Constraints as a Denotational Semantics for Object Calculus

Midterm presentation

Thomas Sixuan Lou

February 1, 2017

Project Overview & Motivations

Motivation: We want to solve *local equations* and *local behaviors* over arbitrary algebras.

Project Overview & Motivations

Motivation: We want to solve *local equations* and *local behaviors* over arbitrary algebras.

- local equations (constraints) $\sim$ classes.

Project Overview & Motivations

Motivation: We want to solve *local equations* and *local behaviors* over arbitrary algebras.

- local equations (constraints) $\sim$ classes.
- local solutions $\sim$ objects.

Project Overview & Motivations

Motivation: We want to solve *local equations* and *local behaviors* over arbitrary algebras.

- local equations (constraints) $\sim$ classes.
- local solutions $\sim$ objects.
- constraint resolution $\sim$ semantics for object calculus.

Object Calculus in a Nutshell

- Values (“objects” / “instances”) of a type are classified into “classes”.

Object Calculus in a Nutshell

- Values (“objects”/“instances”) of a type are classified into “classes”.
- Functions (“methods”) have different behaviors depending on the “class” of its arguments.

Object Calculus in a Nutshell

- Values (“objects” / “instances”) of a type are classified into “classes”.
- Functions (“methods”) have different behaviors depending on the “class” of its arguments.
- Each “class” enforces different internal structures and different behaviors when passed to methods.

Object Calculus in a Nutshell

- Values (“objects” / “instances”) of a type are classified into “classes”.
- Functions (“methods”) have different behaviors depending on the “class” of its arguments.
- Each “class” enforces different internal structures and different behaviors when passed to methods.
- We represent each class by a record of field and function types.

Object Calculus in a Nutshell

- Values (“objects” / “instances”) of a type are classified into “classes”.
- Functions (“methods”) have different behaviors depending on the “class” of its arguments.
- Each “class” enforces different internal structures and different behaviors when passed to methods.
- We represent each class by a record of field and function types.
- We represent the behavior of each method with a “dispatch matrix”

$$\prod_{c \in \mathcal{C}} \prod_{d \in \mathcal{D}} (\tau^c \to \rho_d)$$

Object Calculus: examples

Adding Constraints in Classes

Now, we allow the declaration of *local equations* in objects.

Adding Constraints in Classes

Now, we allow the declaration of *local equations* in objects.

$$\mathit{Class}\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

Adding Constraints in Classes

Now, we allow the declaration of *local equations* in objects.

$$\mathit{Class}\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

An instance solves constraints declared by its class.

Adding Constraints in Classes

Now, we allow the declaration of *local equations* in objects.

$$\mathit{Class}\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

An instance solves constraints declared by its class.

$$\mathit{Obj}\ \{s \mapsto a, t \mapsto b\} : \mathit{Class}\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

Adding Constraints in Classes

Now, we allow the declaration of *local equations* in objects.

$$\mathit{Class}\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

An instance solves constraints declared by its class.

$$\mathit{Obj}\ \{s \mapsto a, t \mapsto b\} : \mathit{Class}\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

and also first-order constraints!

Adding Constraints in Classes

Now, we allow the declaration of *local equations* in objects.

$$Class\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

An instance solves constraints declared by its class.

$$Obj\ \{s \mapsto a, t \mapsto b\} : Class\ \{s \hookrightarrow A, t \hookrightarrow B, s = t\}$$

and also first-order constraints!

$$Class\ \{s = t, \neg\phi, \phi_1 \wedge \phi_2, \phi_1 \vee \phi_2, \exists x.\phi\}$$

Object Calculus with Constraints: examples

Problem Statement

The Calculus

Semantics of Primitive Constraints

Semantics of the Calculus