Signatures and models for syntax and operational semantics in the presence of variable binding

Ambroise LAFONT[1]

[1]DAPI
IMT Atlantique

PhD, 2019

Outline

Outline

Ingredients

- Programming languages (PLs) as graphs
  - (**Syntax**) vertices = terms
  - (**Semantics**) arrows = reductions between terms
- Parallel substitution: variables $\mapsto$ terms
  - monads and modules over them
- (untyped PLs)

Example

λ-calculus with β-reduction

$$S, T \quad ::= x | S\ T | \lambda x.S$$

$$(\lambda x.t)\, u \xrightarrow{\beta} t[x \mapsto u] \qquad + \quad \text{congruences}$$

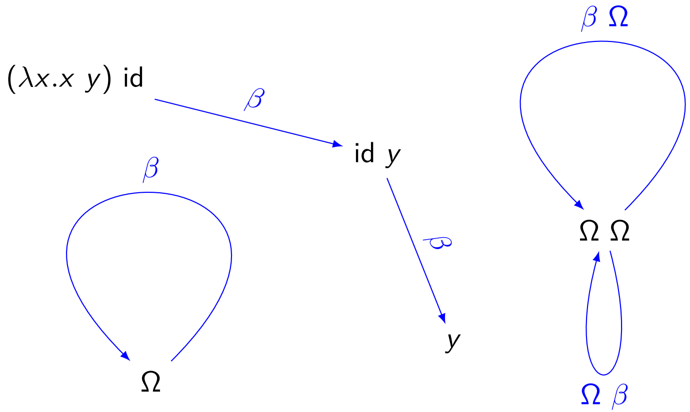
modulo α-equivalence, e.g.

$$\lambda x.x = \lambda y.y$$

Outline

PLs as graphs

Example: λ-calculus with β-reduction



- (**Syntax**) vertices = terms
- (**Semantics**) arrows = reductions (dedicated syntax: Cf labels)

Graphs

Definition

Graph = a quadruple (A, V, σ, τ) where

$$A \underset{\tau}{\overset{\sigma}{\rightrightarrows}} V$$

$A = \{\text{total set of arrows}\}$ $\qquad$ $V = \{\text{vertices}\}$

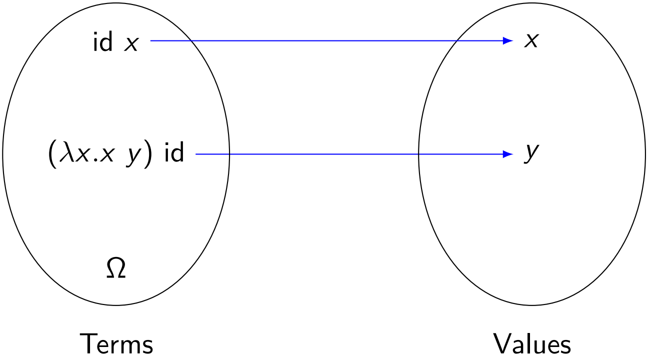
$$\begin{array}{rccl} \sigma : & A & \to & V \\ & t \xrightarrow{r} u & \mapsto & t \end{array} \qquad \begin{array}{rccl} \tau : & A & \to & V \\ & t \xrightarrow{r} u & \mapsto & u \end{array}$$

$$\sigma(r) \xrightarrow{r} \tau(r)$$

PLs as bipartite graphs

Example: λ-calculus cbv with big-step operational semantics

- term $\rightarrow$ value
- variables $=$ placeholders for values



Bipartite graphs

Definition

Bipartite graph = a quadruple (A, V_1, V_2, ∂) where

$$V_1 \xleftarrow{\sigma} A \xrightarrow{\tau} V_2$$

$$A = \{\text{arrows}\} \qquad \begin{aligned} V_1 &= \{\text{vertices in first group}\} \\ V_2 &= \{\text{vertices in second group}\} \end{aligned}$$

For simplicity, we focus on the particular case of **graphs**: $V_1 = V_2$.

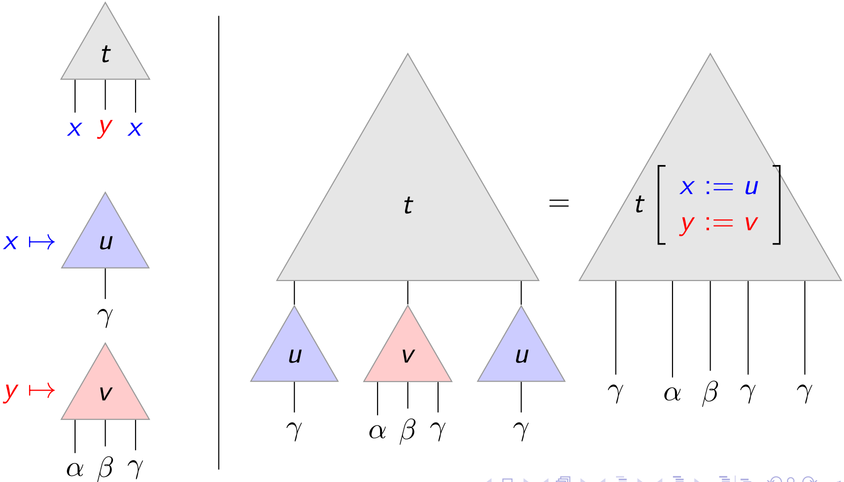
Outline

1. Reduction monads
   - Graphs
   - Substitution
2. General signatures
3. Syntax
4. Semantics

Parallel substitution

Syntax comes with substitution

terms (e.g. λ-terms) = trees with free variables as (distinguished) leaves.



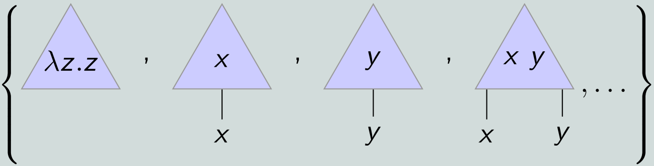
Parallel substitution made formal

Free variables indexing

$$X \mapsto \{\text{terms taking free variables in } X\}$$

Example: λ-calculus

$L(\{x, y\}) =$



Parallel substitution

For any $f : X \to L(Y)$,

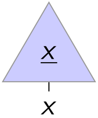
$$\begin{aligned} \mathrm{bind}_f : \quad L(X) &\to L(Y) \\ t &\mapsto t[x \mapsto f(x)] \qquad (\text{or } \ t[f]) \end{aligned}$$

Monads

λ-calculus as a monad (L, bind, η)

1. Parallel substitution (L, bind)
2. Variables are terms

$$\begin{array}{cccc} \eta_X : & X & \rightarrow & L(X) \\ & x & \mapsto & \end{array}$$



3. Monadics laws:

$$\underline{x}[f] = f(x) \qquad\qquad t[x \mapsto \underline{x}] = t$$

+ associativity:

$$t[f][g] = t\left[x \mapsto f(x)[g]\right]$$

Substitution for semantics

Our notion of PL:

- (**Syntax**) a monad (L, bind, η)
- (**Semantics**) graphs $A(X) \underset{\tau}{\overset{\sigma}{\rightrightarrows}} L(X)$ for each X

$$A(X) = \begin{array}{c}\text{total set of reductions between} \\ \text{terms taking free variables in } X\end{array}$$

- reductions: substitution of variables with L-**terms**.

$$\frac{t \xrightarrow{r} u}{t[f] \xrightarrow{r[f]} u[f]}$$

Substitution for semantics made formal

A as a **module** over L

For any $f : X \to \boldsymbol{L}(Y)$,

$$\begin{array}{rrl} \mathrm{bind}_f : & A(X) & \to A(Y) \\ & r & \mapsto r[x \mapsto f(x)] \quad (\text{or } r[f]) \end{array}$$

s.t.

$$r[x \mapsto \underline{x}] = r \qquad\qquad r[f][g] = r\,[x \mapsto f(x)[g]]$$

σ and τ as L-**module morphisms**

Source and target of $r[f]$ unspecified by the module structure on A.

$$\frac{\sigma(r) \xrightarrow{r} \tau(r)}{\sigma(r)[f] \xrightarrow{r[f]} \tau(r)[f]} \quad \text{enforces} \quad \begin{array}{l} \sigma(r[f]) = \sigma(r)[f] \\ \tau(r[f]) = \sigma(r)[f] \end{array}$$

Commutation with substitution $\Leftrightarrow$ Module morphisms $\sigma, \tau : A \to L$.

Reduction monads

Definition

Reduction monad: a quadruple (L, A, σ, τ) s.t.

- $L =$ a monad
- $A =$ a module over L
- $\sigma, \tau : A \to L$ are L-module morphisms.

Example

λ-calculus with β-reduction.

How can we specify a reduction monad?

1. signature for the (syntactic) operations for the monad;
2. reduction rules, involving some specified syntactic operations.

Use of a general notion of **signature** managing this dependancy.

Outline

Specify reduction monads

Overview

- A signature is a sequence of arities $A_1, \ldots, A_n$

Outline

Operations as module morphisms

Application commutes with substitution

$$(t\ u)[x \mapsto v_x] = t[x \mapsto v_x]\ u[x \mapsto v_x]$$

Categorical formulation

$LC \times LC$ supports LC-substitution	$\rightsquigarrow$	$LC \times LC$ is a **module over** LC
application commutes with substitution	$\rightsquigarrow$	$\mathrm{app} : LC \times LC \to LC$ is a **module morphism**

[Hirschowitz-Maggesi 2007 : Modules over Monads and Linearity]

Examples of modules

module over a monad R: supports the R-monadic substitution

- R itself

- $M \times N$ for any modules M and N

  e.g. $\mathrm{R} \times \mathrm{R}$: $f : X \to R(Y)$

  $(\mathrm{t},\mathrm{u})[\mathrm{x} \mapsto \mathrm{f}(\mathrm{x})] := (\mathrm{t}[\mathrm{x} \mapsto \mathrm{f}(\mathrm{x})], \mathrm{u}[\mathrm{x} \mapsto \mathrm{f}(\mathrm{x})])$

- M' = **derivative of a module** M: $M'(X) = M(\underbrace{X \coprod}_{\text{disjoint union}} \underbrace{\{\diamond\}}_{\text{fresh variable}})$.

  used to model an operation binding a variable (Cf next slide).

Operations as module morphisms

operations = **module morphisms** = maps commuting with substitution.

$$\mathrm{app} : \mathrm{LC} \times \mathrm{LC} \rightarrow \mathrm{LC}$$

$$\mathrm{abs} : \mathrm{LC'} \rightarrow \mathrm{LC}$$

$$\begin{aligned} \mathrm{abs}_X : \mathrm{LC}(X \coprod \{\diamond\}) &\rightarrow \mathrm{LC}(X) \\ t &\mapsto \lambda\diamond.t \end{aligned}$$

Combining operations into a single one using disjoint union

$$[\mathrm{app}, \mathrm{abs}] : (\mathrm{LC} \times \mathrm{LC}) \coprod \mathrm{LC'} \rightarrow \mathrm{LC}$$

Outline

Summary

- The first main message of your talk in one or two lines.
- The second main message of your talk in one or two lines.
- Perhaps a third message, but not more than that.

- Outlook
  - What we have not done yet.
  - Even more stuff.

For Further Reading I

A. Author.
Handbook of Everything.
Some Press, 1990.

S. Someone.
On this and that.
Journal on This and That. 2(1):50–100, 2000.