HARVARD

School of Engineering
and Applied Sciences

Control-Flow Analysis

CS252r Spring 2011

Includes (a lot of) material from slides for
Principles of Program Analysis
by Nielson, Nielson, and Hankin

http://www2.imm.dtu.dk/~riis/PPA/ppasup2004.html

Outline

- What’s the problem?
- 0-CFA
- Uniform k-CFA
- The *k*-CFA paradox

What is control-flow analysis?

- Data-flow analysis relied on a **control-flow graph**
- How do we construct CFG?
- For intra-procedural analysis, relatively straightforward
  - Identify basic blocks, control-flow structures
    - We will not delve into this
- For inter-procedural analysis
  - If functions/procedures are not first-class, relatively simple
  - For languages with **dynamic dispatch**, it's harder
    - Dynamic dispatch: which procedure/function gets invoked depends on runtime values
    - Functional languages, OO, imperative languages with procedures as parameters, ...

CFA in higher-order languages

- We’ll mostly focus today on control-flow analysis of functional languages
    - For each function application, which functions may be applied?
- E.g.

```
let f = fn x => x 1;
    g = fn y => y+2;
    h = fn z => z+3
in (f g) + (f h)
```

Syntax of language

e	$\in$	**Exp**	expressions (or labelled terms)
t	$\in$	**Term**	terms (or unlabelled expressions)
f, x	$\in$	**Var**	variables
c	$\in$	**Const**	constants
op	$\in$	**Op**	binary operators
ℓ	$\in$	**Lab**	labels

$$e \ ::= \ t^{\ell}$$

$$\begin{array}{lll} t & ::= & c \mid x \mid \texttt{fn}\ x\ \texttt{=>}\ e_0 \mid \texttt{fun}\ f\ x\ \texttt{=>}\ e_0 \mid e_1\ e_2 \\ & \mid & \texttt{if}\ e_0\ \texttt{then}\ e_1\ \texttt{else}\ e_2 \mid \texttt{let}\ x\ \texttt{=}\ e_1\ \texttt{in}\ e_2 \mid e_1\ \mathit{op}\ e_2 \end{array}$$

Examples

$$\texttt{((fn x => x}^{1}\texttt{)}^{2}\texttt{ (fn y => y}^{3}\texttt{)}^{4}\texttt{)}^{5}$$

$$\begin{array}{l}
\texttt{(let f = (fn x => (x}^{1}\texttt{ 1}^{2}\texttt{)}^{3}\texttt{)}^{4}\texttt{;} \\
\texttt{ in (let g = (fn y => y}^{5}\texttt{)}^{6}\texttt{;} \\
\texttt{~~~~~in (let h = (fn z => z}^{7}\texttt{)}^{8} \\
\texttt{~~~~~~~~~~in ((f}^{9}\texttt{ g}^{10}\texttt{)}^{11}\texttt{ + (f}^{12}\texttt{ h}^{13}\texttt{)}^{14}\texttt{)}^{15}\texttt{)}^{16}\texttt{)}^{17}\texttt{)}^{18}
\end{array}$$

$$\begin{array}{l}
\texttt{(let g = (fun f x => (f}^{1}\texttt{ (fn y => y}^{2}\texttt{)}^{3}\texttt{)}^{4}\texttt{)}^{5} \\
\texttt{ in (g}^{6}\texttt{ (fn z => z}^{7}\texttt{)}^{8}\texttt{)}^{9}\texttt{)}^{10}
\end{array}$$

0-CFA

- 0-CFA is an context-insensitive CFA.
- Result of a 0-CFA analysis is pair (Ĉ, ^ρ)
  - Ĉ is an **abstract cache**
  - ^ρ is an **abstract environment**

Actually, just functions

$$\begin{array}{llllll} \hat{v} & \in & \widehat{\textbf{Val}} & = & \mathcal{P}(\textbf{Term}) & \textit{abstract values} \\ \hat{\rho} & \in & \widehat{\textbf{Env}} & = & \textbf{Var} \to \widehat{\textbf{Val}} & \textit{abstract environments} \\ \hat{C} & \in & \widehat{\textbf{Cache}} & = & \textbf{Lab} \to \widehat{\textbf{Val}} & \textit{abstract caches} \end{array}$$

- Notes:
  - Could combine these into one entity: (Var ∪ Lab) → ^Val
  - Could also require A-normal form, where all subterms are appropriately labeled by variables

Example

$((\texttt{fn x => x}^{1})^{2}\ (\texttt{fn y => y}^{3})^{4})^{5}$

	$(\widehat{\mathsf{C}}_{\mathsf{e}}, \widehat{\rho}_{\mathsf{e}})$	$(\widehat{\mathsf{C}}'_{\mathsf{e}}, \widehat{\rho}'_{\mathsf{e}})$	$(\widehat{\mathsf{C}}''_{\mathsf{e}}, \widehat{\rho}''_{\mathsf{e}})$
1	$\{\texttt{fn y => y}^{3}\}$	$\{\texttt{fn y => y}^{3}\}$	$\{\texttt{fn x => x}^{1}, \texttt{fn y => y}^{3}\}$
2	$\{\texttt{fn x => x}^{1}\}$	$\{\texttt{fn x => x}^{1}\}$	$\{\texttt{fn x => x}^{1}, \texttt{fn y => y}^{3}\}$
3	$\emptyset$	$\emptyset$	$\{\texttt{fn x => x}^{1}, \texttt{fn y => y}^{3}\}$
4	$\{\texttt{fn y => y}^{3}\}$	$\{\texttt{fn y => y}^{3}\}$	$\{\texttt{fn x => x}^{1}, \texttt{fn y => y}^{3}\}$
5	$\{\texttt{fn y => y}^{3}\}$	$\{\texttt{fn y => y}^{3}\}$	$\{\texttt{fn x => x}^{1}, \texttt{fn y => y}^{3}\}$
x	$\{\texttt{fn y => y}^{3}\}$	$\emptyset$	$\{\texttt{fn x => x}^{1}, \texttt{fn y => y}^{3}\}$
y	$\emptyset$	$\emptyset$	$\{\texttt{fn x => x}^{1}, \texttt{fn y => y}^{3}\}$
	Acceptable	***Not acceptable***	***Acceptable but less precise***

Abstract specification

- What does it mean for (Ĉ, ^ρ) to be acceptable?
- Define relation indicating when (Ĉ, ^ρ) is acceptable 0-CFA of expression e

$$\boxed{(\widehat{\mathsf{C}}, \widehat{\rho}) \models e}$$

$$\models \;:\; (\widehat{\mathbf{Cache}} \times \widehat{\mathbf{Env}} \times \mathbf{Exp}) \rightarrow \{\mathit{true}, \mathit{false}\}$$

Abstract specification

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models c^{\ell}$ always

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models x^{\ell}$ $\underline{\text{iff}}$ $\widehat{\rho}(x) \subseteq \widehat{\mathsf{C}}(\ell)$

$$\begin{aligned}(\widehat{\mathsf{C}}, \widehat{\rho}) &\models (\texttt{let}\ x\ \texttt{=}\ t_1^{\ell_1}\ \texttt{in}\ t_2^{\ell_2})^{\ell} \\ \underline{\text{iff}}\quad & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_1^{\ell_1} \wedge (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_2^{\ell_2} \wedge \\ & \widehat{\mathsf{C}}(\ell_1) \subseteq \widehat{\rho}(x) \wedge \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\mathsf{C}}(\ell)\end{aligned}$$

Abstract specification

$$
\begin{array}{lll}
(\widehat{\mathsf{C}}, \widehat{\rho}) \models (\texttt{if } t_0^{\ell_0} \texttt{ then } t_1^{\ell_1} \texttt{ else } t_2^{\ell_2})^{\ell} & & \\
\quad \underline{\text{iff}} & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_0^{\ell_0} \; \wedge & \\
 & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_1^{\ell_1} \; \wedge \; (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_2^{\ell_2} \wedge & \\
 & \widehat{\mathsf{C}}(\ell_1) \subseteq \widehat{\mathsf{C}}(\ell) \; \wedge \; \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\mathsf{C}}(\ell) &
\end{array}
$$

$$
\begin{array}{ll}
(\widehat{\mathsf{C}}, \widehat{\rho}) \models (t_1^{\ell_1} \; op \; t_2^{\ell_2})^{\ell} & \\
\quad \underline{\text{iff}} & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_1^{\ell_1} \; \wedge \; (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_2^{\ell_2}
\end{array}
$$

Abstract specification

$$(\widehat{\mathsf{C}}, \widehat{\rho}) \models (\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0})^{\ell} \text{ iff } \{\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}\} \subseteq \widehat{\mathsf{C}}(\ell)$$

$$\begin{aligned}
&(\widehat{\mathsf{C}}, \widehat{\rho}) \models (t_1^{\ell_1}\ t_2^{\ell_2})^{\ell} \\
&\quad \underline{\text{iff}} \quad (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_1^{\ell_1} \ \wedge\ (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_2^{\ell_2} \ \wedge \\
&\qquad\quad (\forall (\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \widehat{\mathsf{C}}(\ell_1) : \quad (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_0^{\ell_0} \ \wedge \\
&\qquad\qquad \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\rho}(x) \ \wedge\ \widehat{\mathsf{C}}(\ell_0) \subseteq \widehat{\mathsf{C}}(\ell))
\end{aligned}$$

Abstract specification

$$(\widehat{\mathsf{C}}, \widehat{\rho}) \models (\texttt{fun}\ f\ x\ \texttt{=>}\ e_0)^\ell \text{ iff } \{\texttt{fun}\ f\ x\ \texttt{=>}\ e_0\} \subseteq \widehat{\mathsf{C}}(\ell)$$

$$(\widehat{\mathsf{C}}, \widehat{\rho}) \models (t_1^{\ell_1}\ t_2^{\ell_2})^\ell$$

$$\begin{array}{lll}
\underline{\text{iff}} & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_1^{\ell_1} \ \wedge\ (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_2^{\ell_2} \ \wedge & \\
& (\forall (\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \widehat{\mathsf{C}}(\ell_1) : & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_0^{\ell_0} \ \wedge \\
& & \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\rho}(x) \ \wedge\ \widehat{\mathsf{C}}(\ell_0) \subseteq \widehat{\mathsf{C}}(\ell)) \ \wedge \\
& (\forall (\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \widehat{\mathsf{C}}(\ell_1) : & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_0^{\ell_0} \ \wedge \\
& & \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\rho}(x) \ \wedge\ \widehat{\mathsf{C}}(\ell_0) \subseteq \widehat{\mathsf{C}}(\ell) \ \wedge \\
& & \{\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}\} \subseteq \widehat{\rho}(f))
\end{array}$$

What's acceptable

$((\texttt{fn x => x}^1)^2\ (\texttt{fn y => y}^3)^4)^5$

	$(\widehat{C}_e, \widehat{\rho}_e)$	$(\widehat{C}'_e, \widehat{\rho}'_e)$
1	$\{\texttt{fn y => y}^3\}$	$\{\texttt{fn y => y}^3\}$
2	$\{\texttt{fn x => x}^1\}$	$\{\texttt{fn x => x}^1\}$
3	$\emptyset$	$\emptyset$
4	$\{\texttt{fn y => y}^3\}$	$\{\texttt{fn y => y}^3\}$
5	$\{\texttt{fn y => y}^3\}$	$\{\texttt{fn y => y}^3\}$
x	$\{\texttt{fn y => y}^3\}$	$\emptyset$
y	$\emptyset$	$\emptyset$

$(\widehat{C}_e, \widehat{\rho}_e) \models ((\texttt{fn x => x}^1)^2\ (\texttt{fn y => y}^3)^4)^5$

$(\widehat{C}'_e, \widehat{\rho}'_e) \not\models ((\texttt{fn x => x}^1)^2\ (\texttt{fn y => y}^3)^4)^5$

Abstract specification

- Note that we can't define ⊨ by structural induction on expressions

$$
\begin{array}{lll}
(\widehat{\mathsf{C}}, \widehat{\rho}) \models (t_1^{\ell_1}\ t_2^{\ell_2})^{\ell} & & \\
\quad \underline{\text{iff}} & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_1^{\ell_1} \ \land\ (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_2^{\ell_2} \ \land & \\
 & (\forall(\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \widehat{\mathsf{C}}(\ell_1) : & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_0^{\ell_0} \ \land \\
 & & \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\rho}(x) \ \land\ \widehat{\mathsf{C}}(\ell_0) \subseteq \widehat{\mathsf{C}}(\ell)) \ \land \\
 & (\forall(\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \widehat{\mathsf{C}}(\ell_1) : & (\widehat{\mathsf{C}}, \widehat{\rho}) \models t_0^{\ell_0} \ \land \\
 & & \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\rho}(x) \ \land\ \widehat{\mathsf{C}}(\ell_0) \subseteq \widehat{\mathsf{C}}(\ell) \ \land \\
 & & \{\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}\} \subseteq \widehat{\rho}(f))
\end{array}
$$

- Instead, define ⊨ coinductively
  - Want the **greatest fixed point** that satisfies equations for ⊨
- Note: not an algorithm for solving, but a specification

Semantic correctness

- Also need to show that acceptability of analysis results implies semantic correctness
  - That is, ^C and ^p accurately describe the concrete execution.
  - Like a type-soundness statement

Syntax-directed 0-CFA

- Another formulation of 0-CFA that approximates the abstract specification
  - i.e., Define $\vDash_s$ such that

$$\text{if } (\hat{C}, \hat{}\rho) \vDash_s e \text{ then } (\hat{C}, \hat{}\rho) \vDash e$$

Syntax-directed 0-CFA

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s c^{\ell}$ always

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s x^{\ell}$ <u>iff</u> $\widehat{\rho}(x) \subseteq \widehat{\mathsf{C}}(\ell)$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s (\texttt{if } t_0^{\ell_0} \texttt{ then } t_1^{\ell_1} \texttt{ else } t_2^{\ell_2})^{\ell}$

<u>iff</u> $(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s t_0^{\ell_0} \; \wedge$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s t_1^{\ell_1} \; \wedge \; (\widehat{\mathsf{C}}, \widehat{\rho}) \models_s t_2^{\ell_2} \; \wedge$

$\widehat{\mathsf{C}}(\ell_1) \subseteq \widehat{\mathsf{C}}(\ell) \; \wedge \; \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\mathsf{C}}(\ell)$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s (\texttt{let } x \texttt{ = } t_1^{\ell_1} \texttt{ in } t_2^{\ell_2})^{\ell}$

<u>iff</u> $(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s t_1^{\ell_1} \; \wedge \; (\widehat{\mathsf{C}}, \widehat{\rho}) \models_s t_2^{\ell_2} \; \wedge$

$\widehat{\mathsf{C}}(\ell_1) \subseteq \widehat{\rho}(x) \; \wedge \; \widehat{\mathsf{C}}(\ell_2) \subseteq \widehat{\mathsf{C}}(\ell)$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s (t_1^{\ell_1} \; op \; t_2^{\ell_2})^{\ell}$ <u>iff</u> $(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s t_1^{\ell_1} \; \wedge \; (\widehat{\mathsf{C}}, \widehat{\rho}) \models_s t_2^{\ell_2}$

Syntax-directed 0-CFA

$$
\begin{aligned}
(\hat{C}, \hat{\rho}) \models_s (\texttt{fn}\ x\ \texttt{=>}\ e_0)^{\ell} & \\
\underline{\text{iff}} \quad & \{\texttt{fn}\ x\ \texttt{=>}\ e_0\} \subseteq \hat{C}(\ell) \ \wedge \\
& (\hat{C}, \hat{\rho}) \models_s e_0
\end{aligned}
$$

Note: *may check some function bodies that aren't reachable, in return for enabling induction on syntax*

$$
\begin{aligned}
(\hat{C}, \hat{\rho}) \models_s (\texttt{fun}\ f\ x\ \texttt{=>}\ e_0)^{\ell} & \\
\underline{\text{iff}} \quad & \{\texttt{fun}\ f\ x\ \texttt{=>}\ e_0\} \subseteq \hat{C}(\ell) \ \wedge \\
& (\hat{C}, \hat{\rho}) \models_s e_0 \ \wedge \ \{\texttt{fun}\ f\ x\ \texttt{=>}\ e_0\} \subseteq \hat{\rho}(f)
\end{aligned}
$$

$$
\begin{aligned}
(\hat{C}, \hat{\rho}) \models_s (t_1^{\ell_1}\ t_2^{\ell_2})^{\ell} & \\
\underline{\text{iff}} \quad & (\hat{C}, \hat{\rho}) \models_s t_1^{\ell_1} \ \wedge \ (\hat{C}, \hat{\rho}) \models_s t_2^{\ell_2} \ \wedge \\
& (\forall (\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \hat{C}(\ell_1) : \\
& \qquad \hat{C}(\ell_2) \subseteq \hat{\rho}(x) \ \wedge \ \hat{C}(\ell_0) \subseteq \hat{C}(\ell) \quad \boxed{}) \ \wedge \\
& (\forall (\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \hat{C}(\ell_1) : \\
& \qquad \hat{C}(\ell_2) \subseteq \hat{\rho}(x) \ \wedge \ \hat{C}(\ell_0) \subseteq \hat{C}(\ell) \quad \boxed{})
\end{aligned}
$$

Syntax-directed 0-CFA

- For any expression e, there is a least $(\hat{C}, \hat{\rho})$ such that $(\hat{C}, \hat{\rho}) \models_s e$
- Can turn this syntax-directed 0-CFA specification into an equivalent algorithm that generates a set of constraints
    - Least solution to set of constraints is least solution to syntax-directed 0-CFA

Constraint-based 0-CFA

$\mathcal{C}_\star[\![e_\star]\!]$ is a set of constraints of the form

$$\mathit{lhs} \subseteq \mathit{rhs}$$

$$\{t\} \subseteq \mathit{rhs}' \Rightarrow \mathit{lhs} \subseteq \mathit{rhs}$$

where

$$\begin{aligned} \mathit{rhs} &::= \mathsf{C}(\ell) \mid \mathsf{r}(x) \\ \mathit{lhs} &::= \mathsf{C}(\ell) \mid \mathsf{r}(x) \mid \{t\} \end{aligned}$$

and all occurrences of t are of the form `fn` x `=>` e_0 or `fun` f x `=>` e_0

Constraint-based 0-CFA

$$\mathcal{C}_\star[\![c^\ell]\!] = \emptyset$$

$$\mathcal{C}_\star[\![x^\ell]\!] = \{\, \mathsf{r}(x) \subseteq \mathsf{C}(\ell) \,\}$$

$$\begin{aligned}\mathcal{C}_\star[\![(\texttt{if } t_0^{\ell_0} \texttt{ then } t_1^{\ell_1} \texttt{ else } t_2^{\ell_2})^\ell]\!] &= \mathcal{C}_\star[\![t_0^{\ell_0}]\!] \cup \mathcal{C}_\star[\![t_1^{\ell_1}]\!] \cup \mathcal{C}_\star[\![t_2^{\ell_2}]\!] \\ &\cup \{\, \mathsf{C}(\ell_1) \subseteq \mathsf{C}(\ell) \,\} \\ &\cup \{\, \mathsf{C}(\ell_2) \subseteq \mathsf{C}(\ell) \,\}\end{aligned}$$

$$\begin{aligned}\mathcal{C}_\star[\![(\texttt{let } x \texttt{ = } t_1^{\ell_1} \texttt{ in } t_2^{\ell_2})^\ell]\!] &= \mathcal{C}_\star[\![t_1^{\ell_1}]\!] \cup \mathcal{C}_\star[\![t_2^{\ell_2}]\!] \\ &\cup \{\, \mathsf{C}(\ell_1) \subseteq \mathsf{r}(x) \,\} \cup \{\, \mathsf{C}(\ell_2) \subseteq \mathsf{C}(\ell) \,\}\end{aligned}$$

$$\mathcal{C}_\star[\![(t_1^{\ell_1} \; \mathit{op} \; t_2^{\ell_2})^\ell]\!] = \mathcal{C}_\star[\![t_1^{\ell_1}]\!] \cup \mathcal{C}_\star[\![t_2^{\ell_2}]\!]$$

Constraint-based 0-CFA

$$\mathcal{C}_\star[\![(\texttt{fn}\ x\ \texttt{=>}\ e_0)^{\ell}]\!] \;=\; \{\ \{\texttt{fn}\ x\ \texttt{=>}\ e_0\} \subseteq \mathsf{C}(\ell)\ \} \;\cup\; \mathcal{C}_\star[\![e_0]\!]$$

$$\begin{aligned}
\mathcal{C}_\star[\![(\texttt{fun}\ f\ x\ \texttt{=>}\ e_0)^{\ell}]\!] \;=\;& \{\ \{\texttt{fun}\ f\ x\ \texttt{=>}\ e_0\} \subseteq \mathsf{C}(\ell)\ \} \;\cup\; \mathcal{C}_\star[\![e_0]\!] \\
\cup\;& \{\ \{\texttt{fun}\ f\ x\ \texttt{=>}\ e_0\} \subseteq \mathsf{r}(f)\ \}
\end{aligned}$$

$$\begin{aligned}
\mathcal{C}_\star[\![(t_1^{\ell_1}\ \ t_2^{\ell_2})^{\ell}]\!] = \mathcal{C}_\star[\![t_1^{\ell_1}]\!] \cup \mathcal{C}_\star[\![t_2^{\ell_2}]\!]& \\
\cup\ \{\ \{t\} \subseteq \mathsf{C}(\ell_1) \Rightarrow \mathsf{C}(\ell_2) \subseteq \mathsf{r}(x) &\ \mid t = (\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \mathbf{Term}_\star\} \\
\cup\ \{\ \{t\} \subseteq \mathsf{C}(\ell_1) \Rightarrow \mathsf{C}(\ell_0) \subseteq \mathsf{C}(\ell) &\ \mid t = (\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \mathbf{Term}_\star\} \\
\cup\ \{\ \{t\} \subseteq \mathsf{C}(\ell_1) \Rightarrow \mathsf{C}(\ell_2) \subseteq \mathsf{r}(x) &\ \mid t = (\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \mathbf{Term}_\star\} \\
\cup\ \{\ \{t\} \subseteq \mathsf{C}(\ell_1) \Rightarrow \mathsf{C}(\ell_0) \subseteq \mathsf{C}(\ell) &\ \mid t = (\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}) \in \mathbf{Term}_\star\}
\end{aligned}$$

Example

$$
\begin{aligned}
&\mathcal{C}_\star[\![((\texttt{fn x => x}^1)^2\ (\texttt{fn y => y}^3)^4)^5]\!] = \\
&\quad \{\ \{\texttt{fn x => x}^1\} \subseteq \mathsf{C}(2), \\
&\quad\ \ \mathsf{r}(\texttt{x}) \subseteq \mathsf{C}(1), \\
&\quad\ \ \{\texttt{fn y => y}^3\} \subseteq \mathsf{C}(4), \\
&\quad\ \ \mathsf{r}(\texttt{y}) \subseteq \mathsf{C}(3), \\
&\quad\ \ \{\texttt{fn x => x}^1\} \subseteq \mathsf{C}(2) \Rightarrow \mathsf{C}(4) \subseteq \mathsf{r}(\texttt{x}), \\
&\quad\ \ \{\texttt{fn x => x}^1\} \subseteq \mathsf{C}(2) \Rightarrow \mathsf{C}(1) \subseteq \mathsf{C}(5), \\
&\quad\ \ \{\texttt{fn y => y}^3\} \subseteq \mathsf{C}(2) \Rightarrow \mathsf{C}(4) \subseteq \mathsf{r}(\texttt{y}), \\
&\quad\ \ \{\texttt{fn y => y}^3\} \subseteq \mathsf{C}(2) \Rightarrow \mathsf{C}(3) \subseteq \mathsf{C}(5)\ \}
\end{aligned}
$$

Correctness

Translating syntactic entities to sets of terms:

$$\begin{aligned} (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathsf{C}(\ell)]\!] &= \widehat{\mathsf{C}}(\ell) \\ (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathsf{r}(x)]\!] &= \widehat{\rho}(x) \\ (\widehat{\mathsf{C}}, \widehat{\rho})[\![\{t\}]\!] &= \{t\} \end{aligned}$$

Satisfaction relation for constraints: $(\widehat{\mathsf{C}}, \widehat{\rho}) \models_c (\mathit{lhs} \subseteq \mathit{rhs})$

$$\begin{aligned} & (\widehat{\mathsf{C}}, \widehat{\rho}) \models_c (\mathit{lhs} \subseteq \mathit{rhs}) \\ \underline{\text{iff}} \quad & (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathit{lhs}]\!] \subseteq (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathit{rhs}]\!] \end{aligned}$$

$$\begin{aligned} & (\widehat{\mathsf{C}}, \widehat{\rho}) \models_c (\{t\} \subseteq \mathit{rhs}' \Rightarrow \mathit{lhs} \subseteq \mathit{rhs}) \\ \underline{\text{iff}} \quad & (\{t\} \subseteq (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathit{rhs}']\!] \wedge (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathit{lhs}]\!] \subseteq (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathit{rhs}]\!]) \\ \vee \quad & (\{t\} \not\subseteq (\widehat{\mathsf{C}}, \widehat{\rho})[\![\mathit{rhs}']\!]) \end{aligned}$$

Proposition: $(\widehat{\mathsf{C}}, \widehat{\rho}) \models_s e_\star$ if and only if $(\widehat{\mathsf{C}}, \widehat{\rho}) \models_c \mathcal{C}_\star[\![e_\star]\!]$.

Adding data-flow analysis

- Current domain equations

Actually, just functions

$$\begin{array}{lllll} \widehat{v} & \in & \widehat{\mathbf{Val}} & = & \mathcal{P}(\mathbf{Term}) & \textit{abstract values} \\ \widehat{\rho} & \in & \widehat{\mathbf{Env}} & = & \mathbf{Var} \to \widehat{\mathbf{Val}} & \textit{abstract environments} \\ \widehat{\mathsf{C}} & \in & \widehat{\mathbf{Cache}} & = & \mathbf{Lab} \to \widehat{\mathbf{Val}} & \textit{abstract caches} \end{array}$$

- Idea: extend abstract values to include other things than just functions
- E.g., let Data be set of **abstract data values**
  - e.g., {tt, ff, -, 0, +}

$$\widehat{v} \in \widehat{\mathbf{Val}}_d = \mathcal{P}(\mathbf{Term} \cup \mathbf{Data}) \quad \text{abstract values}$$

Abstract data values

- For each constant *c*, need abstract data value d_c
- For each operator *op* need abstract operator *op* : Data×Data→P(Data)

Data$_{\text{sign}}$ $=$ {tt, ff, -, 0, +}

$d_{\mathtt{true}} =$ tt

$d_{7} =$ +

$\widehat{+}$ is defined from

d_{+}	tt	ff	-	0	+
tt	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
ff	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
-	$\emptyset$	$\emptyset$	{-}	{-}	{-, 0, +}
0	$\emptyset$	$\emptyset$	{-}	{0}	{+}
+	$\emptyset$	$\emptyset$	{-, 0, +}	{+}	{+}

Data-flow and control-flow spec

$$(\hat{\mathsf{C}}, \hat{\rho}) \models_d c^\ell \quad \underline{\text{iff}} \quad \{d_c\} \subseteq \hat{\mathsf{C}}(\ell)$$

$$(\hat{\mathsf{C}}, \hat{\rho}) \models_d x^\ell \quad \underline{\text{iff}} \quad \hat{\rho}(x) \subseteq \hat{\mathsf{C}}(\ell)$$

$$\begin{aligned}
&(\hat{\mathsf{C}}, \hat{\rho}) \models_d (\texttt{if } t_0^{\ell_0} \texttt{ then } t_1^{\ell_1} \texttt{ else } t_2^{\ell_2})^\ell \\
&\quad \underline{\text{iff}} \quad (\hat{\mathsf{C}}, \hat{\rho}) \models_d t_0^{\ell_0} \ \wedge \\
&\qquad (d_{\texttt{true}} \in \hat{\mathsf{C}}(\ell_0) \Rightarrow (\boxed{(\hat{\mathsf{C}}, \hat{\rho}) \models_d t_1^{\ell_1}} \ \wedge \ \hat{\mathsf{C}}(\ell_1) \subseteq \hat{\mathsf{C}}(\ell))) \ \wedge \\
&\qquad (d_{\texttt{false}} \in \hat{\mathsf{C}}(\ell_0) \Rightarrow (\boxed{(\hat{\mathsf{C}}, \hat{\rho}) \models_d t_2^{\ell_2}} \ \wedge \ \hat{\mathsf{C}}(\ell_2) \subseteq \hat{\mathsf{C}}(\ell)))
\end{aligned}$$

$$\begin{aligned}
&(\hat{\mathsf{C}}, \hat{\rho}) \models_d (\texttt{let } x \texttt{ = } t_1^{\ell_1} \texttt{ in } t_2^{\ell_2})^\ell \\
&\quad \underline{\text{iff}} \quad (\hat{\mathsf{C}}, \hat{\rho}) \models_d t_1^{\ell_1} \ \wedge \ (\hat{\mathsf{C}}, \hat{\rho}) \models_d t_2^{\ell_2} \ \wedge \ \hat{\mathsf{C}}(\ell_1) \subseteq \hat{\rho}(x) \ \wedge \ \hat{\mathsf{C}}(\ell_2) \subseteq \hat{\mathsf{C}}(\ell)
\end{aligned}$$

$$\begin{aligned}
&(\hat{\mathsf{C}}, \hat{\rho}) \models_d (t_1^{\ell_1} \ \mathsf{op} \ t_2^{\ell_2})^\ell \\
&\quad \underline{\text{iff}} \quad (\hat{\mathsf{C}}, \hat{\rho}) \models_d t_1^{\ell_1} \ \wedge \ (\hat{\mathsf{C}}, \hat{\rho}) \models_d t_2^{\ell_2} \ \wedge \ \hat{\mathsf{C}}(\ell_1) \ \widehat{\mathsf{op}} \ \hat{\mathsf{C}}(\ell_2) \subseteq \hat{\mathsf{C}}(\ell)
\end{aligned}$$

- Is flow sensitive: can determine whether true or false branches can be taken

Arbitrary lattices

- $^{\wedge}\mathrm{Val} = \mathcal{P}(\mathrm{Term} \cup \mathrm{Data}) = \mathcal{P}(\mathrm{Term}) \times \mathcal{P}(\mathrm{Data})$
- Could also use an arbitrary lattice
  $^{\wedge}\mathrm{Val} = \mathcal{P}(\mathrm{Term} \cup \mathrm{Data}) = \mathcal{P}(\mathrm{Term}) \times L$

Adding contexts

- 0-CFA is a context-insensitive (or **mono-variant**) analysis
    - Does not distinguish various instances of program variables and program points from each other
- Context-sensitive (or **poly-variant**) analysis does distingtuish

Uniform *k*-CFA

$$\begin{array}{llll} \hat{v} \in \widehat{\mathbf{Val}} & = & \mathcal{P}(\mathbf{Term}) & \textit{abstract values} \\ \hat{\rho} \in \widehat{\mathbf{Env}} & = & \mathbf{Var} \to \widehat{\mathbf{Val}} & \textit{abstract environments} \\ \hat{C} \in \widehat{\mathbf{Cache}} & = & \mathbf{Lab} \to \widehat{\mathbf{Val}} & \textit{abstract caches} \end{array}$$

- Idea: extend ^Val to include context information
- Contexts δ will record last *k* dynamic call-sites

Definition point of free variables of terms

$$\begin{array}{llll} \delta \in \Delta & = & \mathbf{Lab}^{\leq k} & \text{context information} \\ ce \in \mathbf{CEnv} & = & \mathbf{Var} \to \Delta & \text{context environments} \\ \hat{v} \in \widehat{\mathbf{Val}} & = & \mathcal{P}(\mathbf{Term} \times \mathbf{CEnv}) & \text{abstract values} \\ \hat{\rho} \in \widehat{\mathbf{Env}} & = & (\mathbf{Var} \times \Delta) \to \widehat{\mathbf{Val}} & \text{abstract environments} \\ \hat{C} \in \widehat{\mathbf{Cache}} & = & (\mathbf{Lab} \times \Delta) \to \widehat{\mathbf{Val}} & \text{abstract caches} \end{array}$$

- Called “uniform” because both environment and cache use same precision

Acceptability relation

$$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} e$$

- ce is current context environment
  - i.e., for free variables of e, in which context were they bound?
  - Changes as variables are bound
- δ is current context
  - Changes as functions are applied

Acceptability relation

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} c^{\ell}$ always

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} x^{\ell}$ $\quad$ iff $\quad$ $\widehat{\rho}(x, ce(x)) \subseteq \widehat{\mathsf{C}}(\ell, \delta)$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} (\texttt{if } t_0^{\ell_0} \texttt{ then } t_1^{\ell_1} \texttt{ else } t_2^{\ell_2})^{\ell}$

iff $\quad$ $(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_0^{\ell_0} \;\wedge\; (\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_1^{\ell_1} \;\wedge\; (\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_2^{\ell_2} \;\wedge$
$\widehat{\mathsf{C}}(\ell_1, \delta) \subseteq \widehat{\mathsf{C}}(\ell, \delta) \;\wedge\; \widehat{\mathsf{C}}(\ell_2, \delta) \subseteq \widehat{\mathsf{C}}(\ell, \delta)$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} (\texttt{let } x \texttt{ = } t_1^{\ell_1} \texttt{ in } t_2^{\ell_2})^{\ell}$

iff $\quad$ $(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_1^{\ell_1} \;\wedge\; (\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce'}_{\delta} t_2^{\ell_2} \;\wedge$
$\widehat{\mathsf{C}}(\ell_1, \delta) \subseteq \widehat{\rho}(x, \delta) \;\wedge\; \widehat{\mathsf{C}}(\ell_2, \delta) \subseteq \widehat{\mathsf{C}}(\ell, \delta)$
where $ce' = ce[x \mapsto \delta]$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} (t_1^{\ell_1} \; op \; t_2^{\ell_2})^{\ell}$ $\quad$ iff $\quad$ $(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_1^{\ell_1} \;\wedge\; (\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_2^{\ell_2}$

Acceptability relation

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} (\texttt{fn}\ x\ \texttt{=>}\ e_0)^{\ell}$ <u>iff</u> $\{(\texttt{fn}\ x\ \texttt{=>}\ e_0, ce)\} \subseteq \widehat{\mathsf{C}}(\ell, \delta)$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} (\texttt{fun}\ f\ x\ \texttt{=>}\ e_0)^{\ell}$ <u>iff</u> $\{(\texttt{fun}\ f\ x\ \texttt{=>}\ e_0, ce)\} \subseteq \widehat{\mathsf{C}}(\ell, \delta)$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} (t_1^{\ell_1}\ t_2^{\ell_2})^{\ell}$

<u>iff</u> $(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_1^{\ell_1} \ \wedge\ (\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce}_{\delta} t_2^{\ell_2}\ \wedge$

$(\forall (\texttt{fn}\ x\ \texttt{=>}\ t_0^{\ell_0}, ce_0) \in \widehat{\mathsf{C}}(\ell_1, \delta) :$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce_0'}_{\delta_0} t_0^{\ell_0}\ \wedge\ \widehat{\mathsf{C}}(\ell_2, \delta) \subseteq \widehat{\rho}(x, \delta_0)\ \wedge\ \widehat{\mathsf{C}}(\ell_0, \delta_0) \subseteq \widehat{\mathsf{C}}(\ell, \delta)$

where $\delta_0 = \lceil \delta, \ell \rceil_k$ and $ce_0' = ce_0[x \mapsto \delta_0])\ \wedge$

$(\forall (\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}, ce_0) \in \widehat{\mathsf{C}}(\ell_1, \delta) :$

$(\widehat{\mathsf{C}}, \widehat{\rho}) \models^{ce_0'}_{\delta_0} t_0^{\ell_0}\ \wedge\ \widehat{\mathsf{C}}(\ell_2, \delta) \subseteq \widehat{\rho}(x, \delta_0)\ \wedge\ \widehat{\mathsf{C}}(\ell_0, \delta_0) \subseteq \widehat{\mathsf{C}}(\ell, \delta)\ \wedge$

$\{(\texttt{fun}\ f\ x\ \texttt{=>}\ t_0^{\ell_0}, ce_0)\} \subseteq \widehat{\rho}(f, \delta_0)$

where $\delta_0 = \lceil \delta, \ell \rceil_k$ and $ce_0' = ce_0[f \mapsto \delta_0, x \mapsto \delta_0])$

Example

$$\texttt{(let f = (fn x => x}^{1}\texttt{)}^{2}\texttt{ in ((f}^{3}\texttt{ f}^{4}\texttt{)}^{5}\texttt{ (fn y => y}^{6}\texttt{)}^{7}\texttt{)}^{8}\texttt{)}^{9}$$

- Contexts of interest for uniform 1-CFA

Λ: the initial context
5: the context when the application point labelled 5 has been passed
8: the context when the application point labelled 8 has been passed

- Context environments of interest for uniform 1-CFA

$\mathsf{ce}_0 = [\,]$ the initial (empty) context environment
$\mathsf{ce}_1 = \mathsf{ce}_0[\texttt{f} \mapsto \Lambda]$ the context environment for the analysis of the body of the `let`-construct
$\mathsf{ce}_2 = \mathsf{ce}_0[\texttt{x} \mapsto 5]$ the context environment used for the analysis of the body of `f` initiated at the application point 5
$\mathsf{ce}_3 = \mathsf{ce}_0[\texttt{x} \mapsto 8]$ the context environment used for the analysis of the body of `f` initiated at the application point 8.

Example

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(1,5) = \{(\texttt{fn x => x}^{1},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(2,\Lambda) = \{(\texttt{fn x => x}^{1},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(4,\Lambda) = \{(\texttt{fn x => x}^{1},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(7,\Lambda) = \{(\texttt{fn y => y}^{6},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(9,\Lambda) = \{(\texttt{fn y => y}^{6},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(1,8) = \{(\texttt{fn y => y}^{6},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(3,\Lambda) = \{(\texttt{fn x => x}^{1},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(5,\Lambda) = \{(\texttt{fn x => x}^{1},\mathsf{ce}_0)\}$$

$$\widehat{\mathsf{C}}_{\mathsf{id}}'(8,\Lambda) = \{(\texttt{fn y => y}^{6},\mathsf{ce}_0)\}$$

$$\widehat{\rho}_{\mathsf{id}}'(\texttt{f},\Lambda) = \{(\texttt{fn x => x}^{1},\mathsf{ce}_0)\}$$

$$\widehat{\rho}_{\mathsf{id}}'(\texttt{x},5) = \{(\texttt{fn x => x}^{1},\mathsf{ce}_0)\} \qquad \widehat{\rho}_{\mathsf{id}}'(\texttt{x},8) = \{(\texttt{fn y => y}^{6},\mathsf{ce}_0)\}$$

This is an acceptable analysis result:

$$(\widehat{\mathsf{C}}_{\mathsf{id}}', \widehat{\rho}_{\mathsf{id}}') \models^{\mathsf{ce}_0}_{\Lambda} (\texttt{let f = (fn x => x}^{1})^{2}\ \texttt{in}\ ((\texttt{f}^{3}\ \texttt{f}^{4})^{5}\ (\texttt{fn y => y}^{6})^{7})^{8})^{9}$$

Complexity

$$\begin{array}{lcllr}
\delta & \in & \Delta & = & \mathbf{Lab}^{\leq k} & \text{context information} \\
ce & \in & \mathbf{CEnv} & = & \mathbf{Var} \to \Delta & \text{context environments} \\
\widehat{v} & \in & \widehat{\mathbf{Val}} & = & \mathcal{P}(\mathbf{Term} \times \mathbf{CEnv}) & \text{abstract values} \\
\widehat{\rho} & \in & \widehat{\mathbf{Env}} & = & (\mathbf{Var} \times \Delta) \to \widehat{\mathbf{Val}} & \text{abstract environments} \\
\widehat{\mathsf{C}} & \in & \widehat{\mathbf{Cache}} & = & (\mathbf{Lab} \times \Delta) \to \widehat{\mathbf{Val}} & \text{abstract caches}
\end{array}$$

- k-CFA has worst-case exponential complexity in size of program
  - Size n program, p variables
  - Δ has O(n) elements
  - Size of CEnv is $O(n^p)$
  - ^Val is powerset of pairs (t, ce), and there are $O(n \times n^p)$ pairs, so Val has height $O(n \times n^p)$
  - p = O(n)
- 0-CFA has worst-case polynomial complexity

Variations on *k*-CFA

Uniform *k*-CFA

$$
\begin{array}{llcll}
ce & \in & \mathbf{CEnv} & = & \mathbf{Var} \to \boxed{\Delta} & \text{context environments} \\
\hat{v} & \in & \widehat{\mathbf{Val}} & = & \mathcal{P}(\mathbf{Term} \times \mathbf{CEnv}) & \text{abstract values} \\
\hat{\rho} & \in & \widehat{\mathbf{Env}} & = & (\mathbf{Var} \times \boxed{\Delta}) \to \widehat{\mathbf{Val}} & \text{abstract environments} \\
\hat{\mathsf{C}} & \in & \widehat{\mathbf{Cache}} & = & (\mathbf{Lab} \times \boxed{\Delta}) \to \widehat{\mathbf{Val}} & \text{abstract caches}
\end{array}
$$

k-CFA

$$
\hat{\mathsf{C}} \in \widehat{\mathbf{Cache}} = (\mathbf{Lab} \times \mathbf{CEnv}) \to \widehat{\mathbf{Val}} \quad \text{abstract caches}
$$

Polynomial *k*-CFA

$$
\hat{v} \in \widehat{\mathbf{Val}} = \mathcal{P}(\mathbf{Term} \times \boxed{\Delta}) \quad \text{abstract values}
$$

k-CFA Paradox

- [Might, Smaragdakis, van Horn, PLDI 10]
- *k*-CFA is exponential for $k \geq 1$
- But *k*-CFA is like using context of *k* most recent call-sites
    - Polynomial for OO languages
        - Doop implemented in Datalog, which only allows polynomial time alogrithms
    - OO has dynamic dispatch
- What gives?

Wait, which *k*-CFA?

- In OO world, translate *k*-CFA to "*k*-call-site sensitive interprocedural pointer analysis with a k-context-sensitive heap and on-the-fly call-graph construction"
    - i.e., data flow (points-to relation) and call-graph dependent on each other
- Is it the same analysis? Yes. And paradox still holds.

Paradox resolved

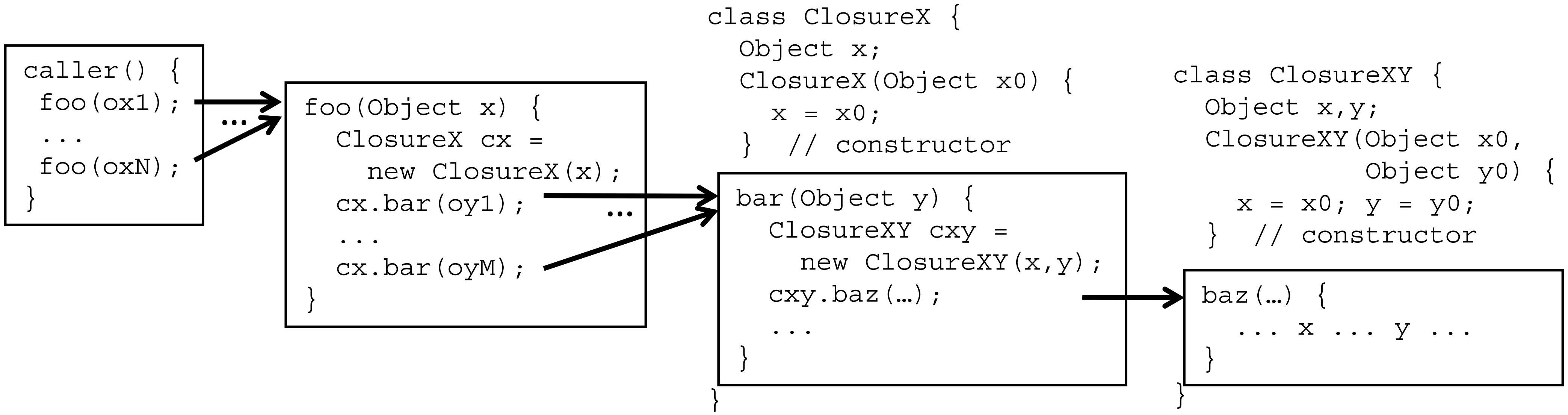
- In functional languages, closures are created incrementally
  - Each variable in a closure could be bound in a different context
  - Source of exponentiallity

$$\begin{array}{rcllll}
\delta & \in & \Delta & = & \mathbf{Lab}^{\leq k} & \text{context information} \\
ce & \in & \mathbf{CEnv} & = & \mathbf{Var} \to \Delta & \text{context environments} \\
\widehat{v} & \in & \widehat{\mathbf{Val}} & = & \mathcal{P}(\mathbf{Term} \times \mathbf{CEnv}) & \text{abstract values} \\
\widehat{\rho} & \in & \widehat{\mathbf{Env}} & = & (\mathbf{Var} \times \Delta) \to \widehat{\mathbf{Val}} & \text{abstract environments} \\
\widehat{\mathsf{C}} & \in & \widehat{\mathbf{Cache}} & = & (\mathbf{Lab} \times \Delta) \to \widehat{\mathbf{Val}} & \text{abstract caches}
\end{array}$$

- In OO languages, closures created explicitly by invoking constructor
  - Variables are *copied*, and so effectively all variables bound in same context
  - CEnv = Δ instead of Var → Δ

Example

- OO program



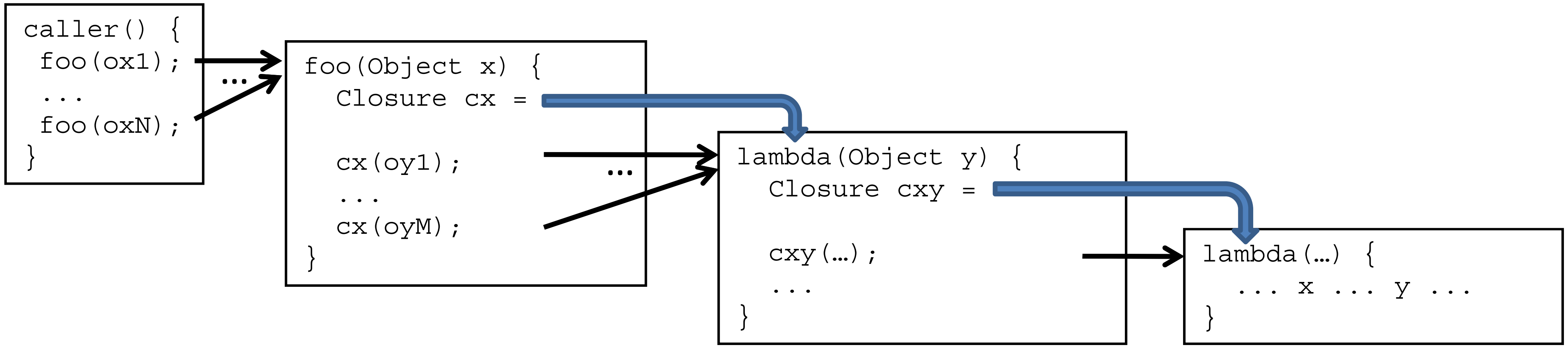
```
caller() {
  foo(ox1);
  ...
  foo(oxN);
}
```

```
foo(Object x) {
  ClosureX cx =
    new ClosureX(x);
  cx.bar(oy1);
  ...
  cx.bar(oyM);
}
```

```
class ClosureX {
  Object x;
  ClosureX(Object x0) {
    x = x0;
  }  // constructor
  bar(Object y) {
    ClosureXY cxy =
      new ClosureXY(x,y);
    cxy.baz(…);
    ...
  }
}
```

```
class ClosureXY {
  Object x,y;
  ClosureXY(Object x0,
            Object y0) {
    x = x0; y = y0;
  }  // constructor
  baz(…) {
    ... x ... y ...
  }
}
```

- Equivalent functional program



```
caller() {
  foo(ox1);
  ...
  foo(oxN);
}
```

```
foo(Object x) {
  Closure cx =

  cx(oy1);
  ...
  cx(oyM);
}
```

```
lambda(Object y) {
  Closure cxy =

  cxy(…);
  ...
}
```

```
lambda(…) {
  ... x ... y ...
}
```

m-CFA

- From this insight, Might, Smaragdakis and Van Horn develop *m*-CFA
  - Contexts are the top *m* stack frames
    - (Different from last *k* call sites when in continuation-passing style)
  - Essentially CEnv = Δ instead of Var → Δ
- Polynomial-time analysis, seems as precise a *k*-CFA for significantly less time