$_0$	MiniMAO$_1$	
Call	Call$_A$	creates a join point
	Bind	looks up advice
	Advise	executes advice
	Call$_B$	does original operation

Key Innovation in Operational Semantics

MiniMAO$_0$	MiniMAO$_1$	
Exec Call	Exec$_A$ Call$_A$	creates a join point
	Bind	looks up advice
	Advise	executes advice
	Exec$_B$ Call$_B$	does original operation

Operational Semantics: Splitting the Call Rule

Call in MiniMAO$_0$

$$\langle \mathbb{E}[v_0.m(\ v_1, \ldots, v_n\ )], J, S \rangle \hookrightarrow \langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S \rangle$$

Operational Semantics: Splitting the Call Rule

Call in MiniMAO$_0$

$\langle \mathbb{E}[v_0.m(\ v_1,\ldots,v_n\ )], J, S\rangle \hookrightarrow$

$\langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S\rangle$

Operational Semantics: Splitting the Call Rule

Call$_{\text{A}}$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\,v_1,\ldots,v_n\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\,\overline{v}\,)], J, S\rangle$$

$$\langle \mathbb{E}[\text{chain } \llbracket b, loc, e, \text{\textvisiblespace}, \text{\textvisiblespace} \rrbracket + \bar{B}, j(\,\overline{v}\,)], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e'\{\!|loc/\,\text{this}|\!\}\{\!|\overline{v}/\,b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\,l\,(\,\overline{v}\,)\,)], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\, v_1, \ldots, v_n \,)], J, S \rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } \boxed{(\!|\text{call}, -, m, -, \tau|\!)}(\, \overline{v} \,)], J, S \rangle$$

$$\langle \mathbb{E}[\text{chain } [\![b, loc, e, \llcorner\!\lrcorner, \llcorner\!\lrcorner]\!] + \bar{B}, j(\, \overline{v} \,)], J, S \rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } e' \{\!| loc / \text{ this} |\!\} \{\!| \overline{v} / \, b |\!\}], j + J, S \rangle$$

$$\langle \mathbb{E}[(\, l \, (\, \overline{v} \,) \,)], J, S \rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\ v_1,\ldots,v_n\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\ \boxed{\overline{v}}\ )], J, S\rangle$$

$$\langle \mathbb{E}[\text{chain } \lceil\!\lfloor b, loc, e, \text{\textvisiblespace}, \text{\textvisiblespace} \rfloor\!\rceil + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc /\ \text{this}|\!\}\{\!|\overline{v} /\ b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_{\text{A}}$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\, v_1, \ldots, v_n \,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\, \overline{v} \,)], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\, \overline{v} \,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\, \overline{v} \,)], j + J, S\rangle$$

$$\langle \mathbb{E}[\text{chain } |\lfloor b, loc, e, \textvisiblespace, \textvisiblespace \rfloor| + \bar{B}, j(\, \overline{v} \,)], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc/\ \text{this}|\!\} \{\!|\overline{v}/\ b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\, v_1, \ldots, v_n\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\,\overline{v}\,)], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\,\overline{v}\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \boxed{\bar{B},}\, j(\,\overline{v}\,)], j + J, S\rangle$$

$$\langle \mathbb{E}[\text{chain } [\![b, loc, e, \text{\textvisiblespace}, \text{\textvisiblespace}]\!] + \bar{B}, j(\,\overline{v}\,)], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{this}|\!\}\{\!|\overline{v} / b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\, l\ (\,\overline{v}\,)\,)], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

CALL$_{\text{A}}$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\, v_1, \ldots, v_n\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\,\overline{v}\,)], J, S\rangle$$

BIND

$$\langle \mathbb{E}[\text{joinpt } j(\,\overline{v}\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\,\overline{v}\,)], \boxed{j + J}, S\rangle$$

$$\langle \mathbb{E}[\text{chain } [\![b, loc, e, \textvisiblespace, \textvisiblespace]\!] + \bar{B}, j(\,\overline{v}\,)], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e'\{\!|loc / \text{this}|\!\}\{\!|\overline{v} / b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\, l\ (\,\overline{v}\,)\,)], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\ v_1,\ldots,v_n\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\ \overline{v}\ )], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\boxed{\text{under}}\ \text{chain } \bar{B}, j(\ \overline{v}\ )], j + J, S\rangle$$

$$\langle \mathbb{E}[\text{chain } [\![b, loc, e, \text{\textvisiblespace}, \text{\textvisiblespace}]\!] + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{ this}|\!\} \{\!|\overline{v} /\ b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\,v_1,\ldots,v_n\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\,\overline{v}\,)], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\,\overline{v}\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\,\overline{v}\,)], j + J, S\rangle$$

Advise

$$\langle \mathbb{E}[\text{chain } \llbracket b, loc, e, \textvisiblespace, \textvisiblespace \rrbracket + \bar{B}, j(\,\overline{v}\,)], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } e'\{\!|loc/\,\text{this}|\!\}\{\!|\overline{v}/\,b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\; l\;(\,\overline{v}\,)\,)], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\ v_1, \ldots, v_n\ )], J, S \rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\ \overline{v}\ )], J, S \rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\ \overline{v}\ )], J, S \rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\ \overline{v}\ )], j + J, S \rangle$$

Advise

$$\langle \mathbb{E}[\text{chain } \boxed{[\![b, loc, e, \llcorner\!\lrcorner, \llcorner\!\lrcorner]\!]} + \bar{B}, j(\ \overline{v}\ )], J, S \rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{this}|\!\} \{\!|\overline{v} / b|\!\}], j + J, S \rangle$$

$$\langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S \rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\, v_1, \ldots, v_n \,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\, \overline{v} \,)], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\, \overline{v} \,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\, \overline{v} \,)], j + J, S\rangle$$

Advise

$$\langle \mathbb{E}[\text{chain } [\![b, loc, e, \textvisiblespace, \textvisiblespace]\!] + \bar{B}, j(\, \overline{v} \,)], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } e' \boxed{\{\!| loc / \text{ this} |\!\}} \{\!| \overline{v} / \, b |\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\; l \; (\, \overline{v} \,) \;)], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_A$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\,v_1,\ldots,v_n\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\,\overline{v}\,)], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\,\overline{v}\,)], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\,\overline{v}\,)], j + J, S\rangle$$

Advise

$$\langle \mathbb{E}[\text{chain } \llbracket b, loc, e, \textvisiblespace, \textvisiblespace \rrbracket + \bar{B}, j(\,\overline{v}\,)], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{this}|\!\} \boxed{\{\!|\overline{v} / b|\!\}}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\ l\ (\,\overline{v}\,)\ )], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_\text{A}$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\ v_1,\ldots,v_n\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\ \overline{v}\ )], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\ \overline{v}\ )], j + J, S\rangle$$

Advise

$$\langle \mathbb{E}[\text{chain } [\![b, loc, e, \sqcup, \sqcup]\!] + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } \boxed{e'}\{\!|loc/\ \text{this}|\!\}\{\!|\overline{v}/\ b|\!\}], j + J, S\rangle$$

$$\langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S\rangle$$

Operational Semantics: Splitting the Call Rule

Call$_{\text{A}}$ in MiniMAO$_1$

$$\langle \mathbb{E}[v_0.m(\ v_1, \ldots, v_n\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{joinpt } (\!|\text{call}, -, m, -, \tau|\!)(\ \overline{v}\ )], J, S\rangle$$

Bind

$$\langle \mathbb{E}[\text{joinpt } j(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[\text{under chain } \bar{B}, j(\ \overline{v}\ )], j + J, S\rangle$$

Advise

$$\langle \mathbb{E}[\text{chain } \lceil\lfloor b, loc, e, \llcorner\!\lrcorner, \llcorner\!\lrcorner \rfloor\rceil + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{ this}|\!\}\{\!|\overline{v} / \ b|\!\}], j + J, S\rangle$$

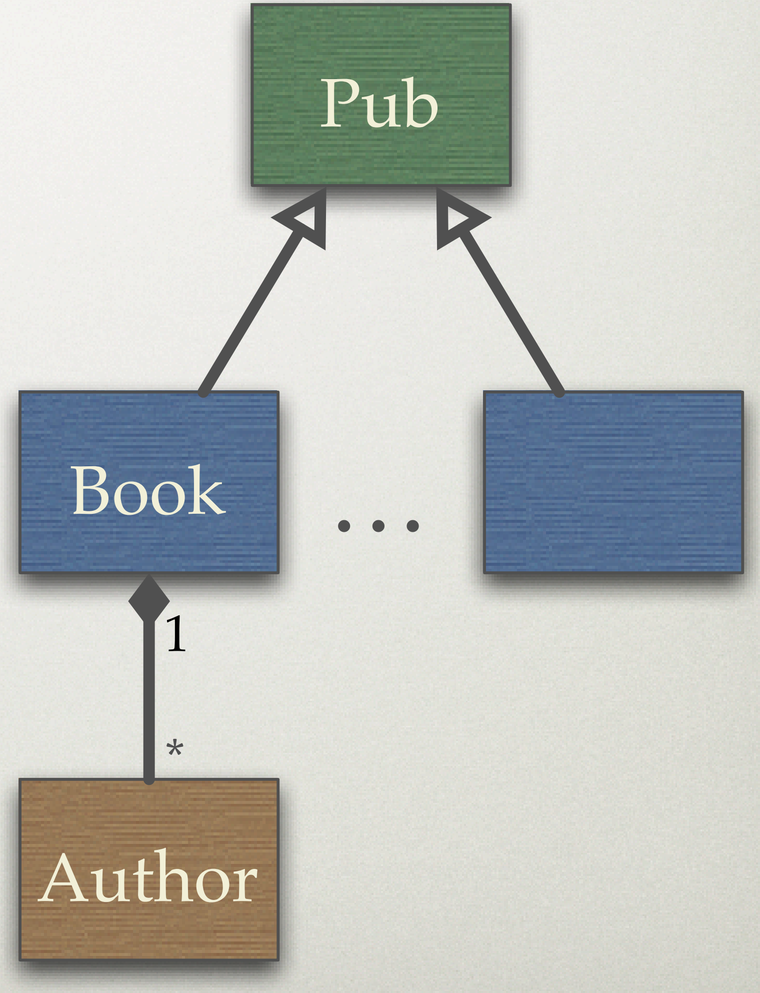
Call$_{\text{B}}$

$$\langle \mathbb{E}[\text{chain } \bullet, (\!|\text{call}, -, m, -, \tau|\!)(\ \overline{v}\ )], J, S\rangle \hookrightarrow \langle \mathbb{E}[(\ l\ (\ \overline{v}\ )\ )], J, S\rangle$$

Some Examples

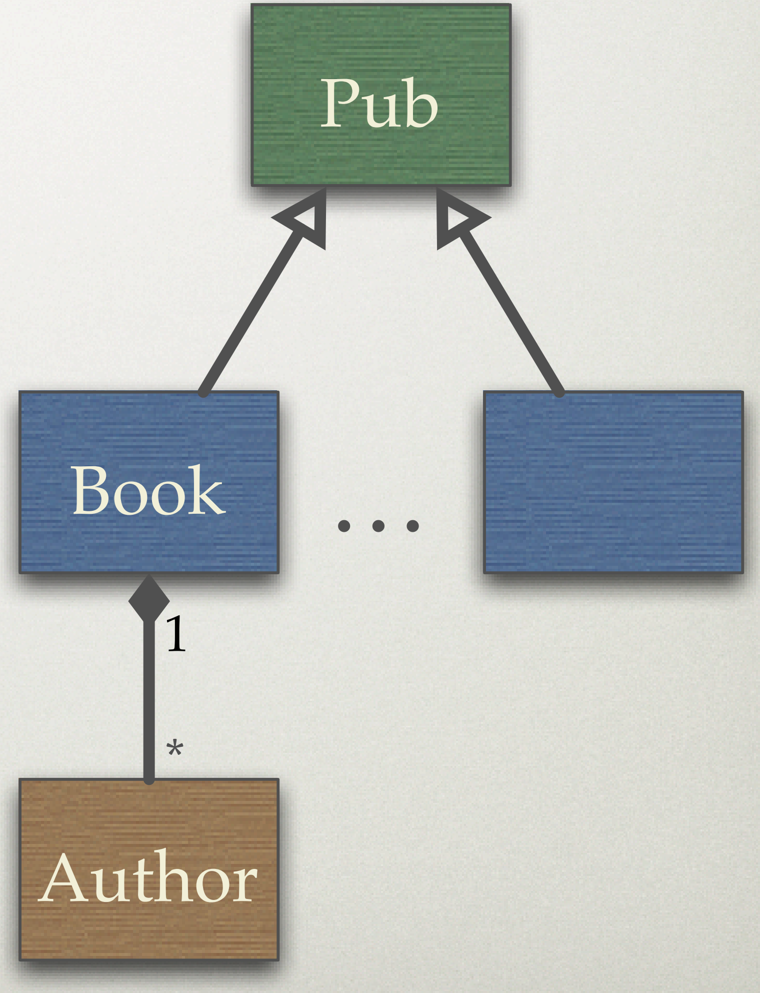
Running Example

Running Example



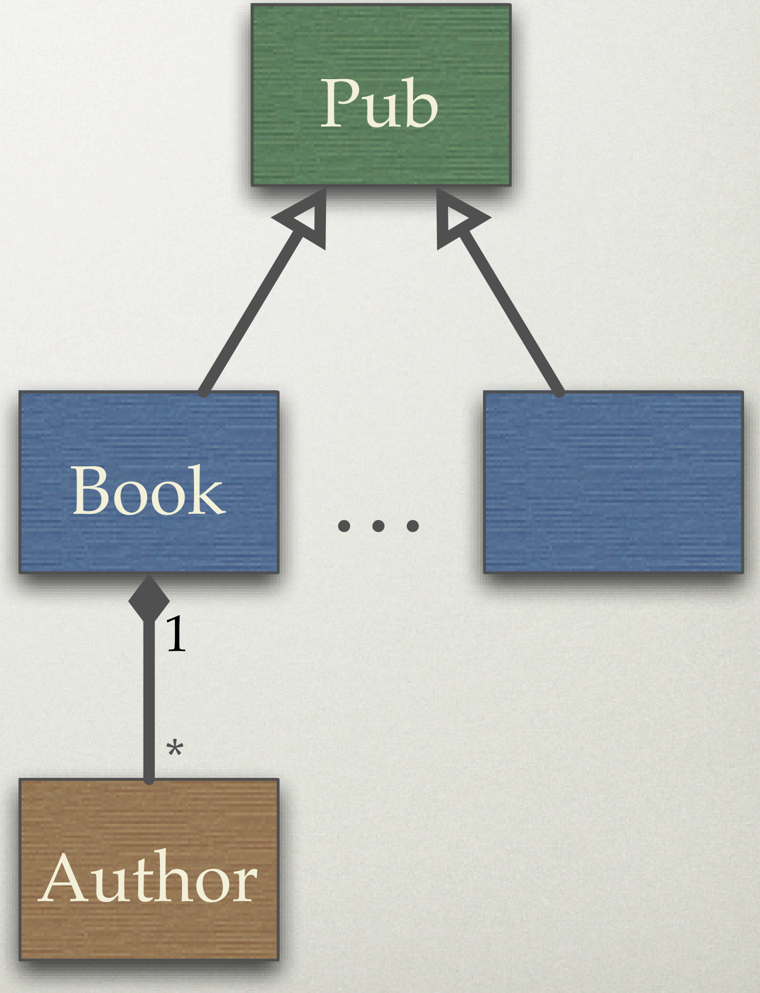
Running Example

- Assume Author includes: royalty(int)



Running Example

- Assume Author includes: royalty(int)
- Log every call to Author's royalty from any Pub



Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:″ + p + “ calls ″ + a );
        a.proceed(amt);
        this.log(“After:″ + p + “ calls ″ + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

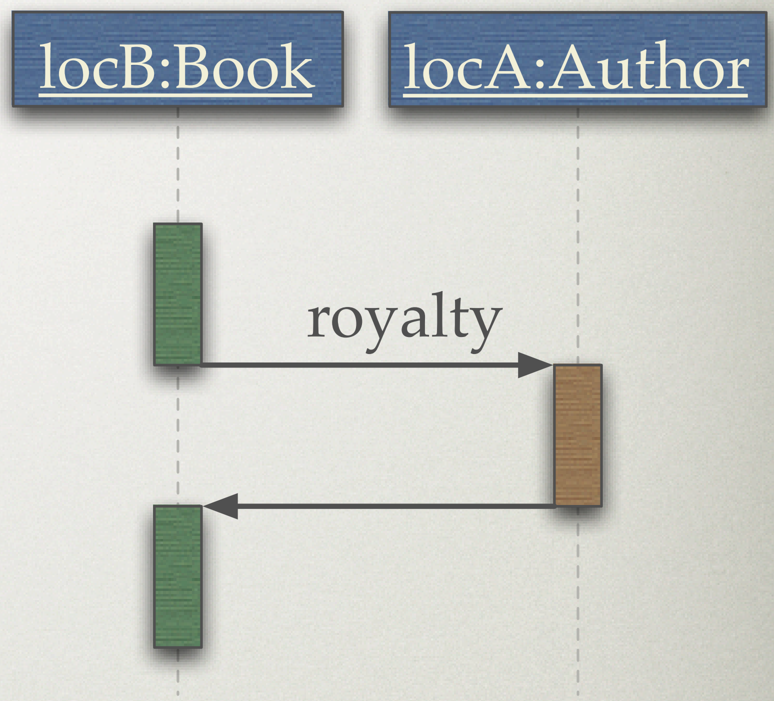
Advice

```
aspect Logger {
    void around(Author a, Pub p, int amt) :
        call(void royalty(..)) && target(Author a) &&
        this(Pub p) && args(int amt)
    {
        this.log(“Before:” + p + “ calls ” + a );
        a.proceed(amt);
        this.log(“After:” + p + “ calls ” + a );
    }
    …
}
```

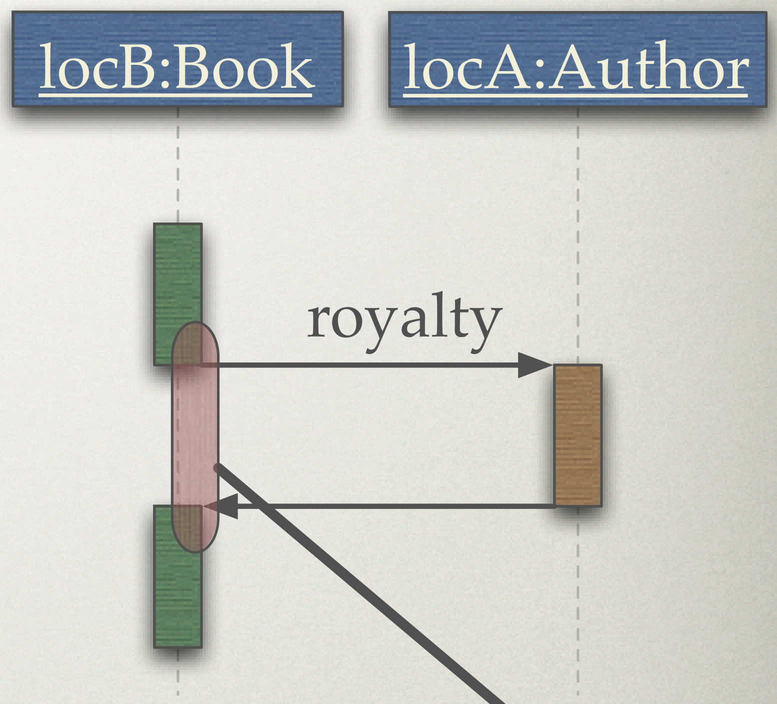
Pointcut Matching for the Bind Rule

Pointcut Matching

Pointcut Matching



Pointcut Matching



$$j = (\!|\ \mathsf{call}, -, \mathsf{royalty}, -,\ \mathsf{Author} \times \mathsf{int} \rightarrow \mathsf{void}\ |\!)$$

Pointcut Matching

matchPCD($j + J$,

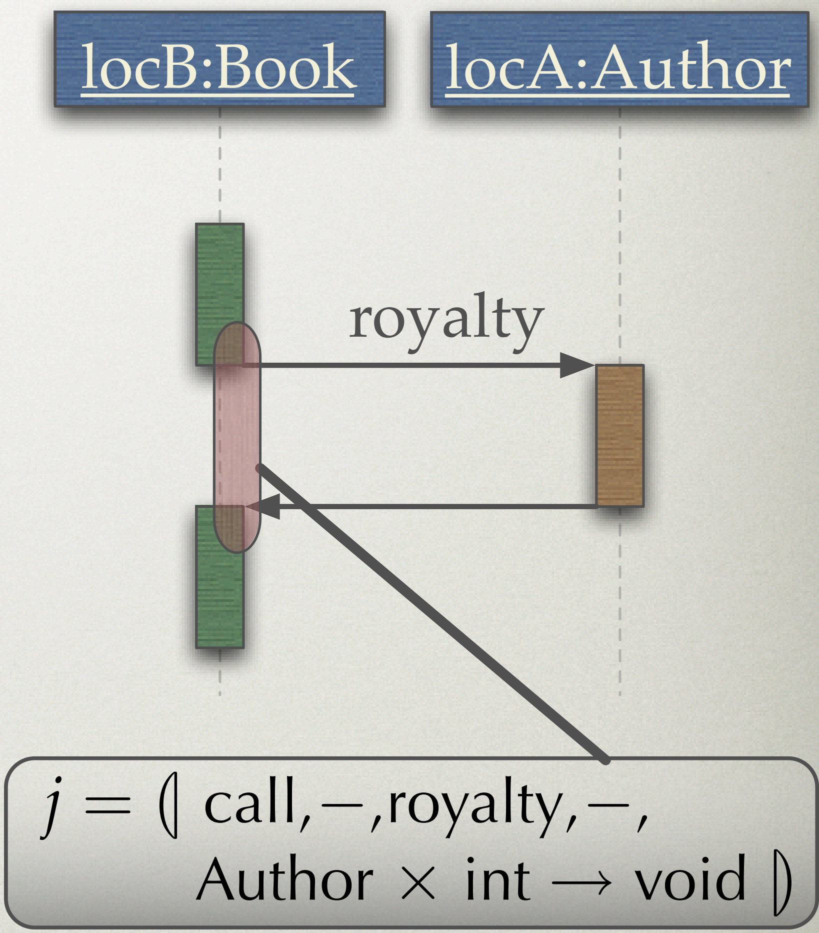
call(void royalty(..))

&& target(Author a)

&& this(Pub p)

&& args(int amt)

) =

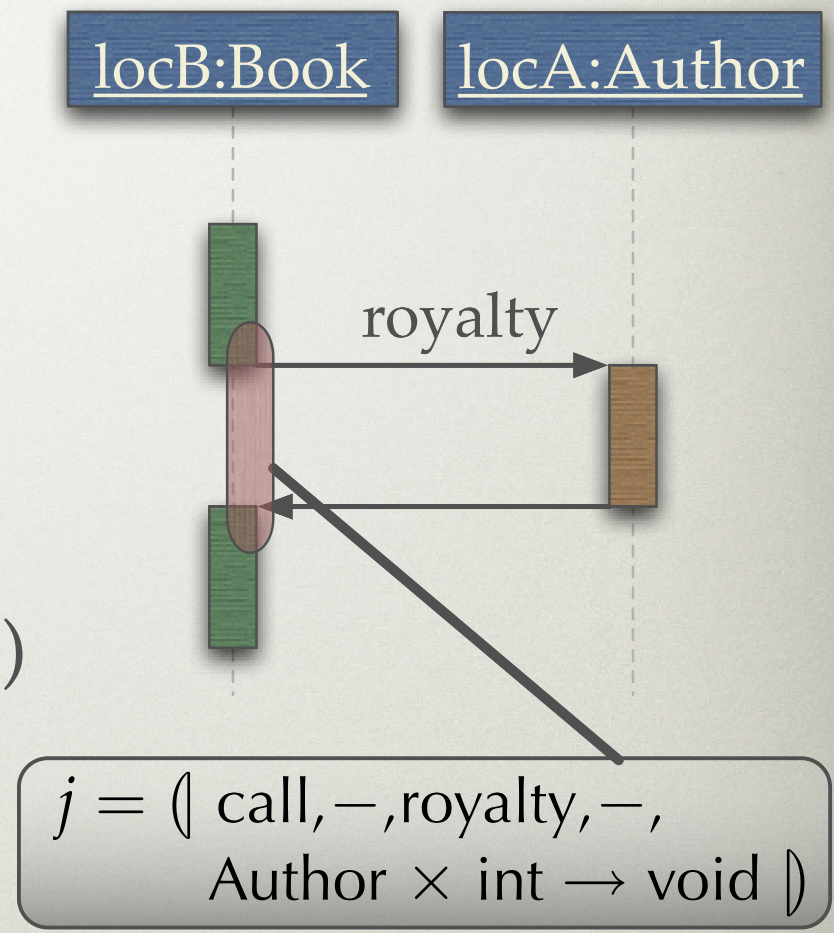


$j = \llparenthesis$ call, −, royalty, −, Author × int → void $\rrparenthesis$

Pointcut Matching

matchPCD(*j* + *J*,
call(void royalty(..))
&& target(Author a)
&& this(Pub p)
&& args(int amt)
) =

matchPCD(*j* + *J*,call(void royalty(..)))
∧ *matchPCD*(*j* + *J*,target(Author a))
∧ *matchPCD*(*j* + *J*,this(Pub p))
∧ *matchPCD*(*j* + *J*,args(int amt))



$$j = (\!|\ \text{call}, -, \text{royalty}, -, \text{Author} \times \text{int} \rightarrow \text{void}\ |\!)$$

Pointcut Matching

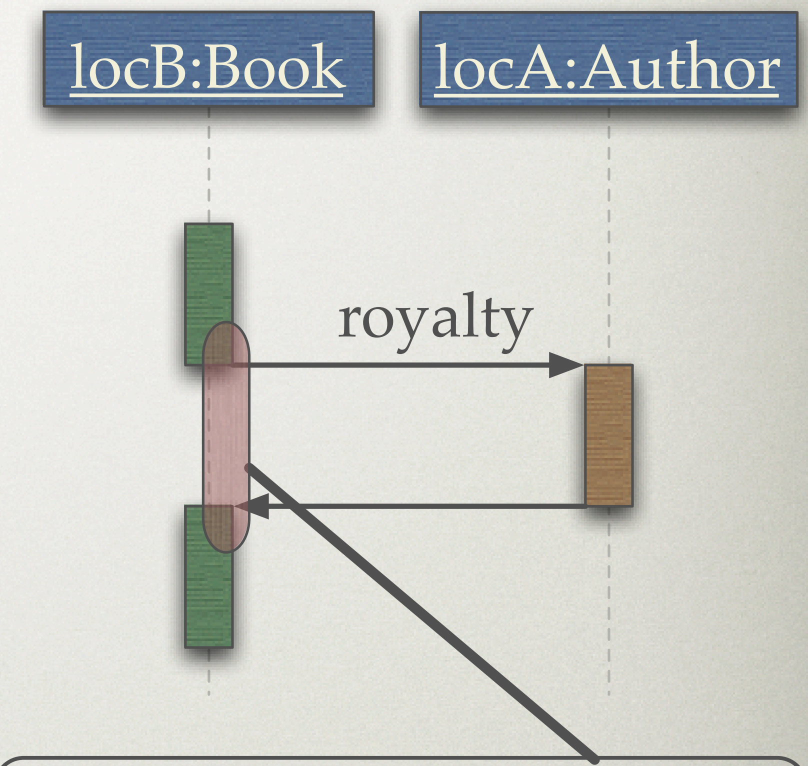
$matchPCD(j + J,$
call(void royalty(..))
&& target(Author a)
&& this(Pub p)
&& args(int amt)
$) =$

$\langle -, - \rangle$

$\wedge$ $matchPCD(j + J,$target(Author a))

$\wedge$ $matchPCD(j + J,$this(Pub p))

$\wedge$ $matchPCD(j + J,$args(int amt))



$j = ⦇$ call, $-$, royalty, $-$,
Author $\times$ int $\rightarrow$ void $⦈$

Pointcut Matching

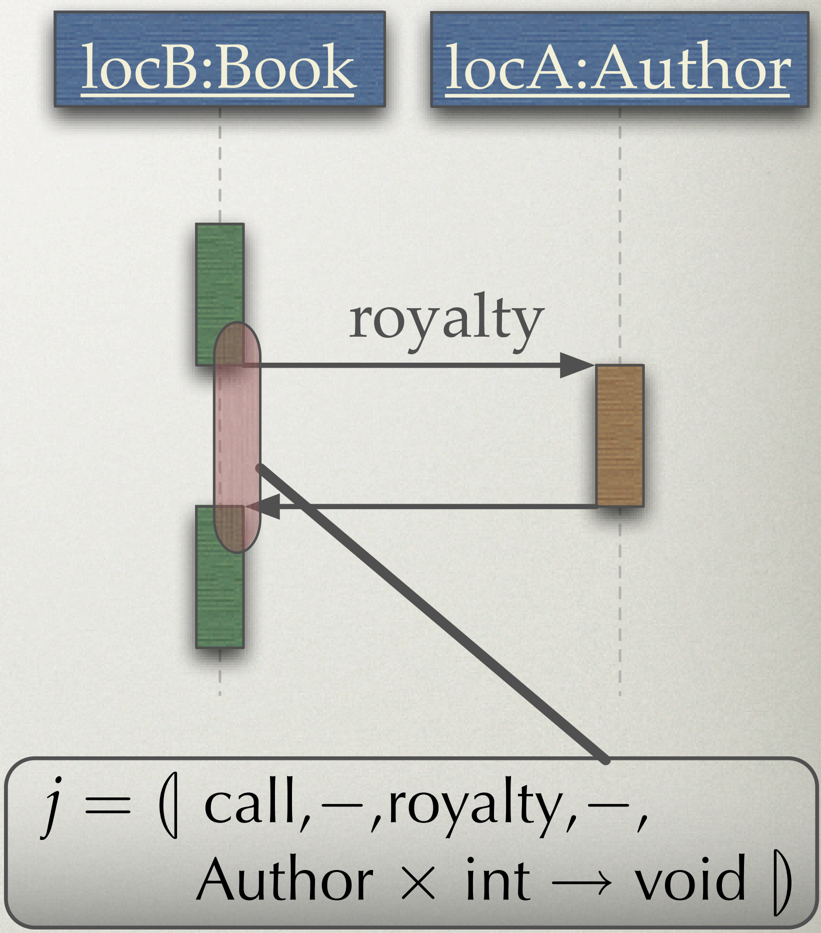
$matchPCD(j + J,$
call(void royalty(..))
&& target(Author a)
&& this(Pub p)
&& args(int amt)
$) =$

$\langle -, - \rangle$

$\wedge \langle -, \text{a} \rangle$

$\wedge$ $matchPCD(j + J,$this(Pub p))

$\wedge$ $matchPCD(j + J,$args(int amt))



$j = (\!|$ call,$-$,royalty,$-$,
Author × int → void $|\!)$

Pointcut Matching

$matchPCD(j + J,$

call(void royalty(..))

&& target(Author a)

&& this(Pub p)

&& args(int amt)

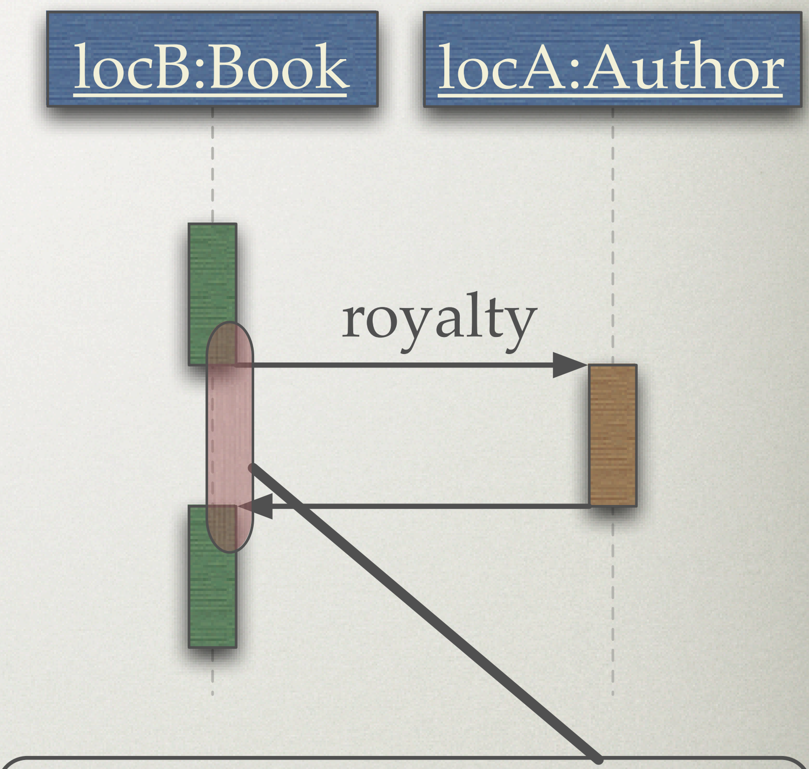
$) =$

$\langle -,- \rangle$

$\wedge \langle -, \mathrm{a} \rangle$

$\wedge \langle \mathrm{p} \mapsto \mathrm{locB}, - \rangle$

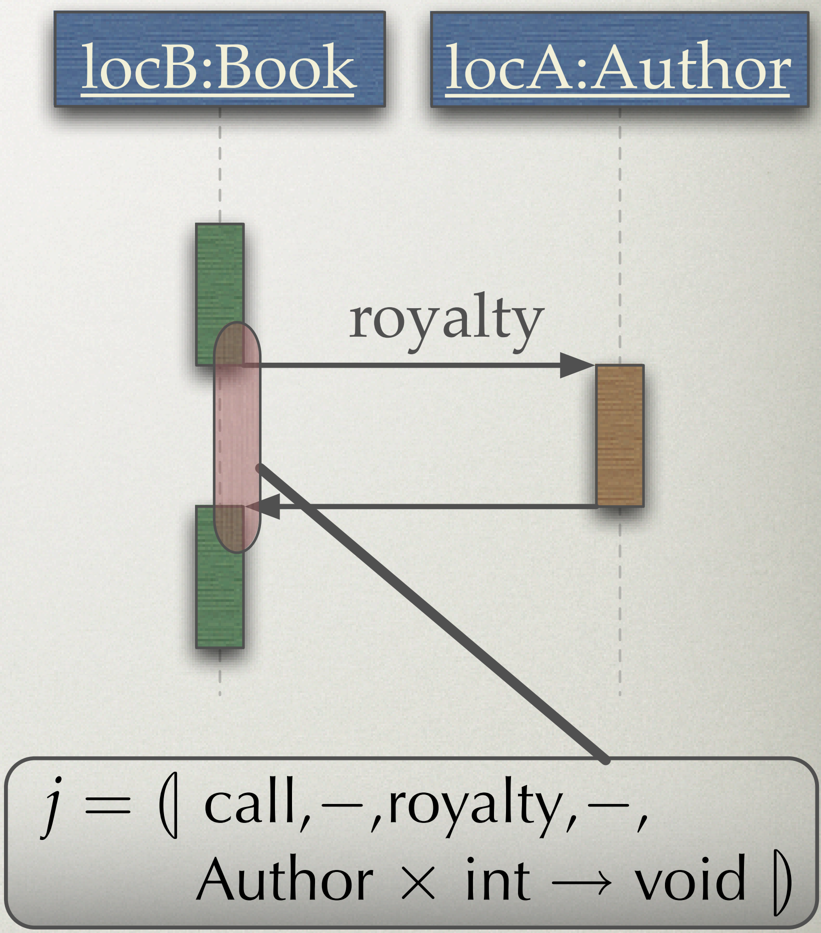
$\wedge$ $matchPCD(j + J,$args(int amt))



$j = (\!|$ call, $-$, royalty, $-$, Author $\times$ int $\rightarrow$ void $|\!)$

Pointcut Matching

matchPCD($j + J$,
  call(void royalty(..))
  && target(Author a)
  && this(Pub p)
  && args(int amt)
) $=$
  $\langle -, - \rangle$
$\wedge\ \langle -, \mathrm{a} \rangle$
$\wedge\ \langle \mathrm{p} \mapsto \mathrm{locB}, - \rangle$
$\wedge\ \langle -, -, \mathrm{amt} \rangle$



$j = \llparenthesis\ \text{call}, -, \text{royalty}, -,$
$\text{Author} \times \text{int} \rightarrow \text{void}\ \rrparenthesis$

Pointcut Matching

matchPCD(j + J,

call(void royalty(..))

&& target(Author a)

&& this(Pub p)

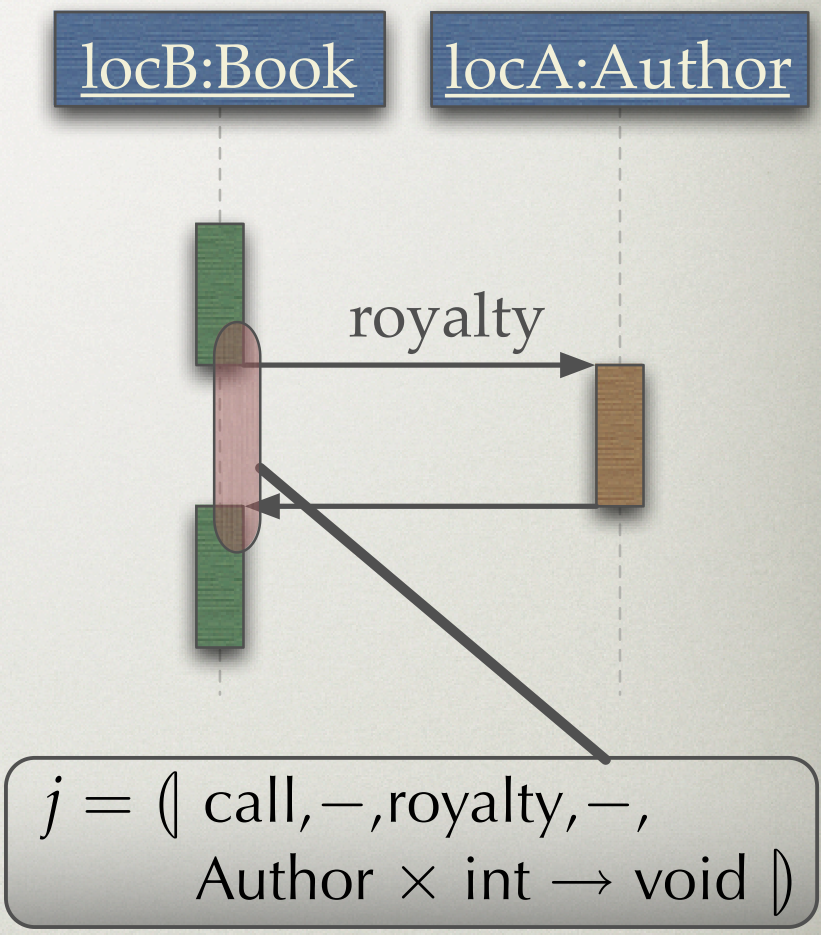
&& args(int amt)

) =

$\langle -, - \rangle$

$\wedge\ \langle -, \mathrm{a} \rangle$

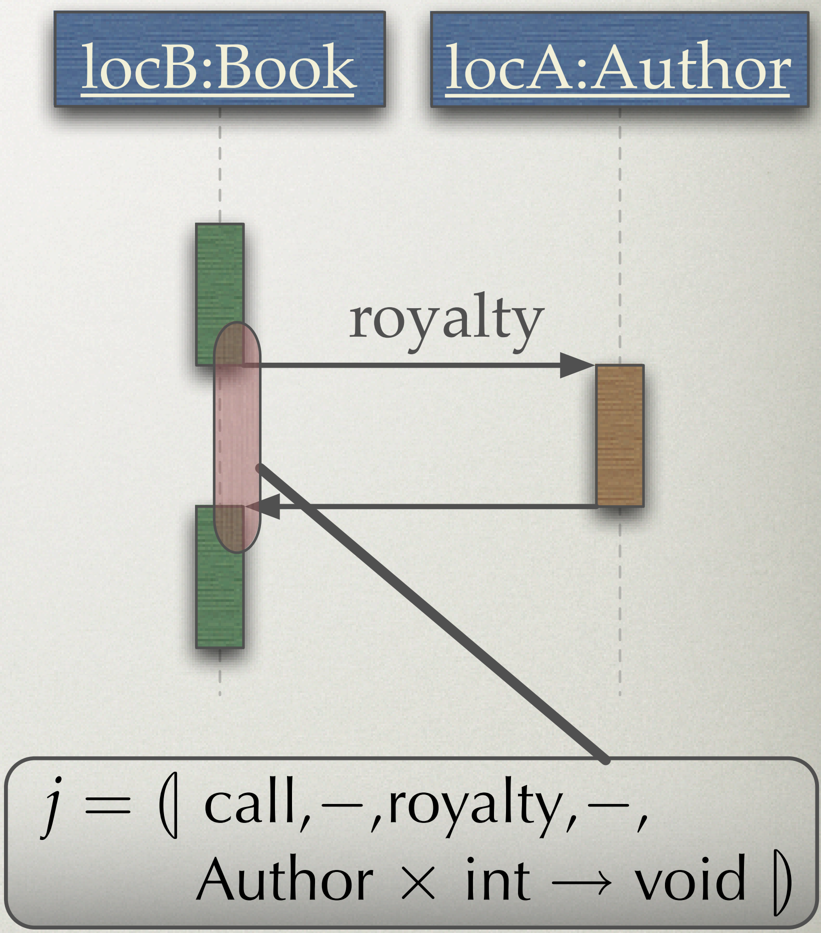
$\wedge\ \langle \mathrm{p} \mapsto \mathrm{locB}, -, \mathrm{amt} \rangle$



$$j = (\!|\ \text{call}, -, \text{royalty}, -,\ \text{Author} \times \text{int} \to \text{void}\ |\!)$$

Pointcut Matching

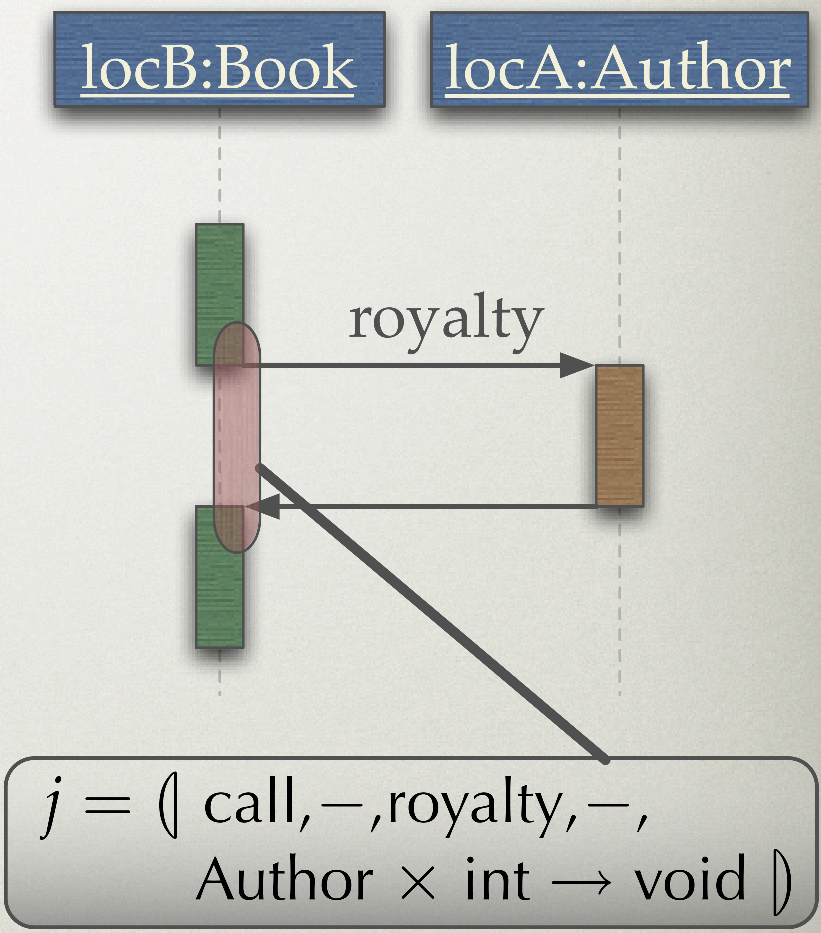
$$\begin{aligned}
&\mathit{matchPCD}(j + J,\\
&\quad \text{call(void royalty(..))}\\
&\quad \text{\&\& target(Author a)}\\
&\quad \text{\&\& this(Pub p)}\\
&\quad \text{\&\& args(int amt)}\\
&) =\\
&\quad \langle -,- \rangle\\
&\wedge \langle \text{p} \mapsto \text{locB, a, amt} \rangle
\end{aligned}$$



$$\begin{aligned} j = (\!|\ &\text{call},-,\text{royalty},-,\\ &\text{Author} \times \text{int} \rightarrow \text{void}\ |\!) \end{aligned}$$

Pointcut Matching

$$
\begin{aligned}
&\mathit{matchPCD}(j + J,\\
&\quad \text{call(void royalty(..))}\\
&\quad \text{\&\& target(Author a)}\\
&\quad \text{\&\& this(Pub p)}\\
&\quad \text{\&\& args(int amt)}\\
&) =\\
&\quad \langle \text{p} \mapsto \text{locB, a, amt} \rangle
\end{aligned}
$$



$j = (\!|\ \text{call}, -, \text{royalty}, -,\ \text{Author} \times \text{int} \rightarrow \text{void}\ |\!)$

Binding Substitution

Binding Substitution

$$e\{\!|\,\mathrm{locA}, 100 \;/\; \langle \mathrm{p} \mapsto \mathrm{locB}, \mathrm{a}, \mathrm{amt} \rangle\,|\!\}$$

Binding Substitution

$$e\{\!|\,\mathrm{locA}, 100 \;/\; \langle \boxed{\mathrm{p} \mapsto \mathrm{locB},}\ \mathrm{a}, \mathrm{amt} \rangle\,|\!\}$$

Binding Substitution

$$
\begin{aligned}
& e\{\!|\mathrm{locA},100\ /\ \langle \boxed{\mathrm{p} \mapsto \mathrm{locB},}\ \mathrm{a}, \mathrm{amt}\rangle\, |\!\} \\
= \quad & e\boxed{\{\!|\mathrm{locB}/\ \mathrm{p}|\!\}}\{\!|\mathrm{locA},100\ /\ \langle \boxed{-,}\ \mathrm{a}, \mathrm{amt}\rangle\, |\!\}
\end{aligned}
$$

Binding Substitution

$$\begin{aligned} & e\{\!|\mathrm{locA}, 100 \;/\; \langle \mathrm{p} \mapsto \mathrm{locB}, \mathrm{a}, \mathrm{amt}\rangle\, |\!\} \\ = \quad & e\{\!|\mathrm{locB} / \mathrm{p}|\!\}\{\!|\mathrm{locA}, \boxed{100} \;/\; \langle -, \mathrm{a}, \boxed{\mathrm{amt}}\rangle\, |\!\} \end{aligned}$$

Binding Substitution

$$\begin{aligned}
& e\{\!\!|\,\mathrm{locA},100 \;/\; \langle \mathrm{p} \mapsto \mathrm{locB}, \mathrm{a}, \mathrm{amt}\rangle \,|\!\!\} \\
= \quad & e\{\!\!|\,\mathrm{locB} / \mathrm{p}\,|\!\!\}\{\!\!|\,\mathrm{locA},\boxed{100} \;/\; \langle -, \mathrm{a}, \boxed{\mathrm{amt}}\rangle \,|\!\!\} \\
= \quad & e\{\!\!|\,\mathrm{locB} / \mathrm{p}\,|\!\!\}\{\!\!|\,\mathrm{locA} \;/\; \langle -, \mathrm{a}\rangle \,|\!\!\}\boxed{\{\!\!|\,100 \;/\; \mathrm{amt}\,|\!\!\}}
\end{aligned}$$

Binding Substitution

$$
\begin{aligned}
& e\{\!\!|\,\mathrm{locA}, 100 \;/\; \langle \mathrm{p} \mapsto \mathrm{locB}, \mathrm{a}, \mathrm{amt} \rangle \,|\!\!\} \\
= \quad & e\{\!\!|\,\mathrm{locB} / \mathrm{p}\,|\!\!\}\{\!\!|\,\mathrm{locA}, 100 \;/\; \langle -, \mathrm{a}, \mathrm{amt} \rangle \,|\!\!\} \\
= \quad & e\{\!\!|\,\mathrm{locB} / \mathrm{p}\,|\!\!\}\{\!\!|\,\boxed{\mathrm{locA}} \;/\; \langle -, \boxed{\mathrm{a}} \rangle \,|\!\!\}\{\!\!|\,100 \;/\; \mathrm{amt}\,|\!\!\}
\end{aligned}
$$

Binding Substitution

$$
\begin{aligned}
& e\{\!|\, \mathrm{locA}, 100 \;/\; \langle \mathrm{p} \mapsto \mathrm{locB}, \mathrm{a}, \mathrm{amt} \rangle \,|\!\} \\
= \quad & e\{\!|\, \mathrm{locB} / \mathrm{p} \,|\!\}\{\!|\, \mathrm{locA}, 100 \;/\; \langle -, \mathrm{a}, \mathrm{amt} \rangle \,|\!\} \\
= \quad & e\{\!|\, \mathrm{locB} / \mathrm{p} \,|\!\}\{\!|\, \boxed{\mathrm{locA}} \;/\; \langle -\boxed{, \mathrm{a}} \rangle \,|\!\}\{\!|\, 100 \;/\; \mathrm{amt} \,|\!\} \\
= \quad & e\{\!|\, \mathrm{locB} / \mathrm{p} \,|\!\}\boxed{\{\!|\, \mathrm{locA} \;/\; \mathrm{a} \,|\!\}}\{\!|\, 100 \;/\; \mathrm{amt} \,|\!\}
\end{aligned}
$$

Binding Substitution

$$
\begin{aligned}
& \boxed{e\{\!|\,\mathrm{locA},100 \;/\; \langle \mathrm{p} \mapsto \mathrm{locB}, \mathrm{a}, \mathrm{amt} \rangle \,|\!\}} \\
= \quad & e\{\!|\,\mathrm{locB} / \mathrm{p}\,|\!\}\{\!|\,\mathrm{locA},100 \;/\; \langle -, \mathrm{a}, \mathrm{amt} \rangle \,|\!\} \\
= \quad & e\{\!|\,\mathrm{locB} / \mathrm{p}\,|\!\}\{\!|\,\mathrm{locA} \;/\; \langle -, \mathrm{a} \rangle \,|\!\}\{\!|\,100 \;/\; \mathrm{amt}\,|\!\} \\
= \quad & \boxed{e\{\!|\,\mathrm{locB} / \mathrm{p}\,|\!\}\{\!|\,\mathrm{locA} \;/\; \mathrm{a}\,|\!\}\{\!|\,100 \;/\; \mathrm{amt}\,|\!\}}
\end{aligned}
$$

Advice Chaining

Advice Chaining

$$\langle \mathbb{E}[\text{chain } \lceil\lfloor b, loc, e, \text{␣}, \text{␣} \rfloor\rceil + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{ this}|\!\}\{\!|\overline{v} /\ b|\!\}], j + J, S\rangle$$

Advice Chaining

$$\langle \mathbb{E}[\text{chain } \llbracket b, loc, \boxed{e,} \llcorner\!\lrcorner, \llcorner\!\lrcorner \rrbracket + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } \boxed{e'} \{\!| loc / \text{this} |\!\}\{\!| \overline{v} / b |\!\}], j + J, S\rangle$$

Advice Chaining

$$\langle \mathbb{E}[\text{chain } \lceil\lfloor b, loc, e, \sqcup, \sqcup \rfloor\rceil + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{ this}|\!\}\{\!|\overline{v} / \ b|\!\}], j + J, S\rangle$$

$$\text{where } e' = \langle\!\langle e \rangle\!\rangle_{\bar{B},j}$$

Advice Chaining

$$\langle \mathbb{E}[\text{chain } \llbracket b, loc, e, \text{␣}, \text{␣} \rrbracket + \bar{B}, j(\ \overline{v}\ )], J, S \rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } e' \{\!| loc / \text{ this} |\!\} \{\!| \overline{v} / \ b |\!\}], j + J, S \rangle$$

$$\text{where } e' = \langle\!\langle e \rangle\!\rangle_{\bar{B},j}$$

$$\langle\!\langle \text{this.log(...); a.proceed(amt)} \rangle\!\rangle_{\bar{B},j}$$

Advice Chaining

$$\begin{array}{l}\langle \mathbb{E}[\text{chain } \llbracket b, loc, e, \textvisiblespace, \textvisiblespace \rrbracket + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow \\ \qquad\qquad\qquad\qquad \langle \mathbb{E}[\text{under } e' \{\!|loc / \text{ this}|\!\} \{\!|\overline{v} / \ b|\!\}], j + J, S\rangle\end{array}$$

$$\text{where } e' = \langle\!\langle e \rangle\!\rangle_{\bar{B},j}$$

$$\begin{array}{rl} & \langle\!\langle \text{this.log(...); a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\ = & \langle\!\langle \text{this.log(...)} \rangle\!\rangle_{\bar{B},j};\ \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \end{array}$$

Advice Chaining

$$\langle \mathbb{E}[\text{chain } \lceil\!\lfloor b, loc, e, \textvisiblespace, \textvisiblespace \rfloor\!\rceil + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!| loc / \text{ this} |\!\} \{\!| \overline{v} / \ b |\!\}], j + J, S\rangle$$

$$\text{where } e' = \langle\!\langle e \rangle\!\rangle_{\bar{B},j}$$

$$\begin{aligned} & \langle\!\langle \text{this.log(...); a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\ = \quad & \langle\!\langle \text{this.log(...)} \rangle\!\rangle_{\bar{B},j}\text{; } \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\ = \quad & \langle\!\langle \text{this} \rangle\!\rangle_{\bar{B},j}\text{.log(...); } \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \end{aligned}$$

Advice Chaining

$$\langle \mathbb{E}[\text{chain } \llbracket b, loc, e, \sqcup, \sqcup \rrbracket + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$
$$\langle \mathbb{E}[\text{under } e' \{\!|loc / \text{ this}|\!\} \{\!|\overline{v} / \ b|\!\}], j + J, S\rangle$$

$$\text{where } e' = \langle\!\langle e \rangle\!\rangle_{\bar{B},j}$$

$$\begin{aligned}
& \langle\!\langle \text{this.log(...); a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\
= \quad & \langle\!\langle \text{this.log(...)} \rangle\!\rangle_{\bar{B},j}; \ \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\
= \quad & \langle\!\langle \text{this} \rangle\!\rangle_{\bar{B},j}\text{.log(...); } \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\
= \quad & \text{this.log(...); } \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j}
\end{aligned}$$

Advice Chaining

$$\langle \mathbb{E}[\text{chain } \llbracket b, loc, e, \textvisiblespace, \textvisiblespace \rrbracket + \bar{B}, j(\ \overline{v}\ )], J, S\rangle \hookrightarrow$$

$$\langle \mathbb{E}[\text{under } e' \{\!| loc / \text{ this} |\!\} \{\!| \overline{v} / \ b |\!\}], j + J, S\rangle$$

$$\text{where } e' = \langle\!\langle e \rangle\!\rangle_{\bar{B},j}$$

$$\begin{aligned}
& \langle\!\langle \text{this.log(...); a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\
= \quad & \langle\!\langle \text{this.log(...)} \rangle\!\rangle_{\bar{B},j};\ \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\
= \quad & \langle\!\langle \text{this} \rangle\!\rangle_{\bar{B},j}\text{.log(...); } \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\
= \quad & \text{this.log(...); } \langle\!\langle \text{a.proceed(amt)} \rangle\!\rangle_{\bar{B},j} \\
= \quad & \text{this.log(...); chain } \bar{B}, j \text{ (a,amt)}
\end{aligned}$$

MiniMAO$_1$ Type System

Pointcut Types

$$\hat{u} \mathbin{\mathbf{.}} \hat{u}' \mathbin{\mathbf{.}} U \mathbin{\mathbf{.}} \hat{u}'' \mathbin{\mathbf{.}} V_1 \mathbin{\mathbf{.}} V_2$$

Pointcut Types

this

$$\boxed{\hat{u}} \,.\, \hat{u}' \,.\, U \,.\, \hat{u}'' \,.\, V_1 \,.\, V_2$$

Pointcut Types

$$\overset{\text{this}}{\hat{u}} \,.\, \overset{\text{target}}{\boxed{\hat{u}'}} \,.\, U \,.\, \hat{u}'' \,.\, V_1 \,.\, V_2$$

Pointcut Types

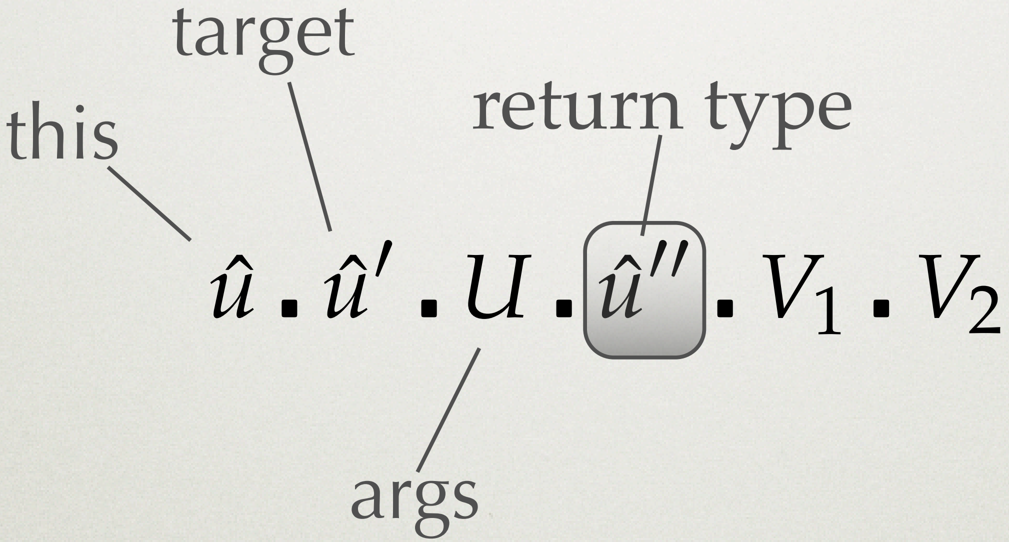
target

this

$$\hat{u} \,.\, \hat{u}' \,.\, \boxed{U} \,.\, \hat{u}'' \,.\, V_1 \,.\, V_2$$

args

Pointcut Types



target

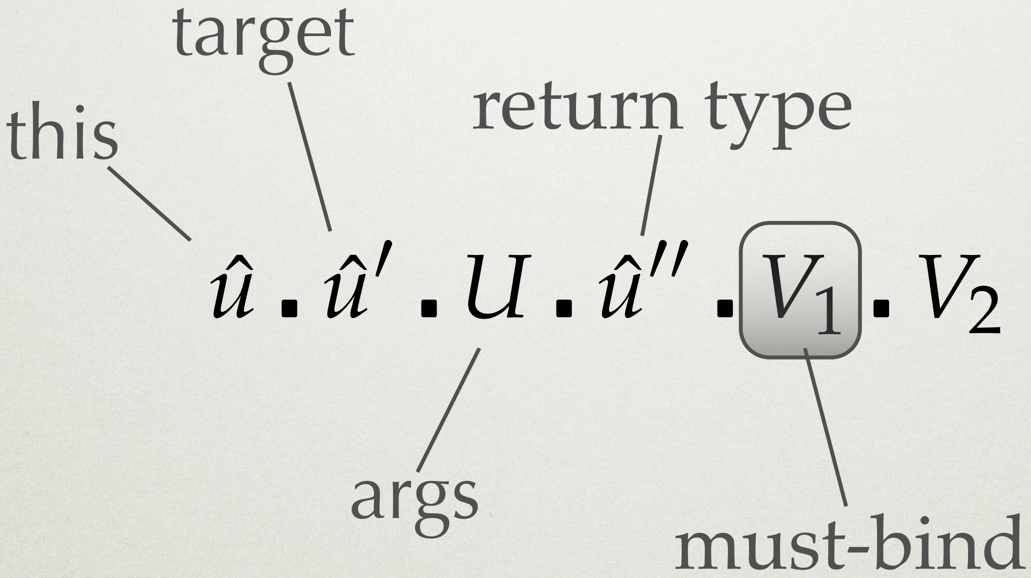
this

return type

$$\hat{u} \,.\, \hat{u}' \,.\, U \,.\, \boxed{\hat{u}''} \,.\, V_1 \,.\, V_2$$

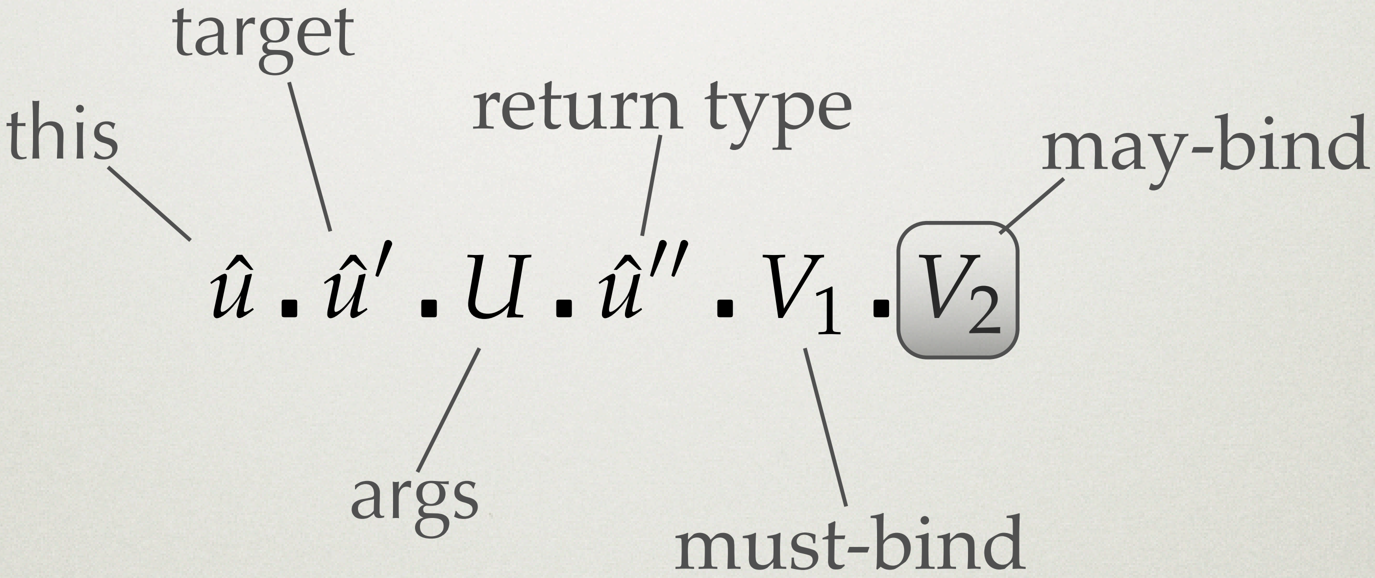
args

Pointcut Types



$$\hat{u} \,.\, \hat{u}' \,.\, U \,.\, \hat{u}'' \,.\, \boxed{V_1} \,.\, V_2$$

Pointcut Types



target

this

return type

may-bind

$$\hat{u} \mathbin{.} \hat{u}' \mathbin{.} U \mathbin{.} \hat{u}'' \mathbin{.} V_1 \mathbin{.} \boxed{V_2}$$

args

must-bind

Pointcut Types

$\Gamma \vdash$ call(void royalty(..)) && target(Author a) &&
this(Pub p) && args(int amt):
$\text{Pub}\,\mathbf{.}\,\text{Author}\,\mathbf{.}\,\langle\text{int}\rangle\,\mathbf{.}\,\text{void}\,\mathbf{.}\,\{\text{p,a,amt}\}\,\mathbf{.}\,\{\text{p,a,amt}\}$

Pointcut Types

$$\Gamma \vdash \text{call(void royalty(..))} \;\; \text{\&\& target(Author a) \&\&}$$

$$\boxed{\text{this(Pub p)}} \; \text{\&\& args(int amt):}$$

$$\boxed{\text{Pub}} \,\mathbf{.}\, \text{Author} \,\mathbf{.}\, \langle \text{int} \rangle \,\mathbf{.}\, \text{void} \,\mathbf{.}\, \{\text{p,a,amt}\} \,\mathbf{.}\, \{\text{p,a,amt}\}$$

Pointcut Types

$$\Gamma \vdash \text{call(void royalty(..))} \;\; \text{\&\&} \; \boxed{\text{target(Author a)}} \; \text{\&\&}$$

$$\text{this(Pub p) \&\& args(int amt):}$$

$$\text{Pub} \,.\, \boxed{\text{Author}} \,.\, \langle\text{int}\rangle \,.\, \text{void} \,.\, \{\text{p,a,amt}\} \,.\, \{\text{p,a,amt}\}$$

Pointcut Types

$$\begin{gathered}\Gamma \vdash \text{call(void royalty(..))} \;\;\&\&\; \text{target(Author a)} \;\&\& \\ \text{this(Pub p)} \;\&\&\; \boxed{\text{args(int amt):}} \\ \text{Pub} \;\mathbf{.}\; \text{Author} \;\mathbf{.}\; \boxed{\langle\text{int}\rangle} \;\mathbf{.}\; \text{void} \;\mathbf{.}\; \{\text{p,a,amt}\} \;\mathbf{.}\; \{\text{p,a,amt}\}\end{gathered}$$

Pointcut Types

$$\Gamma \vdash \boxed{\text{call(void royalty(..))}} \text{ \&\& target(Author a) \&\&}$$

$$\text{this(Pub p) \&\& args(int amt):}$$

$$\text{Pub} \centerdot \text{Author} \centerdot \langle\text{int}\rangle \centerdot \boxed{\text{void}} \centerdot \{\text{p,a,amt}\} \centerdot \{\text{p,a,amt}\}$$

Pointcut Types

$\Gamma \vdash$ call(void royalty(..)) && target(Author a) &&
this(Pub p) && args(int amt):

$$\text{Pub}\,\mathbf{.}\,\text{Author}\,\mathbf{.}\,\langle\text{int}\rangle\,\mathbf{.}\,\text{void}\,\mathbf{.}\,\boxed{\{\text{p,a,amt}\}\,\mathbf{.}\,\{\text{p,a,amt}\}}$$

Advice Typing Rule

T-Adv

$$\frac{\begin{array}{c} \overline{var : t} \vdash pcd : \text{\textvisiblespace} \mathbin{\blacksquare} u_0 \mathbin{\blacksquare} \langle \overline{u} \rangle \mathbin{\blacksquare} u \mathbin{\blacksquare} V \mathbin{\blacksquare} V \\ \overline{var : t}, \text{this} : a, \text{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\vdash t\ \text{around}(\ \overline{t\ var}\ ) : pcd\ \{\ e\ \}\ \text{OK in}\ a}$$

Advice Typing Rule

T-Adv
$$\frac{\begin{array}{c} \overline{var : t} \vdash pcd : \llcorner\!\lrcorner \centerdot u_0 \centerdot \langle \overline{u} \rangle \centerdot u \centerdot V \centerdot V \\ \overline{var : t}, \text{this} : a, \text{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\boxed{\vdash t \ \text{around(} \ \overline{t \ var} \ ) : \ pcd \ \{ \ e \ \} \ \text{OK in} \ a}}$$

Advice Typing Rule

T-Adv

$$\frac{\begin{array}{c}\overline{var : t} \vdash \boxed{pcd} : \textvisiblespace \mathbin{\scriptstyle\blacksquare} u_0 \mathbin{\scriptstyle\blacksquare} \langle \overline{u} \rangle \mathbin{\scriptstyle\blacksquare} u \mathbin{\scriptstyle\blacksquare} V \mathbin{\scriptstyle\blacksquare} V \\ \overline{var : t}, \text{this} : a, \text{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u\end{array}}{\vdash t \text{ around}(\ \overline{t\ var}\ ) : \boxed{pcd}\ \{\ e\ \}\ \text{ OK in } a}$$

Advice Typing Rule

$$\textsc{T-Adv}\quad \frac{\begin{array}{c} \overline{var : t} \vdash \boxed{pcd : \sqcup \mathbin{\blacksquare} u_0 \mathbin{\blacksquare} \langle \overline{u} \rangle \mathbin{\blacksquare} u \mathbin{\blacksquare} V \mathbin{\blacksquare} V} \\ \overline{var : t}, \mathrm{this} : a, \mathrm{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\vdash t\ \mathrm{around}(\ \overline{t\ var}\ ) :\ pcd\ \{\ e\ \}\ \ \mathrm{OK\ in}\ a}$$

Advice Typing Rule

T-Adv

$$\frac{\begin{array}{c} \boxed{\overline{var : t}} \vdash pcd : \textvisiblespace \centerdot u_0 \centerdot \langle \overline{u} \rangle \centerdot u \centerdot V \centerdot V \\ \overline{var : t}, \text{this} : a, \text{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\vdash t \ \text{around}(\boxed{\overline{t\ var}}) : pcd \ \{\ e\ \} \ \text{OK in } a}$$

Advice Typing Rule

$$\textsc{T-Adv}\quad \frac{\begin{array}{c} \overline{var : t} \vdash pcd : \textvisiblespace \centerdot u_0 \centerdot \langle \overline{u} \rangle \centerdot u \centerdot V \centerdot V \\ \overline{var : t}, \mathrm{this} : a, \mathrm{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash \boxed{e : s} \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\vdash t \ \mathrm{around}(\ \overline{t\ var}\ ) : \ pcd \ \{\boxed{e}\} \ \mathrm{OK\ in}\ a}$$

Advice Typing Rule

T-Adv

$$\frac{\begin{array}{c} \overline{var : t} \vdash pcd : \text{\textvisiblespace} \centerdot u_0 \centerdot \langle \overline{u} \rangle \centerdot u \centerdot V \centerdot V \\ \overline{var : t}, \mathrm{this} : a, \mathrm{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : \boxed{s} \\ V = \{\overline{var}\} \qquad \boxed{s \preccurlyeq t} \preccurlyeq u \end{array}}{\vdash \boxed{t}\ \mathrm{around}(\ \overline{t\ var}\ ) :\ pcd\ \{\ e\ \}\ \ \mathrm{OK\ in}\ a}$$

Advice Typing Rule

T-Adv

$$\dfrac{\begin{array}{c} \overline{var : t} \vdash pcd : \text{␣} \centerdot u_0 \centerdot \langle\overline{u}\rangle \centerdot u \centerdot V \centerdot V \\ \boxed{\overline{var : t}}, \mathrm{this} : a, \mathrm{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\vdash t\ \mathrm{around}(\boxed{\overline{t\ var}}) : pcd\ \{\ e\ \}\ \ \mathrm{OK\ in}\ a}$$

Advice Typing Rule

$$\textsc{T-Adv}\quad \frac{\begin{array}{c} \overline{var : t} \vdash pcd : \text{\textvisiblespace} \,\blacksquare\, u_0 \,\blacksquare\, \langle \overline{u} \rangle \,\blacksquare\, u \,\blacksquare\, V \,\blacksquare\, V \\ \overline{var : t}, \boxed{\mathrm{this} : a}, \mathrm{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\vdash t\ \mathrm{around}(\ \overline{t\ var}\ ) : pcd\ \{\ e\ \}\ \ \mathrm{OK\ in}\ \boxed{a}}$$

Advice Typing Rule

T-Adv

$$\frac{\begin{array}{c}\overline{var : t} \vdash pcd : \textvisiblespace \centerdot \boxed{u_0 \centerdot \langle \overline{u} \rangle \centerdot u} \centerdot V \centerdot V \\ \overline{var : t}, \text{this} : a, \boxed{\text{proceed} : (u_0 \times \overline{u} \longrightarrow u)} \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq t \preccurlyeq u\end{array}}{\vdash t \text{ around(} \overline{t\ var} \text{) : } pcd \text{ \{ } e \text{ \} OK in } a}$$

Advice Typing Rule

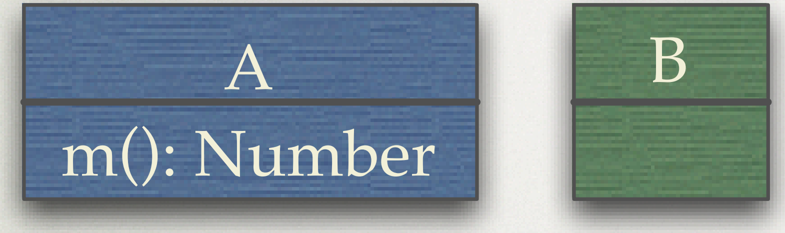
$$\textsc{T-Adv}\ \dfrac{\begin{array}{c} \overline{var : t} \vdash pcd : \textvisiblespace \blacksquare u_0 \blacksquare \langle \overline{u} \rangle \blacksquare u \boxed{\blacksquare V \blacksquare V} \\ \overline{var : t}, \mathrm{this} : a, \mathrm{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ \boxed{V = \{\overline{var}\}} \qquad s \preccurlyeq t \preccurlyeq u \end{array}}{\vdash t\ \mathrm{around}(\ \overline{t\ \boxed{var}}\ ) : pcd\ \{\ e\ \}\ \ \mathrm{OK\ in}\ a}$$

Advice Typing Rule

T-Adv

$$\frac{\begin{array}{c} \overline{var : t} \vdash pcd : \textvisiblespace \centerdot u_0 \centerdot \langle \overline{u} \rangle \centerdot \boxed{u} \centerdot V \centerdot V \\ \overline{var : t}, \text{this} : a, \text{proceed} : (u_0 \times \overline{u} \longrightarrow u) \vdash e : s \\ V = \{\overline{var}\} \qquad s \preccurlyeq \boxed{t \preccurlyeq u} \end{array}}{\vdash \boxed{t}\ \text{around}(\ \overline{t\ var}\ ) :\ pcd\ \{\ e\ \}\ \text{OK in}\ a}$$

Invariant Target Matching

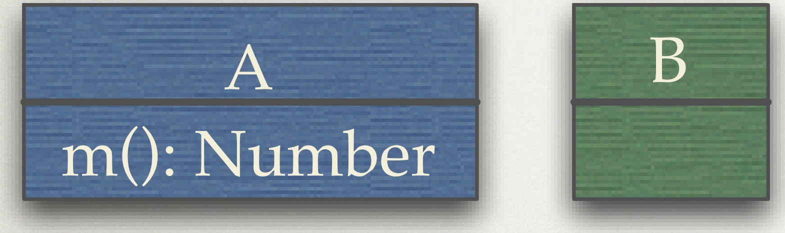


A

m(): Number

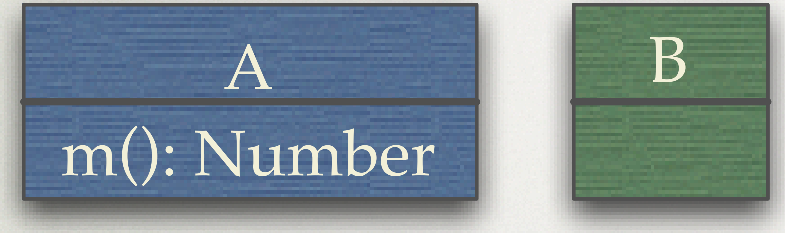
B

Invariant Target Matching



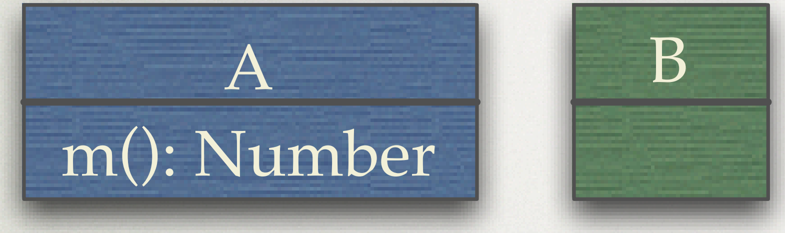
```
Number around(Object t) :
    call(Number m(..))
    && target(Object t)
{
    new B().proceed()
}
```

Invariant Target Matching



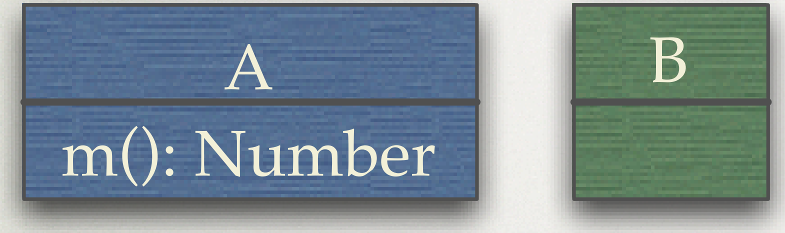
```
Number around(Object t) :
    call(Number m(..))
    && target(Object t)
{
    new B().proceed()
}
```

Invariant Target Matching



```
Number around(Object t) :
    call(Number m(..))
    && target(Object t)
{
    new B().proceed()
}
```

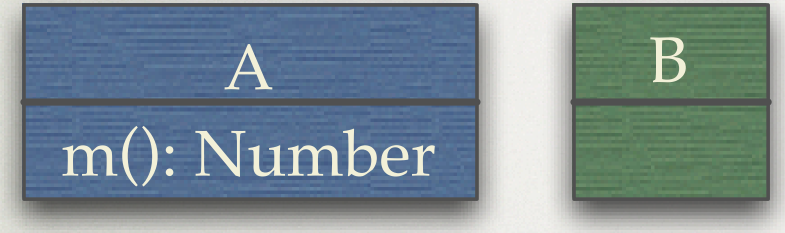
Invariant Target Matching



```
Number around(Object t) :
    call(Number m(..))
    && target(Object t)
{
    new B().proceed()
}
```

- AspectJ
  - Matches A.m(), A <: Object
  - Fails at runtime

Invariant Target Matching



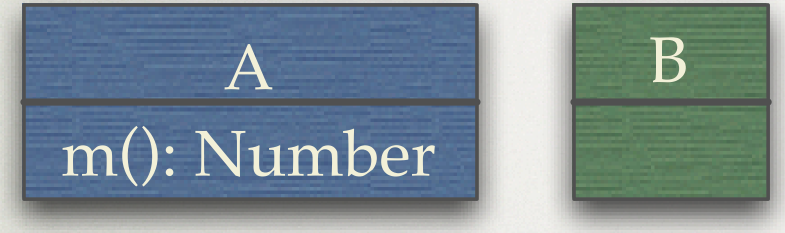
```
Number around(Object t) :
    call(Number m(..))
    && target(Object t)
{
    new B().proceed()
}
```

- AspectJ
  - Matches A.m(), A <: Object
  - Fails at runtime
- MiniMAO$_1$
  - Does not match, A ≠ Object

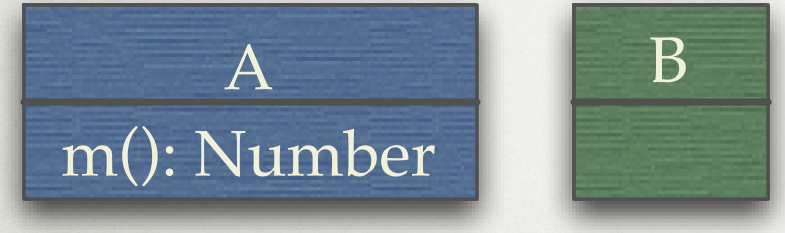
Alternatives to Invariant Target Matching

- Subtype matching without allowing target changes
  - Two forms of target pointcut?
- Global typechecking

Advice Return Type

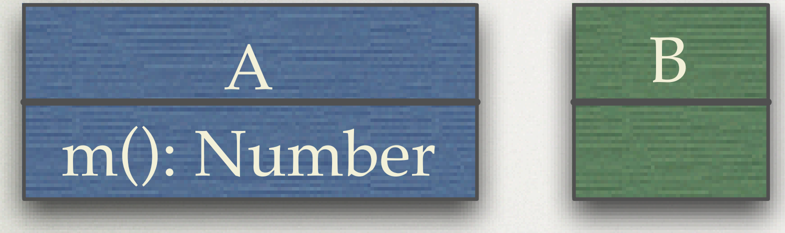


Advice Return Type



```
Number m() { return new Float(0.0); }
```

Advice Return Type

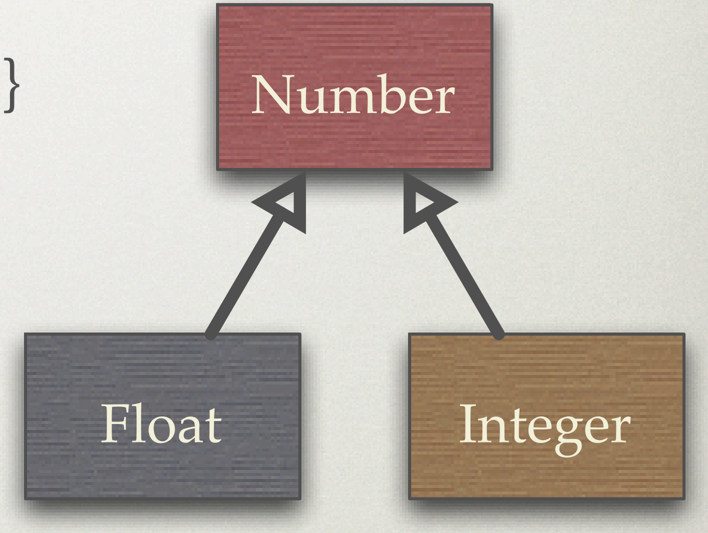


A

m(): Number

B

```
Number m() { return new Float(0.0); }
```

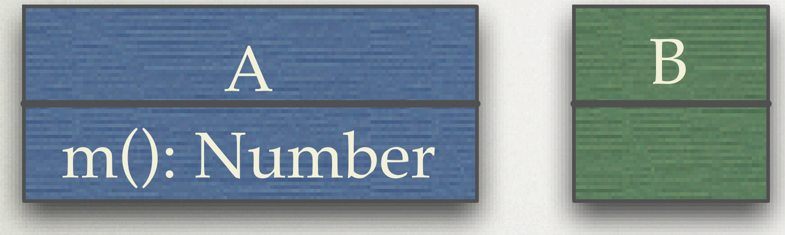


Number

Float

Integer

Advice Return Type

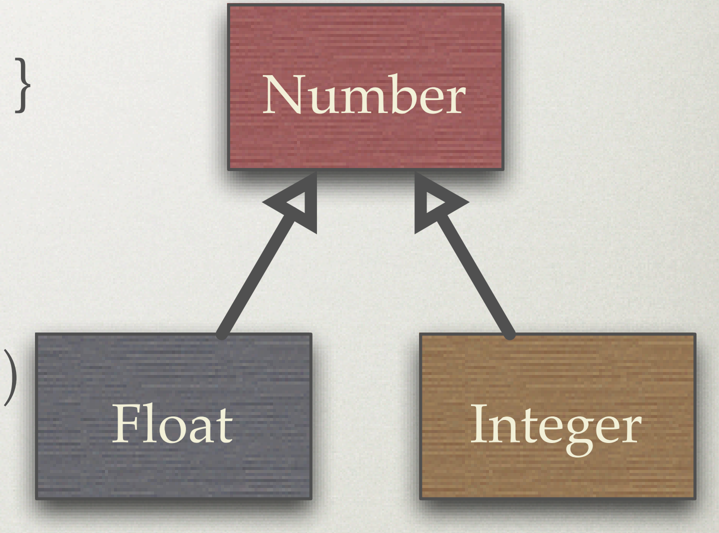


A

m(): Number

B

```
Number m() { return new Float(0.0); }
```



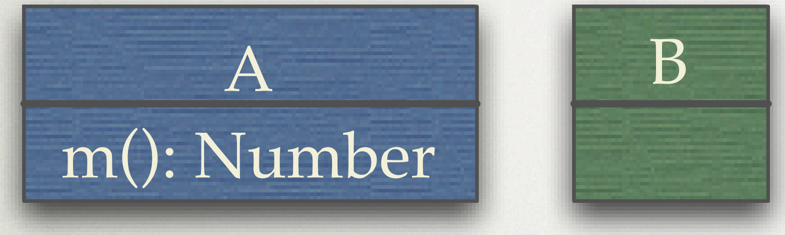
Number

Float

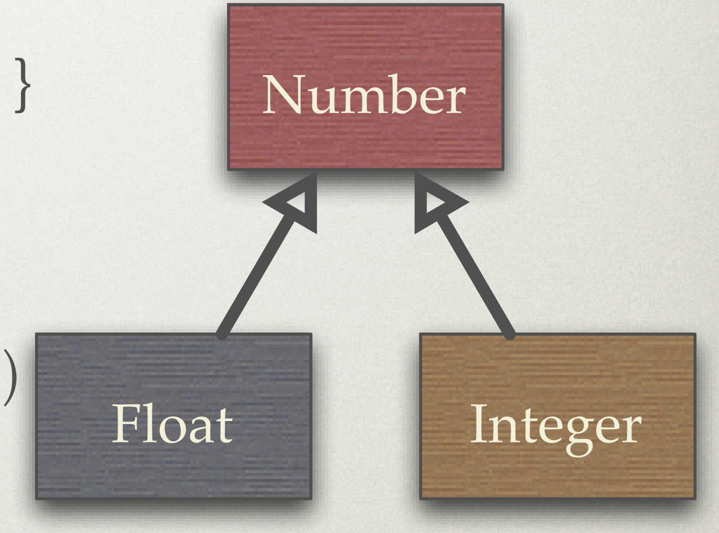
Integer

```
Integer around(A a) :
    call(Number m(..)) && target(A a)
{
    Integer i = a.proceed();
    …
}
```

Advice Return Type

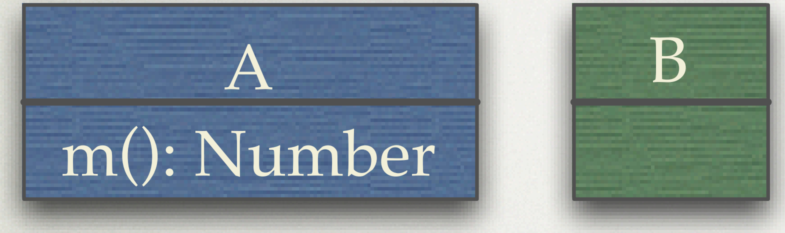


```
Number m() { return new Float(0.0); }
```



```
Integer around(A a) :
    call(Number m(..)) && target(A a)
{
   Integer i = a.proceed();
    …
}
```

Advice Return Type



A

m(): Number

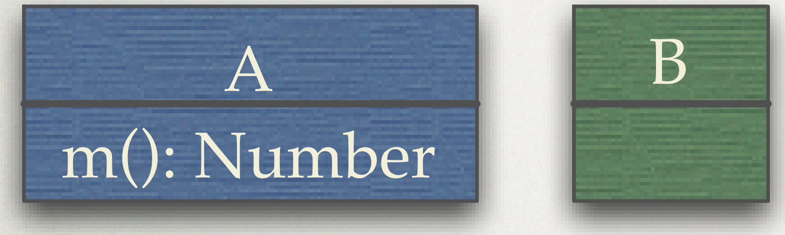
B

```
Number m() { return new Float(0.0); }


Integer around(A a) :
     call(Number m(..)) && target(A a)
{
     Integer i = a.proceed();
     ...
}
```

- AspectJ
  - Typechecks
  - Matches A.m()
  - Fails at runtime

Advice Return Type

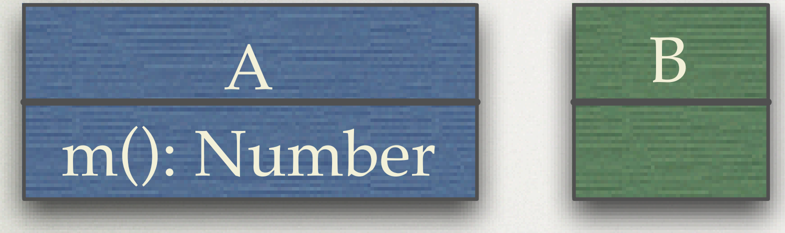


```
Number m() { return new Float(0.0); }


Integer around(A a) :
    call(Number m(..)) && target(A a)
{
   Integer i = a.proceed();
    …
}
```

- AspectJ
  - Typechecks
  - Matches A.m()
  - Fails at runtime
- MiniMAO$_1$
  - Does not typecheck

Advice Return Type



```
Number m() { return new Float(0.0); }

Integer around(A a) :
    call(Number m(..)) && target(A a)
{
   Integer i = a.proceed();
    …
}
```

- AspectJ
  - Typechecks
  - Matches A.m()
  - Fails at runtime
- MiniMAO$_1$
  - Does not typecheck

Alternative to Return Type in Pointcut

- Calculate least upper bound on return types from pointcut
  - Object if not specified
  - Might require casts

Type System Soundness Results

Soundness Theorem

Given a program $P = decl_1 \ldots decl_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = loc$ and $loc \in dom(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $\boxed{P = decl_1 \ldots decl_n \; e,}$ with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = loc$ and $loc \in dom(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = \mathit{decl}_1 \ldots \mathit{decl}_n\ e$, with $\boxed{\vdash P \text{ OK}}$, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = \mathit{loc}$ and $\mathit{loc} \in \mathit{dom}(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = \mathit{decl}_1 \ldots \mathit{decl}_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = \mathit{loc}$ and $\mathit{loc} \in \mathit{dom}(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = \mathit{decl}_1 \ldots \mathit{decl}_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = \mathit{loc}$ and $\mathit{loc} \in \mathit{dom}(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = \mathit{decl}_1 \ldots \mathit{decl}_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\boxed{\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle}$ and either:

- $v = \mathit{loc}$ and $\mathit{loc} \in \mathit{dom}(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = decl_1 \ldots decl_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0\rangle \overset{*}{\hookrightarrow} \langle v, J, S\rangle$ and either:

- $v = loc$ and $loc \in dom(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = decl_1 \ldots decl_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = loc$ and $loc \in dom(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = \mathit{decl}_1 \ldots \mathit{decl}_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = \mathit{loc}$ and $\mathit{loc} \in \mathit{dom}(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Soundness Theorem

Given a program $P = decl_1 \ldots decl_n\ e$, with $\vdash P$ OK, and a valid store S_0, then either the evaluation of e diverges or else $\langle e, \bullet, S_0 \rangle \overset{*}{\hookrightarrow} \langle v, J, S \rangle$ and either:

- $v = loc$ and $loc \in dom(S)$,
- $v =$ null,
- $v =$ NullPointerException, or
- $v =$ ClassCastException

Contributions of MiniMAO$_1$

Contributions of MiniMAO$_1$

- Effect of advice on method selection

Contributions of MiniMAO$_1$

- Effect of advice on method selection
- Primitive operations: Call and Exec

Contributions of MiniMAO$_1$

- Effect of advice on method selection
- Primitive operations: Call and Exec
- Call → Call$_A$, Bind, Advise, Call$_B$

Contributions of MiniMAO$_1$

- Effect of advice on method selection
- Primitive operations: Call and Exec
- Call → Call$_A$, Bind, Advise, Call$_B$
- Sound, static type system

Related Work

- Bruns, et al. (Concur04)
- Dantas and Walker (FOOL05)
- Douence, Motelet, Südholt (Refl01)
- Jagadeesan, Jeffrey, and Riely (ECOOP03)
- Masuhara and Kiczales (ECOOP03)
- Walker, Zdancewic, and Ligatti (ICFP03)
- Wand, Kiczales, and Dutchyn (TOPLAS04)

Conclusions and Future Work

Main Contributions

Main Contributions

- General technique for adding aspects to a core language

Main Contributions

- General technique for adding aspects to a core language
- Sound, static type system

Main Contributions

- General technique for adding aspects to a core language
- Sound, static type system
- Effect of advice on method selection when changing target

Next Steps

Next Steps

- Types for separation of concerns:
    - Concern domains type system

Next Steps

- Types for separation of concerns:
  - Concern domains type system
- Reason within separate concerns:
  - Pre- and post-conditions
  - Implicit under-specification

Next Steps

- Types for separation of concerns:
  - Concern domains type system
- Reason within separate concerns:
  - Pre- and post-conditions
  - Implicit under-specification
- Investigate practical implications

Questions?