

# An Introduction to Control Flow Analysis

---

Benoît Montagu

<https://epicure.gitlabpages.inria.fr/m2-sos/>

SOS, Master Recherche Science Informatique, Univ Rennes

2024-2025

# Introduction

---

## Complex Control Flow (1/3)

Sometimes, it is difficult to know which function is called at some call site

For example, in an **object oriented language**, you could write:

```
1 abstract class A {
2     abstract public int m(int x);
3 }
4 class A1 extends A {
5     public int m(int x) { return x; }
6 }
7 class A2 extends A {
8     public int m(int x) { return 2*x; }
9 }
10 class C {
11     public int foo(boolean flag, int arg) {
12         A obj = null;
13         if (flag) obj = new A1();
14         else obj = new A2();
15         return obj.m(arg);
16     }
17 }
```

On line 15, two programs can be executed: `A1.m` (line 5), or `A2.m` (line 8)

This depends on the value of the argument `flag` (line 11)

1/60

## Complex Control Flow (2/3)

Another example in C, that involves **function pointers**:

```
1 int f1(int x) { return x; }
2 int f2(int x) { return 2*x; }
3 int foo(int flag, int arg) {
4     int (*f)(int);
5     if (flag) f = &f1;
6     else f = &f2;
7     return (*f)(arg);
8 }
```

2/60

## Complex Control Flow (3/3)

The same example, in OCaml:

```
1 let f1 x = x
2 let f2 x = 2 * x

3 let foo flag arg =
4   let g = if flag then f1 else f2 in
5   g arg
```

A more complex example (in continuation passing style):

```
1 let rec fact_cps n =
2   if n <= 0 then fun k -> k 1
3   else
4     let f = fact_cps (n-1) in
5     fun k -> f (fun fact_n_1 -> k (n * fact_n_1))
6 let fact n = fact_cps n (fun result -> result)
```

3/60

## Approximating control flow

Why do we want to know which functions can be called?

- ▶ To perform optimizations (function inlining, code specialization, ...)
- ▶ To determine whether a call might change memory (to enable more optimizations)
- ▶ To determine the stack consumption of a program
- ▶ To detect whether some exception might be raised
- ▶ To detect dead code with precision
- ▶ To be sure that some dangerous function cannot be called
- ▶ Analyze binary code
- ▶ ...

This is a general, difficult problem: [Control Flow Analysis \(CFA\)](#)

4/60

## The Control Flow Analysis Problem

Given a complete program  $P$ :



- ▶ For every call site in  $P$ , determine which functions might be called, and
- ▶ For every function in  $P$ , determine on which arguments it might be called

In this course: we introduce 0-CFA (the simplest version of CFA)

- ▶ On a minimalistic language
- ▶ A definition of what is a correct solution
- ▶ A formulation based on set constraints
- ▶ The main arguments of the soundness proof
- ▶ A reconstruction of 0-CFA based on abstract interpretation
- ▶ Some extensions (very quickly)

5/60

## Some References

-  Flemming Nielson, Hanne Riis Nielson, and Chris Hankin. *Principles of Program Analysis*. Springer Berlin Heidelberg, 1999. DOI: 10.1007/978-3-662-03811-6
-  Benoît Montagu and Thomas P. Jensen. “Trace-Based Control-Flow Analysis”. In: *PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, June 20-25, 2021*. Ed. by Stephen N. Freund and Eran Yahav. ACM, 2021, pp. 482–496. DOI: 10.1145/3453483.3454057

6/60

## A Language with First-Class Functions

## Our language of study

The untyped  $\lambda$ -calculus with *call by value* semantics

► Syntax:

$$t ::= x \quad (\text{variable})$$

$$| \quad tt \quad (\text{application})$$

$$| \quad \lambda x. t \quad (\text{abstraction})$$

$$| \quad (t)^p \quad (\text{annotation})$$

- The form  $(t)^p$  is the term  $t$  *labelled* with the program point  $p$
- Program points are akin to program locations (file/line/column) in the text of a program
- Program points identify subterms of the initial/source program
- The labels  $p$  do not have to be distinct
- We only consider “correctly labeled programs”:
  - Each variable, abstraction and application is enclosed in an annotation
  - There is no annotation in an annotation:  $((t)^{p_1})^{p_2}$  is forbidden

7/60

## Small-step semantics

Values:  $v ::= \lambda x. t$

Substitution:  $t_1[x \leftarrow t_2]$  (replaces every  $x$  in  $t_1$  with  $t_2$ )

Call by value, left to right semantics, for correctly labeled terms:

$$\text{BETAV} \quad \frac{}{(\lambda x. (t)^{p_0})^{p_1} (v)^{p_2} \rightarrow t[x \leftarrow v]}$$

$$\text{APPCTXL} \quad \frac{t_1 \rightarrow t'_1}{t_1 t_2 \rightarrow t'_1 t_2}$$

$$\text{APPCTXR} \quad \frac{t \rightarrow t'}{(v)^p t \rightarrow (v)^p t'}$$

$$\text{ANNOTCTX} \quad \frac{t \rightarrow t'}{(t)^p \rightarrow (t')^p}$$

8/60

## Exercises

### Exercise 2.1

Reduce to its normal form the following term

$$\left( \left( \left( \lambda f. (\lambda x. (f^1 x^2)^3)^4 \right)^5 (\lambda x. (\lambda y. x^6)^7)^8 \right)^9 ((\lambda y. y^{10})^{11} (\lambda z. z^{12})^{13})^{14} \right)^{15}$$

### Exercise 2.2 (At home)

Extend the language and its semantics with `let` bindings as in OCaml.

### Exercise 2.3 (At home)

Extend the language and its semantics with integers, booleans, arithmetic expressions, arithmetic tests, and an `if...then...else` construct.

### Exercise 2.4 (At home)

Extend the semantics with a new construct for recursive functions `fix f x. t` that behaves like OCaml's `let rec f x = t in f`

9/60

## 0-CFA

## 0-CFA

Abstract objects computed by the 0-CFA analysis:

- ▶ An **abstract value**  $\hat{v}$  is a finite set of values (with free variables) that syntactically occur in the source program
- ▶ An **abstract cache**  $\hat{C}$  maps labels to abstract values
  - ☞ Tells which values a program point might produce
- ▶ An **abstract environment**  $\hat{\rho}$  maps variables to abstract values
  - ☞ Tells which values a variable might be instantiated with

10/60

## Example

Consider the program:

$$\left( \left( (\lambda x_1 . x_1^1)^2 (\lambda y . (\lambda z . y^3)^4)^5 \right)^6 (\lambda x_2 . x_2^7)^8 \right)^9 \quad \text{reminder: } (\lambda x . (t)^{p_0})^{p_1} (v)^{p_2} \rightarrow t[x \leftarrow v]$$

$$\rightarrow \left( (\lambda y . (\lambda z . y^3)^4)^6 (\lambda x_2 . x_2^7)^8 \right)^9$$

$$\rightarrow (\lambda z . (\lambda x_2 . x_2^7)^3)^9$$

We have the cache and environment:

$$\hat{C}(1) = \hat{C}(5) = \hat{C}(6) = \{ \lambda y . (\lambda z . y^3)^4 \} \quad \hat{\rho}(x_1) = \{ \lambda y . (\lambda z . y^3)^4 \}$$

$$\hat{C}(2) = \{ \lambda x_1 . x_1^1 \} \quad \hat{\rho}(y) = \{ \lambda x_2 . x_2^7 \}$$

$$\hat{C}(3) = \hat{C}(8) = \{ \lambda x_2 . x_2^7 \} \quad \hat{\rho}(z) = \hat{\rho}(x_2) = \emptyset$$

$$\hat{C}(4) = \hat{C}(9) = \{ \lambda z . y^3 \} \quad \text{☞ } z \text{ and } x_2 \text{ are never instantiated}$$

$$\hat{C}(7) = \emptyset$$

☞  $x_2^7$  never produces a value

11/60

## Acceptability relation

Correct caches and environments for a term  $t$  are specified by the acceptability relation  $(\hat{C}, \hat{\rho}) \models t$

$(\hat{C}, \hat{\rho}) \models t$  reads:

“The abstract cache  $\hat{C}$  and the abstract environment  $\hat{\rho}$  correctly approximate the behaviour of the program  $t$ ”

$$\text{VAR} \frac{\hat{\rho}(x) \subseteq \hat{C}(p)}{(\hat{C}, \hat{\rho}) \models x^p} \quad \text{LAM} \frac{\{ \lambda x . (t_0)^{p_0} \} \subseteq \hat{C}(p)}{(\hat{C}, \hat{\rho}) \models (\lambda x . (t_0)^{p_0})^p}$$

$$(\hat{C}, \hat{\rho}) \models (t_1)^{p_1} \quad (\hat{C}, \hat{\rho}) \models (t_2)^{p_2}$$

$$\text{APP} \frac{\forall (\lambda x . (t_0)^{p_0}) \in \hat{C}(p_1), \quad (\hat{C}, \hat{\rho}) \models (t_0)^{p_0} \wedge \hat{C}(p_2) \subseteq \hat{\rho}(x) \wedge \hat{C}(p_0) \subseteq \hat{C}(p)}{(\hat{C}, \hat{\rho}) \models ((t_1)^{p_1} (t_2)^{p_2})^p}$$

12/60

## Example

Consider the same program:  $t = \left( \left( (\lambda x_1 . x_1^4)^2 (\lambda y . (\lambda z . y^3)^4)^5 \right)^6 (\lambda x_2 . x_2^7)^8 \right)^9$

$$\begin{aligned} \hat{C}(1) = \hat{C}(5) = \hat{C}(6) &= \{\lambda y . (\lambda z . y^3)^4\} & \hat{\rho}(x_1) &= \{\lambda y . (\lambda z . y^3)^4\} \\ \hat{C}(2) &= \{\lambda x_1 . x_1^4\} & \hat{\rho}(y) &= \{\lambda x_2 . x_2^7\} \\ \hat{C}(3) = \hat{C}(8) &= \{\lambda x_2 . x_2^7\} & \hat{\rho}(z) = \hat{\rho}(x_2) &= \emptyset \\ \hat{C}(4) = \hat{C}(9) &= \{\lambda z . y^3\} \\ \hat{C}(7) &= \emptyset \end{aligned}$$

Let us check all the constraints for  $(\hat{C}, \hat{\rho}) \models t$

$$\begin{array}{llll} \hat{\rho}(x_1) \subseteq \hat{C}(1) & (\text{VAR}) \text{ at } 1 & \hat{C}(5) \subseteq \hat{\rho}(x_1) \wedge \hat{C}(1) \subseteq \hat{C}(6) & (\text{APP}) \text{ at } 6 \\ \{\lambda x_1 . x_1^4\} \subseteq \hat{C}(2) & (\text{LAM}) \text{ at } 2 & \hat{\rho}(x_2) \subseteq \hat{C}(7) & (\text{VAR}) \text{ at } 7 \\ \hat{\rho}(y) \subseteq \hat{C}(3) & (\text{VAR}) \text{ at } 3 & \{\lambda x_2 . x_2^7\} \subseteq \hat{C}(8) & (\text{LAM}) \text{ at } 8 \\ \{\lambda z . y^3\} \subseteq \hat{C}(4) & (\text{LAM}) \text{ at } 4 & \hat{C}(8) \subseteq \hat{\rho}(y) \wedge \hat{C}(4) \subseteq \hat{C}(9) & (\text{APP}) \text{ at } 9 \\ \{\lambda y . (\lambda z . y^3)^4\} \subseteq \hat{C}(5) & (\text{LAM}) \text{ at } 5 & & \end{array}$$

All the constraints are satisfied, therefore  $(\hat{C}, \hat{\rho})$  is a valid solution for  $t$ !

13/60

## Many solutions to the acceptability problem

For example, for the previous program:

- There is no constraint at all for  $\hat{\rho}(z)$
- We can choose any value for  $\hat{\rho}(x_2)$  and  $\hat{C}(7)$  as long as  $\hat{\rho}(x_2) \subseteq \hat{C}(7)$
- For  $\hat{C}(8)$ , we can choose any set that contains  $\lambda x_2 . x_2^7$

14/60

## A best solution

### Proposition 3.1

For any program  $t$ , there exists  $\hat{C}$  and  $\hat{\rho}$  such that  $(\hat{C}, \hat{\rho}) \models t$

### Definition 1

For  $f_1$  and  $f_2$  in  $A \rightarrow \wp(B)$ , define:  $f_1 \sqcap f_2 \triangleq \lambda a . f_1(a) \cap f_2(a)$

### Proposition 3.2

If  $(\hat{C}_1, \hat{\rho}_1) \models t$  and  $(\hat{C}_2, \hat{\rho}_2) \models t$ , then  $(\hat{C}_1 \sqcap \hat{C}_2, \hat{\rho}_1 \sqcap \hat{\rho}_2) \models t$

### Exercise 3.1 (At home)

Prove this proposition

### Definition 2

For  $f_1$  and  $f_2$  in  $A \rightarrow \wp(B)$ , define:  $f_1 \sqsubseteq f_2 \triangleq \forall a . f_1(a) \subseteq f_2(a)$

### Theorem 3

For any program  $t$ , there exists a  $\sqsubseteq$ -minimal solution  $(\hat{C}, \hat{\rho})$  such that  $(\hat{C}, \hat{\rho}) \models t$

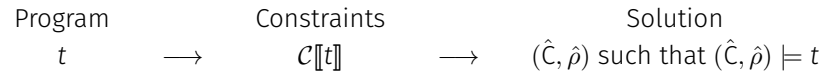
15/60

## Constraint-based 0-CFA

## Constraints

**Problem:** How can we compute the best solution?

**Roadmap:**



**Constraints**  $\mathcal{C}$  are finite sets of atomic constraints  $c$  of two forms:

Inclusion constraints:  $lhs \subseteq rhs$   
 Conditional constraints:  $\{\lambda x. t\} \subseteq rhs' \Rightarrow lhs \subseteq rhs$   
 where  $lhs ::= \{\lambda x. t\} \mid C(p) \mid r(x)$   
 $rhs ::= C(p) \mid r(x)$

We write **Terms**( $t$ ) to denote the set of subterms of the program  $t$

▲ Constraints are just pieces of syntax

16/60

## Constraints for 0-CFA

Let  $t$  be a program.

For any  $t' \in \mathbf{Terms}(t)$ , we define  $\mathcal{C}[[t']]$  by induction on  $t'$  as follows:

$$\begin{aligned} \mathcal{C}[[x^p]] &\triangleq \{r(x) \subseteq C(p)\} \\ \mathcal{C}[[\lambda x. (t_0)^{p_0}]] &\triangleq \mathcal{C}[[t_0]^{p_0}] \cup \{\{\lambda x. (t_0)^{p_0}\} \subseteq C(p)\} \\ \mathcal{C}[[ (t_1)^{p_1} (t_2)^{p_2} ]^p ] &\triangleq \mathcal{C}[[t_1]^{p_1}] \cup \mathcal{C}[[t_2]^{p_2}] \\ &\quad \cup \bigcup_{\lambda x. (t_0)^{p_0} \in \mathbf{Terms}(t)} \{\{\lambda x. (t_0)^{p_0}\} \subseteq C(p_1) \Rightarrow C(p_2) \subseteq r(x)\} \\ &\quad \cup \bigcup_{\lambda x. (t_0)^{p_0} \in \mathbf{Terms}(t)} \{\{\lambda x. (t_0)^{p_0}\} \subseteq C(p_1) \Rightarrow C(p_0) \subseteq C(p)\} \end{aligned}$$

- There are constraints for every function bodies
- Applications do not generate constraints for the called functions

**Remark**

If  $\text{card}(\mathbf{Terms}(t)) = n$ , then  $\text{card}(\mathcal{C}[[t]]) = \mathcal{O}(n^2)$

17/60

## Semantics of constraints

We define what it means for a constraint to be satisfied:

$$\begin{aligned} (\hat{C}, \hat{\rho})[[\{t\}]] &\triangleq \{t\} \\ (\hat{C}, \hat{\rho})[[C(p)]] &\triangleq \hat{C}(p) \\ (\hat{C}, \hat{\rho})[[r(x)]] &\triangleq \hat{\rho}(x) \end{aligned}$$

$$\begin{aligned} (\hat{C}, \hat{\rho}) \models_c lhs \subseteq rhs &\triangleq (\hat{C}, \hat{\rho})[[lhs]] \subseteq (\hat{C}, \hat{\rho})[[rhs]] \\ (\hat{C}, \hat{\rho}) \models_c \{t\} \subseteq rhs' \Rightarrow lhs \subseteq rhs &\triangleq (\hat{C}, \hat{\rho}) \models_c \{t\} \subseteq rhs' \Rightarrow (\hat{C}, \hat{\rho}) \models_c lhs \subseteq rhs \\ (\hat{C}, \hat{\rho}) \models_c \mathcal{C} &\triangleq \forall c \in \mathcal{C}, (\hat{C}, \hat{\rho}) \models_c c \end{aligned}$$

**Theorem 4**

Assume that  $\hat{C}$  and  $\hat{\rho}$  contain only variables/labels/terms found in  $t$ .

If  $(\hat{C}, \hat{\rho}) \models_c \mathcal{C}[[t]]$ , then  $(\hat{C}, \hat{\rho}) \models t$

- We reduced the 0-CFA problem to a constraint satisfiability problem
- Is it an equivalence?

18/60

## Example

$$\Omega = \left( (\lambda x. (x^1 x^2)^3)^4 (\lambda y. (y^5 y^6)^7)^8 \right)^9 \quad (\Omega \text{ is a diverging program})$$

$$\mathcal{C}[[\Omega]] = \left\{ \begin{array}{ll} r(x) \subseteq C(1), & r(x) \subseteq C(2), \\ \{\lambda x. (x^1 x^2)^3\} \subseteq C(1) \Rightarrow C(2) \subseteq r(x), & \{\lambda x. (x^1 x^2)^3\} \subseteq C(1) \Rightarrow C(3) \subseteq C(3), \\ \{\lambda y. (y^5 y^6)^7\} \subseteq C(1) \Rightarrow C(2) \subseteq r(y), & \{\lambda y. (y^5 y^6)^7\} \subseteq C(1) \Rightarrow C(7) \subseteq C(3), \\ \{\lambda x. (x^1 x^2)^3\} \subseteq C(4), & \\ r(y) \subseteq C(5), & r(y) \subseteq C(6), \\ \{\lambda x. (x^1 x^2)^3\} \subseteq C(5) \Rightarrow C(6) \subseteq r(x), & \{\lambda x. (x^1 x^2)^3\} \subseteq C(5) \Rightarrow C(3) \subseteq C(7), \\ \{\lambda y. (y^5 y^6)^7\} \subseteq C(5) \Rightarrow C(6) \subseteq r(y), & \{\lambda y. (y^5 y^6)^7\} \subseteq C(5) \Rightarrow C(7) \subseteq C(7), \\ \{\lambda y. (y^5 y^6)^7\} \subseteq C(8), & \\ \{\lambda x. (x^1 x^2)^3\} \subseteq C(4) \Rightarrow C(8) \subseteq r(x), & \{\lambda x. (x^1 x^2)^3\} \subseteq C(4) \Rightarrow C(3) \subseteq C(9), \\ \{\lambda y. (y^5 y^6)^7\} \subseteq C(4) \Rightarrow C(8) \subseteq r(y), & \{\lambda y. (y^5 y^6)^7\} \subseteq C(4) \Rightarrow C(7) \subseteq C(9) \end{array} \right\}$$

**Minimal solution:**

$$C(1, 2, 5, 6, 8) = r(x, y) = \{\lambda y. (y^5 y^6)^7\}, \quad C(4) = \{\lambda x. (x^1 x^2)^3\}, \quad C(3, 7, 9) = \emptyset$$

19/60

## Solving the constraints

- ▶ There exists an algorithm that computes the minimal solution of  $\mathcal{C}[[t]]$  in  $\mathcal{O}(n^3)$  worst case time complexity, where  $n$  is the size of  $t$
- ▶ It works by propagating information in a graph that represents constraints
- ▶ The details are out of scope for this course
- 🔗 For more details, see the book  
*Principles of Program Analysis*, Chapter 3 Section 4

20/60

## Soundness of 0-CFA

## Is the acceptability relation sound?

- ▶ We want a *theorem* saying that acceptable solutions  $(\hat{C}, \hat{\rho})$  such that  $(\hat{C}, \hat{\rho}) \models t$  “correctly approximate the behaviour of the program  $t$ ”
- ▶ How can we state this?
- ▶ **Crux of the problem:**  
Abstract values are finite sets of values that occur in the program...  
What concrete values are represented by those abstract values?
- ▶ **Intuition of the solution:**  
A free variable  $x$  in an abstract value can be replaced with any values drawn from  $\hat{\rho}(x)$
- ▶ Can we state this more formally?

21/60

## Unfoldings

### Unfolding relation:

$(t, \hat{\rho}) \models_{\top}^{\emptyset} t'$  means “term  $t'$  is obtained from  $t$  by unfolding definitions of  $\hat{\rho}$ ”

$$\frac{t \in \hat{\rho}(x) \quad (t, \hat{\rho}) \models_{\top}^{\emptyset} t' \quad X \cap (\{x\} \cup \text{fv } t') = \emptyset}{(x, \hat{\rho}) \models_{\top}^X t'} \quad \frac{}{(x, \hat{\rho}) \models_{\top}^X x}$$

$$\frac{(t, \hat{\rho}) \models_{\top}^{\{x\} \cup X} t'}{(\lambda x. t, \hat{\rho}) \models_{\top}^X \lambda x. t'} \quad \frac{(t_1, \hat{\rho}) \models_{\top}^X t'_1 \quad (t_2, \hat{\rho}) \models_{\top}^X t'_2}{(t_1 t_2, \hat{\rho}) \models_{\top}^X t'_1 t'_2} \quad \frac{(t, \hat{\rho}) \models_{\top}^X t'}{((t)^P, \hat{\rho}) \models_{\top}^X (t')^P}$$

- ▶ The structure of terms is preserved:  
only variables can be replaced with other terms
- ▶ The set  $X$  forbids unfolding some variables to prevent accidental captures/escapes of variables

22/60

## Unfoldings: Examples

$$\begin{aligned} (x, \{x \mapsto \{\lambda z. z, \lambda y_1. \lambda y_2. y_1\}\}) &\models_{\top}^{\emptyset} x \\ &\models_{\top}^{\emptyset} \lambda z. z \\ &\models_{\top}^{\emptyset} \lambda y_1. \lambda y_2. y_1 \end{aligned}$$

$$\begin{aligned} (x, \{x \mapsto \{\lambda z. z, \lambda y. x\}\}) &\models_{\top}^{\emptyset} \lambda z. z \\ &\models_{\top}^{\emptyset} \lambda y. \lambda z. z \\ &\models_{\top}^{\emptyset} \lambda y. \lambda y. \lambda z. z \\ &\models_{\top}^{\emptyset} \lambda y. \lambda y. \lambda y. \lambda z. z \quad \dots \end{aligned}$$

$$\begin{aligned} (x, \{x \mapsto \{\lambda z. x\}\}) &\models_{\top}^{\emptyset} x \\ &\models_{\top}^{\emptyset} \lambda z. x \\ &\models_{\top}^{\emptyset} \lambda z. \lambda z. x \\ &\models_{\top}^{\emptyset} \lambda z. \lambda z. \lambda z. x \quad \dots \end{aligned}$$

23/60

## Unfoldings: Examples (continued)

Recall:

$$\frac{(t, \hat{\rho}) \models_{\top}^{\{x\} \cup X} t'}{(\lambda x. t, \hat{\rho}) \models_{\top}^X \lambda x. t'} \quad \frac{t \in \hat{\rho}(x) \quad (t, \hat{\rho}) \models_{\top}^{\emptyset} t' \quad X \cap (\{x\} \cup \text{fv } t') = \emptyset}{(x, \hat{\rho}) \models_{\top}^X t'}$$

$$\begin{aligned} (x, \{x \mapsto \{\lambda y. y\}; y \mapsto \{\lambda z. z\}\}) &\models_{\top}^{\emptyset} x \\ &\not\models_{\top}^{\emptyset} \lambda z. z \quad \text{y has escaped!} \end{aligned}$$

$$\begin{aligned} (x, \{x \mapsto \{\lambda y. x_2\}; x_2 \mapsto \{y\}; y \mapsto \{\lambda z. z, \lambda z. y\}\}) &\models_{\top}^{\emptyset} x \\ &\models_{\top}^{\emptyset} \lambda y. x_2 \\ &\not\models_{\top}^{\emptyset} \lambda y. y \quad \text{y is captured!} \\ &\models_{\top}^{\emptyset} \lambda y. \lambda z. z \\ &\not\models_{\top}^{\emptyset} \lambda y. \lambda z. y \quad \text{y is captured!} \\ &\models_{\top}^{\emptyset} \lambda y. \lambda z. \lambda z. z \end{aligned}$$

24/60

## Abstract environments

- ▶  $(\hat{v}, \hat{\rho}) \models_v v$  means “value  $v$  is obtained from  $\hat{v}$  by unfolding definitions of  $\hat{\rho}$ ”

$$\frac{v \in \hat{v} \quad (v, \hat{\rho}) \models_{\top}^{\emptyset} v' \quad \text{fv } v' = \emptyset}{(\hat{v}, \hat{\rho}) \models_v v'}$$

- ▶ An environment  $\rho$  maps variables to *closed* values (i.e., with no free variables)
- ▶  $\hat{\rho} \models_E \rho$  means “environment  $\rho$  is obtained by unfolding definitions of  $\hat{\rho}$ ”

$$\frac{\text{dom } \rho \subseteq \text{dom } \hat{\rho} \quad \forall x \in \text{dom } \rho, (\hat{\rho}(x), \hat{\rho}) \models_v \rho(x)}{\hat{\rho} \models_E \rho}$$

- ☞ An abstract environment denotes a set of concrete environments

25/60

## Examples

$$\begin{aligned} \{x \mapsto \{\lambda z. z, \lambda y_1. \lambda y_2. y_1\}\} &\models_E \{\} \\ &\models_E \{x \mapsto \lambda z. z\} \\ &\models_E \{x \mapsto \lambda y_1. \lambda y_2. y_1\} \end{aligned}$$

$$\begin{aligned} \{x \mapsto \{\lambda z. z, \lambda y. x\}\} &\models_E \{\} \\ &\models_E \{x \mapsto \lambda z. z\} \\ &\models_E \{x \mapsto \lambda y. \lambda z. z\} \\ &\models_E \{x \mapsto \lambda y. \lambda y. \lambda z. z\} \\ &\models_E \dots \end{aligned}$$

$$\begin{aligned} \{x \mapsto \emptyset; y \mapsto \{\lambda z. z\}\} &\models_E \{\} \\ &\models_E \{y \mapsto \lambda z. z\} \end{aligned}$$

$$\begin{aligned} \{x \mapsto \{\lambda z. x\}; y \mapsto \{\lambda z. z\}\} &\models_E \{\} \\ &\models_E \{y \mapsto \lambda z. z\} \end{aligned}$$

26/60



## Soundness 0-CFA

An environment acts on a term by substituting its free variables:

$$x[\rho] \triangleq \begin{cases} \rho(x) & \text{if } x \in \text{dom } \rho \\ x & \text{otherwise} \end{cases} \quad \begin{aligned} (\lambda x. t)[\rho] &\triangleq \lambda x. (t[\rho \setminus x]) \\ (t_1 t_2)[\rho] &\triangleq (t_1[\rho]) (t_2[\rho]) \\ (t)^p[\rho] &\triangleq (t[\rho])^p \end{aligned}$$

### Theorem 5 (Soundness)

If  $(\hat{C}, \hat{\rho}) \models (t)^p$  and  $\hat{\rho} \models_E \rho$  and  $\text{fv } t \subseteq \text{dom } \rho$  and  $t[\rho] \rightarrow^* v$ , then  $(\hat{C}(\rho), \hat{\rho}) \models_V v$

### Informal interpretation:

“if  $(\hat{C}, \hat{\rho})$  is a valid 0-CFA solution for the program  $(t)^p$ ,  
if we replace the *inputs* (i.e., the free variables) of  $t$  with values allowed by  $\hat{\rho}$ ,  
then the value produced by  $t$  belongs to the set of possible values for the label  $p$ ”

### Corollary 6

If  $(\hat{C}, \hat{\rho}) \models (t)^p$  and  $\text{fv } t = \emptyset$  and  $t \rightarrow^* v$ , then  $(\hat{C}(\rho), \hat{\rho}) \models_V v$

27/60

## Soundness of 0-CFA (continued)

The structure of the proof is similar to the soundness proof for the simply-typed  $\lambda$ -calculus (based preservation and progress)

### Lemma 7 (Acceptability is invariant by reduction)

Assume  $(\hat{C}, \hat{\rho}) \models (t_1)^p$  and  $(t_1, \hat{\rho}) \models_{\top}^{\emptyset} t'_1$ :

If  $t'_1 \rightarrow t'_2$ , then there exists  $t_2$  such that  $(\hat{C}, \hat{\rho}) \models (t_2)^p$  and  $(t_2, \hat{\rho}) \models_{\top}^{\emptyset} t'_2$

### Lemma 8 (Value acceptability)

Assume  $(\hat{C}, \hat{\rho}) \models (t)^p$  and  $(t, \hat{\rho}) \models_{\top}^{\emptyset} v$  and  $\text{fv } v = \emptyset$ . Then,  $(\hat{C}(\rho), \hat{\rho}) \models_V v$

28/60

## Example for Lemma 7

$$\hat{\rho}(x_1) = \{ \lambda y. (\lambda z. y^3)^4 \}$$

$$\hat{\rho}(x_2) = \emptyset$$

$$\hat{\rho}(y) = \{ \lambda x_2. x_2^7 \}$$

$$\hat{\rho}(z) = \emptyset$$

$$\hat{C}(9) = \{ \lambda z. y^3 \}$$

$$t'_1 = \left( \left( (\lambda x_1. x_1^2) (\lambda y. (\lambda z. y^3)^4)^5 \right)^6 (\lambda x_2. x_2^7)^8 \right)^9$$

$$(t'_1, \hat{\rho}) \models_{\top}^{\emptyset} t'_1$$

$$\rightarrow t'_2 = \left( (\lambda y. (\lambda z. y^3)^4)^6 (\lambda x_2. x_2^7)^8 \right)^9$$

$$\left( (x_1^6 (\lambda x_2. x_2^7)^8)^9, \hat{\rho} \right) \models_{\top}^{\emptyset} t'_2$$

$$\rightarrow t'_3 = (\lambda z. (\lambda x_2. x_2^7)^3)^9$$

$$\left( (\lambda z. y^3)^9, \hat{\rho} \right) \models_{\top} t'_3$$

and by Lemma 8:

$$(\hat{C}(9), \hat{\rho}) \models_V t'_3$$

29/60

## Some key lemmas for the proof of soundness for 0-CFA

### Lemma 9 (Acceptability weakening)

If  $(\hat{C}, \hat{\rho}) \models (t)^{p_1}$  and  $\hat{C}(p_1) \subseteq \hat{C}(p_2)$ , then  $(\hat{C}, \hat{\rho}) \models (t)^{p_2}$

### Lemma 10 (Value denotation weakening)

If  $(\hat{v}_1, \hat{\rho}) \models_V v$  and  $\hat{v}_1 \subseteq \hat{v}_2$  then  $(\hat{v}_2, \hat{\rho}) \models_V v$

### Lemma 11 (Term denotation weakening)

If  $(t, \hat{\rho}) \models_{\top}^{X_1} t'$  and  $X_2 \subseteq X_1$  then  $(t, \hat{\rho}) \models_{\top}^{X_2} t'$

### Lemma 12 (Term denotation substitution)

If  $(t, \hat{\rho}) \models_{\top}^X t_1$  and  $(\hat{\rho}(x), \hat{\rho}) \models_V t_2$  and  $(t, \hat{\rho}) \models_{\top}^{X \setminus \{x\}} t_1[x \leftarrow t_2]$

### Lemma 13 (Matching lemma)

If  $\hat{\rho} \models_E \rho$  and  $X \cap \text{dom } \rho = \emptyset$ , then  $(t, \hat{\rho}) \models_{\top}^X t[\rho]$

30/60

## 0-CFA as Abstract Interpretation

## 0-CFA by Abstract Interpretation: What For?

### Theorem 5 (Soundness)

If  $(\hat{C}, \hat{\rho}) \models (t)^p$  and  $\hat{\rho} \models_{\text{E}} \rho$  and  $\text{fv } t \subseteq \text{dom } \rho$  and  $t[\rho] \rightarrow^* v$ , then  $(\hat{C}(p), \hat{\rho}) \models_v v$

Some shortcomings about the standard presentation of 0-CFA:

- ▶ The soundness theorem only talks about the *final* value!
- ▶ The soundness theorem does not tell explicitly that  $\hat{\rho}$  describes the possible instantiations of the variables that occur during evaluation!
- ▶ The rules give no explicit method to compute a solution
- ▶ There is no explanation of how to obtain the correct rules

**Abstract Interpretation:** A *systematic* approach to design a static analysis:

- ▶ Is rooted in semantics
- ▶ Is guided by abstractions, in the form of Galois connections:

$$(A, \sqsubseteq) \xleftrightarrow[\alpha]{\gamma} (A^\#, \sqsubseteq^\#)$$

- ▶ Gives satisfactory answers to the 4 remarks from above

31/60

## Small-step semantics

The small-step semantics is a *labelled transition system* (LTS)

It produces a **trace** of events of the form  $t_1 \xrightarrow{tr} t_2$  that describes:

- ▶ which  $\beta$ -reductions have happened:  $\beta(\lambda x. t, v)$
- ▶ which values were produced by a program point:  $\text{return}(p, v)$

The trace represents the *history* of the evaluation

$$\begin{array}{l} \text{BETAV} \frac{}{(\lambda x. t) v \xrightarrow{\beta(\lambda x. t, v)} t[x \leftarrow v]} \quad \text{ANNOT} \frac{}{(v)^p \xrightarrow{\text{return}(p, v)} v} \\ \text{APPCTXTL} \frac{t_1 \xrightarrow{e} t'_1}{t_1 t_2 \xrightarrow{e} t'_1 t_2} \quad \text{APPCTXTR} \frac{t \xrightarrow{e} t'}{vt \xrightarrow{e} vt'} \quad \text{ANNOTCTX} \frac{t \xrightarrow{e} t'}{(t)^p \xrightarrow{e} (t')^p} \\ \text{REFL} \frac{}{t \xrightarrow{\varepsilon} t} \quad \text{STEP} \frac{t_1 \xrightarrow{e} t_2 \quad t_2 \xrightarrow{tr} t_3}{t_1 \xrightarrow{e+tr} t_3} \end{array}$$

32/60

## Exercises

### Exercise 6.1

Reduce to its normal form the following term

$$\left( \left( \left( \lambda f. (\lambda x. (f^1 x^2)^3)^4 \right)^5 (\lambda x. (\lambda y. x^6)^7)^8 \right)^9 ((\lambda y. y^{10})^{11} (\lambda z. z^{12})^{13})^{14} \right)^{15}$$

What are the differences with the first semantics we gave in this course?

33/60

## Roadmap

Abstract caches  
and environments

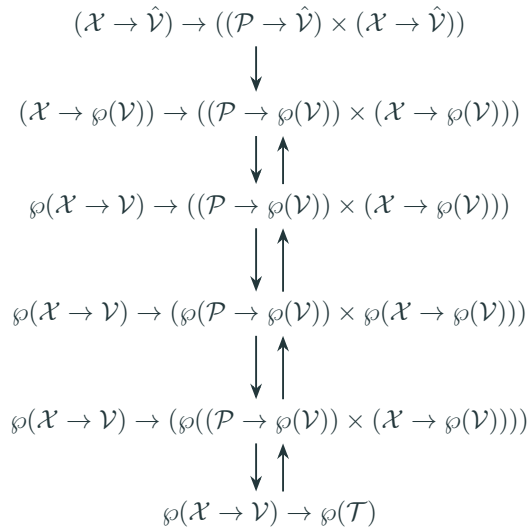
Environments as inputs

Caches and  
environments

Sets of caches and  
sets of environments

Sets of caches  
and environments

Sets of traces



34/60

## Collecting Semantics

Recall: environments are maps from variables to closed values

$$\begin{aligned}
 \text{Collecting semantics: } \mathbb{T}[\![t]\!] &: \{\rho \mid \text{fv } t \subseteq \text{dom } \rho\} \rightarrow \wp(\mathcal{T}) \\
 \mathbb{T}[\![t]\!](\mathcal{I}) &\triangleq \bigcup_{\rho \in \mathcal{I}} \{tr \mid t[\rho] \xrightarrow{tr} v\}
 \end{aligned}$$

The following inclusions are satisfied:

$$\mathbb{T}[\![x^p]\!](\mathcal{I}) = \bigcup_{\rho \in \mathcal{I}} \{\text{return}(\rho, \rho(x))\}$$

$$\mathbb{T}[\![\lambda x. t]^p]\!](\mathcal{I}) = \bigcup_{\rho \in \mathcal{I}} \{\text{return}(\rho, (\lambda x. t)[\rho])\}$$

$$\mathbb{T}[\![(t_1)^{p_1} (t_2)^{p_2}]^p]\!](\mathcal{I}) \subseteq \left\{ \begin{array}{l} tr_1 \uparrow \text{return}(\rho_1, \lambda x. (t_0)^{p_0}) \\ \uparrow tr_2 \uparrow \text{return}(\rho_2, v_2) \\ \uparrow \beta(\lambda x. (t_0)^{p_0}, v_2) \\ \uparrow tr_0 \uparrow \text{return}(\rho_0, v_0) \\ \uparrow \text{return}(\rho, v_0) \end{array} \middle| \begin{array}{l} tr_1 \uparrow \text{return}(\rho_1, \lambda x. (t_0)^{p_0}) \in \mathbb{T}[\![(t_1)^{p_1}]\!](\mathcal{I}), \\ tr_2 \uparrow \text{return}(\rho_2, v_2) \in \mathbb{T}[\![(t_2)^{p_2}]\!](\mathcal{I}), \\ tr_0 \uparrow \text{return}(\rho_0, v_0) \in \mathbb{T}[\![(t_0)^{p_0}]\!](\mathcal{I}[x \mapsto \{v_2\}]) \end{array} \right\}$$

where  $\mathcal{I}[x \mapsto S] = \{\rho[x \mapsto v] \mid \rho \in \mathcal{I}, v \in S\}$

35/60

## Roadmap

Abstract caches  
and environments

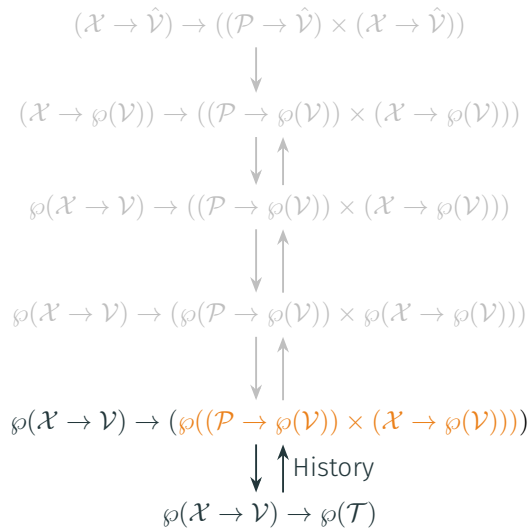
Environments as inputs

Caches and  
environments

Sets of caches and  
sets of environments

Sets of caches  
and environments

Sets of traces



36/60

## From Traces to Caches and Environments

Set-valued functions form a complete lattice:

$$\begin{aligned}
 \perp &\triangleq \lambda k. \emptyset \\
 \{k \mapsto S\} &\triangleq \lambda k'. \text{if } k' = k \text{ then } S \text{ else } \emptyset \\
 m_1 \dot{\cup} m_2 &\triangleq \lambda k. m_1(k) \cup m_2(k) \\
 m_1 \dot{\subseteq} m_2 &\triangleq \forall k, m_1(k) \subseteq m_2(k)
 \end{aligned}$$

Retrieving caches:

$$\begin{aligned}
 \text{cache}(\beta(\lambda x. t, v)) &\triangleq \perp & \text{cache}(\varepsilon) &\triangleq \perp \\
 \text{cache}(\text{return}(\rho, v)) &\triangleq \{\rho \mapsto \{v\}\} & \text{cache}(tr_1 \uparrow tr_2) &\triangleq \text{cache}(tr_1) \dot{\cup} \text{cache}(tr_2)
 \end{aligned}$$

Retrieving environments:

$$\begin{aligned}
 \text{env}(\beta(\lambda x. t, v)) &\triangleq \{x \mapsto \{v\}\} & \text{env}(\varepsilon) &\triangleq \perp \\
 \text{env}(\text{return}(\rho, v)) &\triangleq \perp & \text{env}(tr_1 \uparrow tr_2) &\triangleq \text{env}(tr_1) \dot{\cup} \text{env}(tr_2)
 \end{aligned}$$

37/60

## History Abstraction

$$\alpha_h : \wp(\mathcal{T}) \rightarrow \wp(\mathcal{C} \times \mathcal{E})$$

$$\alpha_h(S) \triangleq \{(\text{cache}(tr), \text{env}(tr)) \mid tr \in S\}$$

$$\gamma_h : \wp(\mathcal{C} \times \mathcal{E}) \rightarrow \wp(\mathcal{T})$$

$$\gamma_h(S) \triangleq \{tr \mid \text{env } tr \sqsubseteq \hat{\rho}, \text{ cache } tr \sqsubseteq \hat{C}, (\hat{C}, \hat{\rho}) \in S\}$$

$$S_1 \sqsubseteq_h S_2 \triangleq \forall (\hat{C}_1, \hat{\rho}_1) \in S_1, \exists (\hat{C}_2, \hat{\rho}_2) \in S_2, \hat{C}_1 \subseteq \hat{C}_2 \wedge \hat{\rho}_1 \subseteq \hat{\rho}_2$$

We have a Galois connection:

$$(\wp(\mathcal{T}), \subseteq) \xrightleftharpoons[\alpha_h]{\gamma_h} (\wp(\mathcal{C} \times \mathcal{E}), \sqsubseteq_h)$$

38/60

## Abstract Semantics After History Abstraction

$$\text{Definition: } \mathbb{T}_h^\# \llbracket t \rrbracket \triangleq \alpha_h \circ \mathbb{T} \llbracket t \rrbracket$$

The following inclusions are satisfied:

$$\mathbb{T}_h^\# \llbracket x^p \rrbracket (\mathcal{I}) = \{(\{\rho \mapsto \{\rho(x)\}\}, \perp) \mid \rho \in \mathcal{I}\}$$

$$\mathbb{T}_h^\# \llbracket (\lambda x. t)^p \rrbracket (\mathcal{I}) = \{(\{\rho \mapsto \{(\lambda x. t)[\rho]\}\}, \perp) \mid \rho \in \mathcal{I}\}$$

$$\mathbb{T}_h^\# \llbracket ((t_1)^{p_1} (t_2)^{p_2})^p \rrbracket (\mathcal{I}) \sqsubseteq_h$$

$$\left\{ \begin{array}{l} (\hat{C}_1, \hat{\rho}_1) \in \mathbb{T}_h^\# \llbracket (t_1)^{p_1} \rrbracket (\mathcal{I}), \\ (\hat{C}_2, \hat{\rho}_2) \in \mathbb{T}_h^\# \llbracket (t_2)^{p_2} \rrbracket (\mathcal{I}), \\ \hat{\rho}_1 \cup \hat{\rho}_2 \cup \{x \mapsto \hat{C}_2(p_2)\} \cup \hat{\rho}_0 \\ \hat{C}_0, \hat{\rho}_0 \in \mathbb{T}_h^\# \llbracket (t_0)^{p_0} \rrbracket (\mathcal{I}[x \mapsto \hat{C}_2(p_2)]) \end{array} \right\}$$

$$\text{where } \mathcal{I}[x \mapsto S] = \{\rho[x \mapsto v] \mid \rho \in \mathcal{I}, v \in S\}$$

39/60

## Roadmap

Abstract caches  
and environments

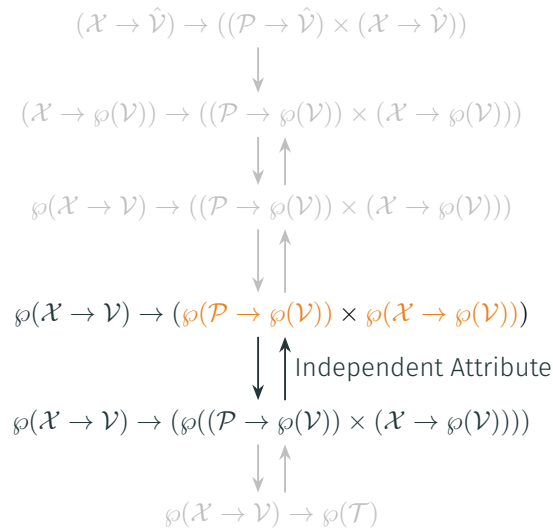
Environments as inputs

Caches and  
environments

Sets of caches and  
sets of environments

Sets of caches  
and environments

Sets of traces



40/60

## Independent Attribute Abstraction

$$\alpha_{ia} : \wp(A \times B) \rightarrow \wp(A) \times \wp(B)$$

$$\alpha_{ia}(S) \triangleq (\{a \mid (a, b) \in S\}, \{b \mid (a, b) \in S\})$$

$$\gamma_{ia} : \wp(A) \times \wp(B) \rightarrow \wp(A \times B)$$

$$\gamma_{ia}(S_A, S_B) \triangleq S_A \times S_B$$

$$(S_A, S_B) \sqsubseteq_{ia} (S'_A, S'_B) \triangleq S_A \subseteq S'_A \wedge S_B \subseteq S'_B$$

We have a Galois connection:

$$(\wp(A \times B), \subseteq) \xrightleftharpoons[\alpha_{ia}]{\gamma_{ia}} (\wp(A) \times \wp(B), \sqsubseteq_{ia})$$

41/60

## Composing Abstractions on Pairs

Given two Galois connections:

$$(A^b, \sqsubseteq_{A^b}) \xleftrightarrow[\alpha_A]{\gamma_A} (A^\#, \sqsubseteq_{A^\#}) \quad \text{and} \quad (B^b, \sqsubseteq_{B^b}) \xleftrightarrow[\alpha_B]{\gamma_B} (B^\#, \sqsubseteq_{B^\#})$$

$$\begin{aligned} \langle \alpha_A, \alpha_B \rangle & : A^b \times B^b \rightarrow A^\# \times B^\# \\ \langle \alpha_A, \alpha_B \rangle (a^b, b^b) & \triangleq (\alpha_A(a^b), \alpha_B(b^b)) \end{aligned}$$

$$\begin{aligned} \langle \gamma_A, \gamma_B \rangle & : A^\# \times B^\# \rightarrow A^b \times B^b \\ \langle \gamma_A, \gamma_B \rangle (a^\#, b^\#) & \triangleq (\gamma_A(a^\#), \gamma_B(b^\#)) \end{aligned}$$

$$(a_1^b, b_1^b) \sqsubseteq_{A^b \times B^b} (a_2^b, b_2^b) \triangleq a_1^b \sqsubseteq_{A^b} a_2^b \wedge b_1^b \sqsubseteq_{B^b} b_2^b$$

$$(a_1^\#, b_1^\#) \sqsubseteq_{A^\# \times B^\#} (a_2^\#, b_2^\#) \triangleq a_1^\# \sqsubseteq_{A^\#} a_2^\# \wedge b_1^\# \sqsubseteq_{B^\#} b_2^\#$$

We have a Galois connection:

$$(A^b \times B^b, \sqsubseteq_{A^b \times B^b}) \xleftrightarrow[\langle \alpha_A, \alpha_B \rangle]{\langle \gamma_A, \gamma_B \rangle} (A^\# \times B^\#, \sqsubseteq_{A^\# \times B^\#})$$

42/60

## Abstract Semantics After Independent Attribute Abstraction

**Definition:**

$$\mathbb{T}_{ia}^\# \llbracket t \rrbracket \triangleq \langle \alpha_{ia}, \alpha_{ia} \rangle \circ \mathbb{T}_h^\# \llbracket t \rrbracket$$

The following inclusions are satisfied:

$$\mathbb{T}_{ia}^\# \llbracket x^p \rrbracket (\mathcal{I}) = (\{\{\rho \mapsto \{\rho(x)\}\} \mid \rho \in \mathcal{I}\}, \{\perp\})$$

$$\mathbb{T}_{ia}^\# \llbracket (\lambda x. t)^p \rrbracket (\mathcal{I}) = (\{\{\rho \mapsto \{(\lambda x. t)[\rho]\}\} \mid \rho \in \mathcal{I}\}, \{\perp\})$$

$$\begin{aligned} \mathbb{T}_{ia}^\# \llbracket ((t_1)^{p_1} (t_2)^{p_2})^p \rrbracket (\mathcal{I}) \subseteq \\ \text{let } (\hat{C}_1, \hat{\rho}_1) = \mathbb{T}_{ia}^\# \llbracket (t_1)^{p_1} \rrbracket (\mathcal{I}) \text{ in} \\ \text{let } (\hat{C}_2, \hat{\rho}_2) = \mathbb{T}_{ia}^\# \llbracket (t_2)^{p_2} \rrbracket (\mathcal{I}) \text{ in} \\ (\hat{C}_1, \hat{\rho}_1) \sqcup (\hat{C}_2, \hat{\rho}_2) \sqcup \\ \bigsqcup_{\lambda x. (t_0)^{p_0} \in \hat{C}_1(\rho_1)} \text{let } (\hat{C}_0, \hat{\rho}_0) = \mathbb{T}_{ia}^\# \llbracket (t_0)^{p_0} \rrbracket (\mathcal{I}[x \mapsto \hat{C}_2(\rho_2)]) \text{ in} \\ (\hat{C}_0, \hat{\rho}_0) \sqcup (\{\{\rho \mapsto \hat{C}_0(\rho_0)\}\}, \{\{x \mapsto \hat{C}_2(\rho_2)\}\}) \end{aligned}$$

43/60

## Roadmap

Abstract caches  
and environments

$$(\mathcal{X} \rightarrow \hat{\mathcal{V}}) \rightarrow ((\mathcal{P} \rightarrow \hat{\mathcal{V}}) \times (\mathcal{X} \rightarrow \hat{\mathcal{V}}))$$

Environments as inputs

$$(\mathcal{X} \rightarrow \wp(\mathcal{V})) \rightarrow ((\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V})))$$

Caches and  
environments

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow ((\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V})))$$

Sets of caches and  
sets of environments

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow (\wp(\mathcal{P} \rightarrow \wp(\mathcal{V})) \times \wp(\mathcal{X} \rightarrow \wp(\mathcal{V})))$$

Sets of caches  
and environments

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow (\wp((\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V}))))$$

Sets of traces

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow \wp(\mathcal{T})$$

Map independent attribute

44/60

## Map Abstraction

This is similar to independent attribute on pairs

$$\alpha_m : \wp(A \rightarrow \wp(B)) \rightarrow (A \rightarrow \wp(B))$$

$$\alpha_m(S) \triangleq \lambda a. \bigcup_{f \in S} f(a)$$

$$\gamma_m : (A \rightarrow \wp(B)) \rightarrow \wp(A \rightarrow \wp(B))$$

$$\gamma_m(f) \triangleq \{g \mid \forall a. g(a) \subseteq f(a)\}$$

We have a Galois connection:

$$(\wp(A \rightarrow \wp(B)), \subseteq) \xleftrightarrow[\alpha_m]{\gamma_m} (A \rightarrow \wp(B), \dot{\subseteq})$$

45/60

## Abstract Semantics after Map Independent Attribute Abstraction

### Definition:

$$\mathbb{T}_M^\# \llbracket t \rrbracket \triangleq \langle \alpha_m, \alpha_m \rangle \circ \mathbb{T}_h^\# \llbracket t \rrbracket : (\wp(\mathcal{X} \rightarrow \mathcal{V})) \rightarrow (\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V}))$$

The following inclusions are satisfied:

$$\mathbb{T}_M^\# \llbracket \mathcal{X}^p \rrbracket (\mathcal{I}) = (\{\rho \mapsto \mathcal{I}(x)\}, \perp) \quad \text{where } \mathcal{I}(x) = \{\rho(x) \mid \rho \in \mathcal{I}\}$$

$$\mathbb{T}_M^\# \llbracket (\lambda x. t)^p \rrbracket (\mathcal{I}) = (\{\rho \mapsto (\lambda x. t)[\mathcal{I}]\}, \perp) \quad \text{where } t[\mathcal{I}] = \{t[\rho] \mid \rho \in \mathcal{I}\}$$

$$\mathbb{T}_M^\# \llbracket ((t_1)^{p_1} (t_2)^{p_2})^p \rrbracket (\mathcal{I}) \sqsubseteq$$

$$\text{let } (\hat{\mathcal{C}}_1, \hat{\rho}_1) = \mathbb{T}_M^\# \llbracket (t_1)^{p_1} \rrbracket (\mathcal{I}) \text{ in}$$

$$\text{let } (\hat{\mathcal{C}}_2, \hat{\rho}_2) = \mathbb{T}_M^\# \llbracket (t_2)^{p_2} \rrbracket (\mathcal{I}) \text{ in}$$

$$(\hat{\mathcal{C}}_1, \hat{\rho}_1) \sqcup (\hat{\mathcal{C}}_2, \hat{\rho}_2) \sqcup$$

$$\bigsqcup_{\lambda x. (t_0)^{p_0} \in \mathcal{C}_1(p_1)} \text{let } (\hat{\mathcal{C}}_0, \hat{\rho}_0) = \mathbb{T}_M^\# \llbracket (t_0)^{p_0} \rrbracket (\mathcal{I}[x \mapsto \hat{\mathcal{C}}_2(p_2)]) \text{ in} \\ (\hat{\mathcal{C}}_0, \hat{\rho}_0) \sqcup (\{\rho \mapsto \hat{\mathcal{C}}_0(\rho_0)\}, \{x \mapsto \hat{\mathcal{C}}_2(p_2)\})$$

46/60

## Roadmap

Abstract caches  
and environments

$$(\mathcal{X} \rightarrow \hat{\mathcal{V}}) \rightarrow ((\mathcal{P} \rightarrow \hat{\mathcal{V}}) \times (\mathcal{X} \rightarrow \hat{\mathcal{V}}))$$

Environments as inputs

$$(\mathcal{X} \rightarrow \wp(\mathcal{V})) \rightarrow ((\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V})))$$

Caches and  
environments

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow ((\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V})))$$

Sets of caches and  
sets of environments

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow (\wp(\mathcal{P} \rightarrow \wp(\mathcal{V})) \times \wp(\mathcal{X} \rightarrow \wp(\mathcal{V})))$$

Sets of caches  
and environments

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow (\wp((\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V}))))$$

Sets of traces

$$\wp(\mathcal{X} \rightarrow \mathcal{V}) \rightarrow \wp(\mathcal{T})$$

47/60

## From sets of functions to set-valued functions

### Definition:

$$\alpha_{m_1} : \wp(A \rightarrow B) \rightarrow (A \rightarrow \wp(B))$$

$$\alpha_{m_1}(S) \triangleq \lambda a. \bigcup_{f \in S} \{f(a)\}$$

$$\gamma_{m_1} : (A \rightarrow \wp(B)) \rightarrow \wp(A \rightarrow B)$$

$$\gamma_{m_1}(f) \triangleq \{g \mid \forall a. g(a) \in f(a)\}$$

We have a Galois connection:

$$(\wp(A \rightarrow B), \subseteq) \xleftrightarrow[\alpha_{m_1}]{\gamma_{m_1}} (A \rightarrow \wp(B), \subseteq)$$

48/60

## Abstract semantics after input abstraction: $\infty$ CFA

### Definition:

$$\mathbb{T}_{\infty\text{CFA}}^\# \llbracket t \rrbracket \triangleq \mathbb{T}_M^\# \llbracket t \rrbracket \circ \gamma_{m_1} : (\mathcal{X} \rightarrow \wp(\mathcal{V})) \rightarrow (\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V}))$$

The following inclusions are satisfied:

$$\mathbb{T}_{\infty\text{CFA}}^\# \llbracket \mathcal{X}^p \rrbracket (\hat{\rho}) = (\{\rho \mapsto \hat{\rho}(x)\}, \perp)$$

$$\mathbb{T}_{\infty\text{CFA}}^\# \llbracket (\lambda x. t)^p \rrbracket (\hat{\rho}) = (\{\rho \mapsto (\lambda x. t)[\gamma_{m_1}(\hat{\rho})]\}, \perp) \quad \text{where } t[\mathcal{I}] = \{t[\rho] \mid \rho \in \mathcal{I}\}$$

$$\mathbb{T}_{\infty\text{CFA}}^\# \llbracket ((t_1)^{p_1} (t_2)^{p_2})^p \rrbracket (\hat{\rho}) \sqsubseteq$$

$$\text{let } (\hat{\mathcal{C}}_1, \hat{\rho}_1) = \mathbb{T}_{\infty\text{CFA}}^\# \llbracket (t_1)^{p_1} \rrbracket (\hat{\rho}) \text{ in}$$

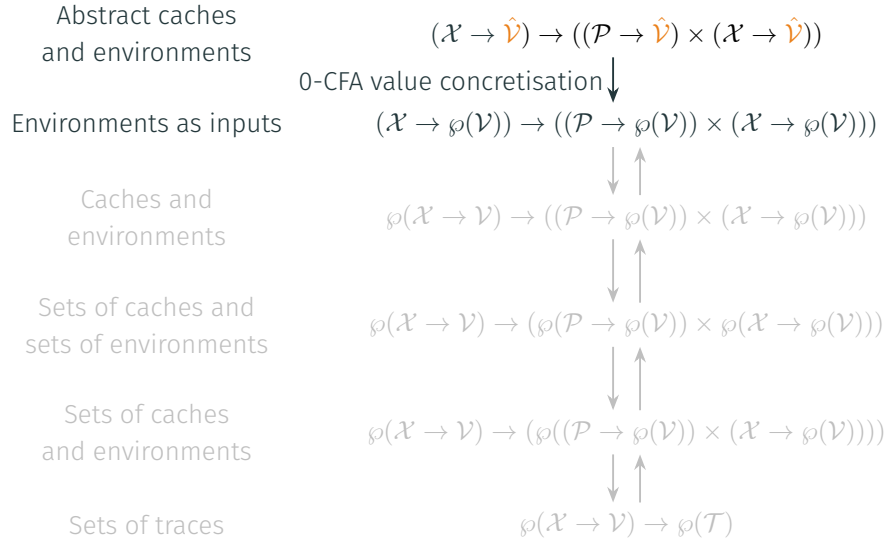
$$\text{let } (\hat{\mathcal{C}}_2, \hat{\rho}_2) = \mathbb{T}_{\infty\text{CFA}}^\# \llbracket (t_2)^{p_2} \rrbracket (\hat{\rho}) \text{ in}$$

$$(\hat{\mathcal{C}}_1, \hat{\rho}_1) \sqcup (\hat{\mathcal{C}}_2, \hat{\rho}_2) \sqcup$$

$$\bigsqcup_{\lambda x. (t_0)^{p_0} \in \hat{\mathcal{C}}_1(p_1)} \text{let } (\hat{\mathcal{C}}_0, \hat{\rho}_0) = \mathbb{T}_{\infty\text{CFA}}^\# \llbracket (t_0)^{p_0} \rrbracket (\hat{\rho}[x \mapsto \hat{\mathcal{C}}_2(p_2)]) \text{ in} \\ (\hat{\mathcal{C}}_0, \hat{\rho}_0) \sqcup (\{\rho \mapsto \hat{\mathcal{C}}_0(\rho_0)\}, \{x \mapsto \hat{\mathcal{C}}_2(p_2)\})$$

49/60

## Roadmap



50/60

## 0-CFA value abstraction

We reuse the unfolding relation from the first part of the course:

$$\begin{aligned}
 \gamma_{0\text{CFA}}^{\hat{\rho}} & : \wp(\mathcal{V}) \rightarrow \wp(\mathcal{V} \cap \mathbf{Terms}(t)) \\
 \gamma_{0\text{CFA}}^{\hat{\rho}}(S) & \triangleq \{V' \mid (v, \hat{\rho}) \models_V V', v \in S\}
 \end{aligned}$$

$$S_1 \subseteq S_2 \Rightarrow \gamma_{0\text{CFA}}^{\hat{\rho}}(S_1) \subseteq \gamma_{0\text{CFA}}^{\hat{\rho}}(S_2)$$

There is no best abstraction!

**Example:** Take  $\hat{\rho} = \{x_1 \mapsto \lambda y. y; x_2 \mapsto \lambda y. y\}$

We have several minimal solutions to abstract the set  $\{\lambda z. \lambda y. y\}$ :

- ▶  $\gamma_{0\text{CFA}}^{\hat{\rho}}\{\lambda z. x_1\}$
- ▶  $\gamma_{0\text{CFA}}^{\hat{\rho}}\{\lambda z. x_2\}$
- ▶  $\gamma_{0\text{CFA}}^{\hat{\rho}}\{\lambda z. \lambda y. y\}$

But they are not comparable!

51/60

## Abstract Semantics for 0-CFA

We want:

$$\begin{aligned}
 \mathbb{T}_{0\text{CFA}}^{\#} \llbracket t \rrbracket & : (\mathcal{X} \rightarrow \hat{\mathcal{V}}) \rightarrow ((\mathcal{P} \rightarrow \hat{\mathcal{V}}) \times (\mathcal{X} \rightarrow \hat{\mathcal{V}})) \\
 \text{such that: } \mathbb{T}_{0\text{CFA}}^{\#} \llbracket t \rrbracket \circ \gamma_{0\text{CFA}}^{\hat{\rho}} & \sqsubseteq \langle \gamma_{0\text{CFA}}^{\hat{\rho}}, \gamma_{0\text{CFA}}^{\hat{\rho}} \rangle \circ \mathbb{T}_{0\text{CFA}}^{\#} \llbracket t \rrbracket(\hat{\rho})
 \end{aligned}$$

It suffices that the following inclusions are satisfied:

$$\mathbb{T}_{0\text{CFA}}^{\#} \llbracket x^p \rrbracket(\hat{\rho}) = (\{p \mapsto \hat{\rho}(x)\}, \perp)$$

$$\mathbb{T}_{0\text{CFA}}^{\#} \llbracket (\lambda x. t)^p \rrbracket(\hat{\rho}) = (\{p \mapsto \lambda x. t\}, \perp)$$

$$\begin{aligned}
 \mathbb{T}_{0\text{CFA}}^{\#} \llbracket ((t_1)^{p_1} (t_2)^{p_2})^p \rrbracket(\hat{\rho}) & \sqsubseteq \\
 \text{let } (\hat{C}_1, \hat{\rho}_1) & = \mathbb{T}_{0\text{CFA}}^{\#} \llbracket (t_1)^{p_1} \rrbracket(\hat{\rho}) \text{ in} \\
 \text{let } (\hat{C}_2, \hat{\rho}_2) & = \mathbb{T}_{0\text{CFA}}^{\#} \llbracket (t_2)^{p_2} \rrbracket(\hat{\rho}) \text{ in} \\
 (\hat{C}_1, \hat{\rho}_1) \sqcup (\hat{C}_2, \hat{\rho}_2) \sqcup \\
 \bigsqcup_{\lambda x. (t_0)^{p_0} \in \hat{C}_1(p_1)} \text{let } (\hat{C}_0, \hat{\rho}_0) & = \mathbb{T}_{0\text{CFA}}^{\#} \llbracket (t_0)^{p_0} \rrbracket(\hat{\rho} \dot{\cup} \{x \mapsto \hat{C}_2(p_2)\}) \text{ in} \\
 (\hat{C}_0, \hat{\rho}_0) \sqcup (\{p \mapsto \hat{C}_0(p_0)\}, \{x \mapsto \hat{C}_2(p_2)\})
 \end{aligned}$$

52/60

## Definition of 0-CFA

**Definition:**

$$\begin{aligned}
 \mathbb{T}^{\#} \llbracket \cdot \rrbracket & : \mathcal{T} \rightarrow (\mathcal{X} \rightarrow \wp(\mathcal{V})) \rightarrow ((\mathcal{P} \rightarrow \wp(\mathcal{V})) \times (\mathcal{X} \rightarrow \wp(\mathcal{V}))) \\
 & \triangleq \text{Ifp } \lambda f. \lambda t. \lambda \hat{\rho}. \\
 & \text{match } t \text{ with} \\
 & | x^p \Rightarrow (\{p \mapsto \hat{\rho}(x)\}, \perp) \\
 & | (\lambda x. t_0)^p \Rightarrow (\{p \mapsto \lambda x. t_0\}, \perp) \\
 & | ((t_1)^{p_1} (t_2)^{p_2})^p \Rightarrow \\
 & \quad \text{let } (\hat{C}_1, \hat{\rho}_1) = f((t_1)^{p_1})(\hat{\rho}) \text{ in} \\
 & \quad \text{let } (\hat{C}_2, \hat{\rho}_2) = f((t_2)^{p_2})(\hat{\rho}) \text{ in} \\
 & \quad (\hat{C}_1, \hat{\rho}_1) \sqcup (\hat{C}_2, \hat{\rho}_2) \sqcup \\
 & \quad \bigsqcup_{\lambda x. (t_0)^{p_0} \in \hat{C}_1(p_1)} \text{let } (\hat{\rho}_0) = f((t_0)^{p_0})(\hat{C}, \hat{\rho} \dot{\cup} \{x \mapsto \hat{C}_2(p_2)\}) \text{ in} \\
 & \quad (\hat{C}_0, \hat{\rho}_0) \sqcup (\{p \mapsto \hat{C}_0(p_0)\}, \{x \mapsto \hat{C}_2(p_2)\})
 \end{aligned}$$

Why does this definition make sense?

53/60

## Soundness of 0-CFA

### Proposition 6.1

$$\mathbb{T}_{\text{OCFA}}^\sharp[[t]](\perp) \subseteq \mathbb{T}^\sharp[[t]](\perp)$$

### Corollary 14

For the concretization  $\gamma$  that we have defined piece after piece:

$$\mathbb{T}[[t]](\emptyset) \subseteq \gamma(\mathbb{T}^\sharp[[t]](\perp))$$

- ▶ The proof follows from the soundness of all the abstractions we have used
- ▶  $\mathbb{T}^\sharp[[t]]$  computes an abstraction of the traces produced by  $t$ , i.e.:
  - ▶ Which  $\beta$ -reductions might have been performed, and
  - ▶ Which values might have been produced at each program point
- ▶  $\mathbb{T}^\sharp[[t]]$  is directly implementable!  
Fixpoint iteration builds a partial function of the so far computed results
- 👉 Source code available on the course web page

54/60

## Example #1

$$\left( \left( (\lambda x_1. x_1^4)^2 (\lambda y. (\lambda z. y^3)^4)^5 \right)^6 (\lambda x_2. x_2^7)^8 \right)^9$$

Changes at iteration 1: ([ 8 -> {  $\lambda x_2. x_2^7$  } ], [])

Changes at iteration 2: ([ 2 -> {  $\lambda x_1. x_1^4$  }; 5 -> {  $\lambda y. (\lambda z. y^3)^4$  } ], [])

Changes at iteration 3: ([], [  $x_1$  -> {  $\lambda y. (\lambda z. y^3)^4$  } ])

Changes at iteration 4: ([ 1 -> {  $\lambda y. (\lambda z. y^3)^4$  } ], [])

Changes at iteration 5: ([ 6 -> {  $\lambda y. (\lambda z. y^3)^4$  } ], [])

Changes at iteration 6: ([], [  $y$  -> {  $\lambda x_2. x_2^7$  } ])

Changes at iteration 7: ([ 4 -> {  $\lambda z. y^3$  } ], [])

Changes at iteration 8: ([ 9 -> {  $\lambda z. y^3$  } ], [])

Solution:

[ 1 -> {  $\lambda y. (\lambda z. y^3)^4$  }; 2 -> {  $\lambda x_1. x_1^4$  }; 3 ->  $\emptyset$ ;

4 -> {  $\lambda z. y^3$  }; 5 -> {  $\lambda y. (\lambda z. y^3)^4$  }; 6 -> {  $\lambda y. (\lambda z. y^3)^4$  };

7 ->  $\emptyset$ ; 8 -> {  $\lambda x_2. x_2^7$  }; 9 -> {  $\lambda z. y^3$  },

[  $x_1$  -> {  $\lambda y. (\lambda z. y^3)^4$  };  $x_2$  ->  $\emptyset$ ;  $y$  -> {  $\lambda x_2. x_2^7$  } ]

55/60

## Example #2

$$\left( (\lambda x. (x^1 x^2)^3)^4 (\lambda y. (y^5 y^6)^7)^8 \right)^9$$

Changes at iteration 1:

[ 4 -> {  $\lambda x. (x^1 x^2)^3$  }; 8 -> {  $\lambda y. (y^5 y^6)^7$  } ], []

Changes at iteration 2: ([], [  $x$  -> {  $\lambda y. (y^5 y^6)^7$  } ])

Changes at iteration 3:

[ 1 -> {  $\lambda y. (y^5 y^6)^7$  }; 2 -> {  $\lambda y. (y^5 y^6)^7$  } ], []

Changes at iteration 4: ([], [  $y$  -> {  $\lambda y. (y^5 y^6)^7$  } ])

Changes at iteration 5:

[ 5 -> {  $\lambda y. (y^5 y^6)^7$  }; 6 -> {  $\lambda y. (y^5 y^6)^7$  } ], []

Solution:

[ 1 -> {  $\lambda y. (y^5 y^6)^7$  }; 2 -> {  $\lambda y. (y^5 y^6)^7$  }; 3 ->  $\emptyset$ ;

4 -> {  $\lambda x. (x^1 x^2)^3$  }; 5 -> {  $\lambda y. (y^5 y^6)^7$  }; 6 -> {  $\lambda y. (y^5 y^6)^7$  };

7 ->  $\emptyset$ ; 8 -> {  $\lambda y. (y^5 y^6)^7$  }; 9 ->  $\emptyset$  ],

[  $x$  -> {  $\lambda y. (y^5 y^6)^7$  };  $y$  -> {  $\lambda y. (y^5 y^6)^7$  } ]

56/60

## Soundness of the Acceptability Relation

Define  $(\hat{C}, \hat{\rho}) \Vdash t \triangleq \mathbb{T}_{\text{OCFA}}^\sharp[[t]](\hat{\rho}) \subseteq (\hat{C}, \hat{\rho})$

Then, the following implications are satisfied:

$$\frac{\hat{\rho}(x) \subseteq \hat{C}(p)}{(\hat{C}, \hat{\rho}) \Vdash x^p} \qquad \frac{\{\lambda x. (t_0)^{p_0}\} \subseteq \hat{C}(p)}{(\hat{C}, \hat{\rho}) \Vdash (\lambda x. (t_0)^{p_0})^p}$$

$$\frac{\begin{array}{l} (\hat{C}, \hat{\rho}) \Vdash (t_1)^{p_1} \quad (\hat{C}, \hat{\rho}) \Vdash (t_2)^{p_2} \\ \forall (\lambda x. (t_0)^{p_0}) \in \hat{C}(p_1), \\ (\hat{C}, \hat{\rho}) \Vdash (t_0)^{p_0} \quad \wedge \quad \hat{C}(p_2) \subseteq \hat{\rho}(x) \quad \wedge \quad \hat{C}(p_0) \subseteq \hat{C}(p) \end{array}}{(\hat{C}, \hat{\rho}) \Vdash ((t_1)^{p_1} (t_2)^{p_2})^p}$$

These are the same rules as the ones of the acceptability relation!

👉 This gives a semantic proof of the soundness for the rules of acceptability

57/60



## Beyond 0-CFA

## Limitations of 0-CFA

- ▶ **Context insensitive:** 0-CFA does not distinguish different instances of variables

$$\left( \text{let } f = (\lambda x. x^1)^2 \text{ in } \left( (f^3 f^4)^5 (\lambda y. y^6)^7 \right)^8 \right)^9$$

With the version based on constraints, we have:

$$\hat{C}(9) = \{\lambda x. x^1, \lambda y. y^6\} \quad \leftarrow \text{this is imprecise!}$$

- ▶ **Remark:** with the version derived by abstract interpretation, we get:

$$\hat{C}(9) = \{\lambda y. y^6\}$$

- ▶ **Flow insensitive:** all parts of a program are analysed, even unreachable ones

$$\left( \text{let } \omega = \left( \left( \lambda x_1. (x_1^1 x_1^2)^3 \right)^4 \left( \lambda x_2. (x_2^5 x_2^6)^7 \right)^8 \right)^9 \text{ in } \right)^{17} \\ \left( \omega^{10} \left( (\lambda y. y^{11})^{12} (\lambda z. z^{13})^{14} \right)^{15} \right)^{16}$$

With both versions of the analysis, we obtain:  $\hat{\rho}(y) = \{\lambda z. z^{13}\}$

58/60

## Extensions

Many aspects were not covered in this course! To cite a few:

- ▶ Context sensitivity: *k*-CFA, polymorphic splitting, ...
- ▶ Reachability-based CFA
- ▶ Recursive functions
- ▶ Booleans, conditionals, arithmetic
- ▶ Algebraic datatypes
- ▶ Imperative features
- ▶ Object oriented features
- ▶ Laziness
- ▶ Non-structural control flow:
  - ▶ Exceptions
  - ▶ Continuations/`callcc`
  - ▶ Delimited continuations/coroutines
  - ▶ Effect handlers
- ▶ Concurrency
- ▶ Reflexion

59/60

## Open Questions for CFA

- ▶ Relational domains for CFA?
- ▶ Modular analysis?
- ▶ Scalable analysis?  
 $\mathcal{O}(n^3)$  does not scale ☹

Some ongoing research at Inria Rennes:

- ▶ [Salto](#) project: static analysis for OCaml programs  
Based on a variant of CFA, designed using abstract interpretation

60/60