

Hommage pour le départ en retraite de Pierre Cointe, 14/09/2023

**Quelque réflexion¹ : une décennie de
collaboration avec Pierre Cointe *et al.***

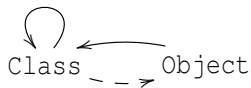
Jacques.Malenfant (at) lip6.fr

Sorbonne Université, CNRS, UMR 7606, LIP6

¹ Non, ce n'est pas une faute d'orthographe, c'est volontaire ;-)

Of structural reflection : ObjVLisp *fixpoints*

- Self-describing `Class`, **fixpoint** of the *instance-of* relation...
- But also constructed from the protocol :
 $instantiate \Rightarrow allocate \circ initialize$
- How to get this by bootstrapping ?

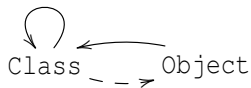


Of structural reflection : ObjVLisp *fixpoints*

- Self-describing `Class`, **fixpoint** of the *instance-of* relation...

- But also constructed from the protocol :

$instantiate \Rightarrow allocate \circ initialize$



- How to get this by bootstrapping ?

- In fact, `Class` is also a **fixpoint** of the function :

$$f(\text{Class}) = \text{Class}$$

where $f : \mathbf{C} \rightarrow \mathbf{O} := instantiate(\text{inits})$

$$instantiate : \mathbf{O}^* \rightarrow \mathbf{C} \rightarrow \mathbf{O}$$

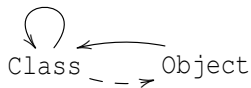
$$\text{Object}, \text{Class} \in \mathbf{C} \wedge \mathbf{C} \subseteq \mathbf{O}$$

Of structural reflection : ObjVLisp *fixpoints*

- Self-describing `Class`, **fixpoint** of the *instance-of* relation...

- But also constructed from the protocol :

instantiate \Rightarrow *allocate* \circ *initialize*



- How to get this by bootstrapping ?

- In fact, `Class` is also a **fixpoint** of the function :

$$f(\text{Class}) = \text{Class}$$

where $f : \mathbf{C} \rightarrow \mathbf{O} := \text{instantiate}(\text{inits})$

$$\text{instantiate} : \mathbf{O}^* \rightarrow \mathbf{C} \rightarrow \mathbf{O}$$

$$\text{Object}, \text{Class} \in \mathbf{C} \wedge \mathbf{C} \subseteq \mathbf{O}$$

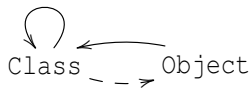
- Interest ?

Of structural reflection : ObjVLisp *fixpoints*

- Self-describing `Class`, **fixpoint** of the *instance-of* relation...

- But also constructed from the protocol :

$instantiate \Rightarrow allocate \circ initialize$



- How to get this by bootstrapping ?

- In fact, `Class` is also a **fixpoint** of the function :

$f(\text{Class}) = \text{Class}$

where $f : \mathbf{C} \rightarrow \mathbf{O} := instantiate(\text{inits})$

$instantiate : \mathbf{O}^* \rightarrow \mathbf{C} \rightarrow \mathbf{O}$

$\text{Object}, \text{Class} \in \mathbf{C} \wedge \mathbf{C} \subseteq \mathbf{O}$

- Interest ? \Rightarrow essentially, *fun* !

- possible denotational (fixpoint) semantics of ObjVLisp objects ?
- \Rightarrow bridge towards P.L. theoreticians...

On behavioral reflection : looking for a *fixpoint*, again !

- Can an object reflectively describe and redefine the way its methods are executed ?

On behavioral reflection : looking for a *fixpoint*, again !

- Can an object reflectively describe and redefine the way its methods are executed ?
- Protocol : *method execution* \Rightarrow *lookup* \circ *apply*
- Reify them as methods \Rightarrow recur the protocol !

On behavioral reflection : looking for a *fixpoint*, again !

- Can an object reflectively describe and redefine the way its methods are executed ?
- Protocol : *method execution* \Rightarrow *lookup* \circ *apply*
- Reify them as methods \Rightarrow recur the protocol !
- Two main lines of work :
 - 1 Where to put those reified methods (class, metaclass, meta-object) ?
 - 2 How to formalise this protocol ?
- Approach : define as a rewriting system
 - *lookup* \circ *apply* \Rightarrow rewriting rules “complementing” α -conversion and β -reduction : [Reflection'92]
 - redefined with a proof of termination and confluence : [LASC'96]
 - *i.e.*, proof of existence of a unique ***fixpoint*** of the rewriting process for each starting reducible expression.

Quelques remerciements plus personnels

***Merci Pierre pour ton encadrement,
ton soutien et une collaboration fructueuse !***