Cartesian closed 2-categories and rewriting

Aurore Alcolei

A brief presentation of Tom Hirschowitz's paper,
Cartesian closed 2-categories and permutation equivalence in higher-order rewriting

June 7, 2014

Context

Construction

Context

A lot of calculi :

- λ-calcul in cbv, cbn, lazy, optimal,
- λ-calcul with let rec / refs / call/cc,
- π-calcul, etc.

$\longrightarrow$ same kind of proofs again and again

$\longrightarrow$ a common point: the abstractions (binding)

Aim:

Having a framework to specify the semantic of (any) programming language with binding.

$\Longrightarrow$ providing tools to specify/automate proofs and construction for these languages

Previous Works

What's already there:

- Higher-Order Rewrite Systems (HRSs) from T.Nipkow
  → no notion of model, does not express reduction steps (binary relation)
- Categorical approach using Cartesian Closed Categories (CCC) by J.Lambek
  → no notion of reductions, model for *equational* theories.

Ideas:

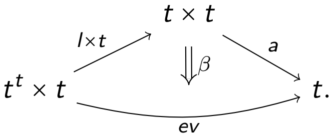
- Making signatures for HRSs into a category (**Sig**)
- Adding a dimension to Lambek's approach $\rightsquigarrow$ using **2**-Cartesian Closed Categories.

In a nutshell

Programming language/rewriting systems with binding as a 2-category where

- objects are types
- morphisms (1-cells) are terms
- morphisms between parallel morphisms (2-cells) are reductions

What would that mean?



Context

Construction

An example

Pure λ-calculus:

- grammar: $M, N \in \Lambda(\Gamma) := x \in \Gamma \mid \lambda x.M \mid MN$ (Γ set of variables)
- reduction rules:

$$(\beta) : (\lambda x.M)N \to M[N/x]$$

$$(\xi) : M \to M' \implies \lambda x.M \to \lambda x.M'$$

$$(R) : N \to N' \implies MN' \to MN'$$

$$(L) : M \to M' \implies MN \to M'N$$

Signature

Example: 2-signature for pure λ-calculus

$$\Sigma_\Lambda = \left(\{t\}, \left\{ \begin{array}{llcl} l: & [t^t] & \longrightarrow & t \\ a: & [t, t] & \longrightarrow & t \end{array} \right\}, \left\{ \beta : a(\!| l(\!| x |\!), y |\!) \to x(y) \right\} \right)$$

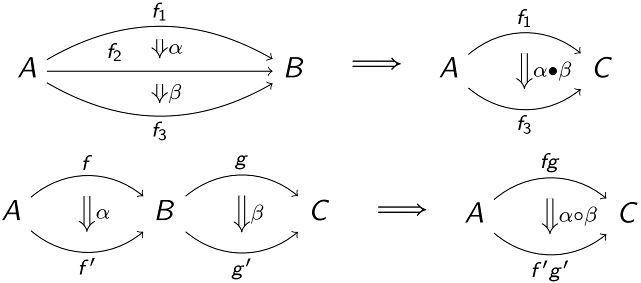
Three sets :

1. Basic types (sorts): $X_0 = \{t\}$.
2. Operations, l and a, with their type.
3. Rules β. redex and reduction are of the same type.

$$\mathbf{Sig} \underset{\mathcal{W}}{\overset{\mathcal{H}}{\underset{\perp}{\rightleftarrows}}} \mathbf{2\text{-}CCCat}$$

Cartesian Closed 2-Category

∘ 2-category := category enriched over **Cat** $\rightsquigarrow$ 2-cells, identities, vertical and horizontal compositions...



∘ Cartesian closed 2-category := 2-category with finite product and exponential, both **preserving** the 2-categorical structure.

Step 1: 1-Signature

Types of a signature:

obtained by applying (the monad)

$$\begin{array}{rccl} \mathcal{L}_0\colon & \mathbf{Sets} & \longrightarrow & \mathbf{Sets} \\ & X & \longmapsto & \{A, B := x \in X_0 \mid A \times B \mid \mathbf{1} \mid B^A\} \end{array}$$

on X_0

→ 1-signature:

- sequent := element of $\mathcal{S}_0(X) = \mathcal{L}_0(X)^* \times \mathcal{L}_0(X)$
- 1-signature := (X_0, X_1) with $\varphi_1\colon \begin{array}{ccc} X_1 & \longrightarrow & \mathcal{S}_0(X) \\ c & \longmapsto & (dom(c), cod(c)) \end{array}$

Step 2: 2-Signature (1)

Terms of a signature:

generated by

+ simply typed λ-calculus
+ pairing and projections
+ $$\frac{\ldots \qquad \Gamma \vdash M_i \colon \Delta_i \qquad \ldots}{\Gamma \vdash c(\!| M_1, \ldots, M_n |\!) \colon A} \; c \in X_1(\Delta, A)$$

modulo β η reduction $(^*)$

Examples:

- $[\![\lambda x.M]\!] = l(\!|\lambda x.[\![M]\!]|\!)$
- $[\![MN]\!] = a(\!|[\![M]\!], [\![N]\!]|\!)$
- $[\![x]\!] = x$

$(^*)$ to get a structure close to the CCC

Step 2: 2-Signature (2)

→ monad $\mathcal{L}_1$: $\textbf{Sig}_1 \longrightarrow \textbf{Sig}_1$ such that

$$\begin{cases} \mathcal{L}_1(X)_0 = X_0 \text{ and} \\ \mathcal{L}_1(X)_1 = \text{ terms of the signature} \end{cases}$$

→ $\mathcal{L}_1(X)_{||}$:= set of pairs of terms with same type

→ 2-signature :

(X, X_2) where X is a 1-signature and X_2 is equipped with

$$\begin{array}{rccl} \varphi_2: & X_2 & \longrightarrow & \mathcal{L}_1(X)_{||} \\ & r & \longmapsto & \text{(a term, its reduction by r)} \end{array}$$

Ex : $\beta \mapsto (a(\!|l(\!|x|\!), y|\!), x[y]) \in \mathcal{L}_1(Sigma)_{||}$

Step 3: The adjunction

A similar approach:

- Defining a monad $\mathcal{L}$ over **Sig**
- Reductions are generated by a 2λ-calculus:
  - Context rules :

$$\frac{}{\Gamma, x : A, \Delta \vdash x : x \to x : A} \qquad \frac{\Gamma, x{:}\,A \vdash P : M \to N{:}\,B}{\Gamma \vdash (\lambda x{:}\,A.P) : \lambda x{:}\,A.M \to \lambda x{:}\,A.N{:}\,B^A}$$

$$\frac{\Gamma \vdash P_1 : M_1 \to N_1{:}\,G_1 \qquad \dots \qquad \Gamma \vdash P_n : M_n \to N_n{:}\,G_n}{\Gamma \vdash c(\!|P_1, \dots, P_n|\!) : c(\!|M_1, \dots, M_n|\!) \to c(\!|N_1, \dots, N_n|\!){:}\,A}\ (c \in X_1(G \vdash A))$$

...

  - Special rule :

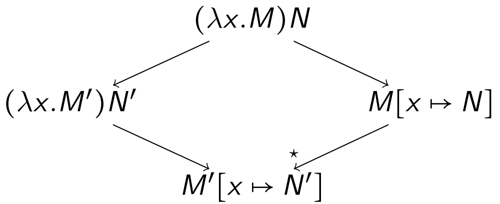
$$\frac{\dots \qquad \Gamma \vdash P_i{:}\,M_i \to N_i{:}\,G_i \qquad \dots}{\Gamma \vdash r\langle\!\langle P_1, \dots, P_n \rangle\!\rangle{:}\,M[M_1, \dots, M_n] \to N[N_1, \dots, N_n]{:}\,A}\ (r \in X(G \vdash M, N{:}A))$$

- Modulo some equations ... to get a structure close to the 2-CCC

Example of equation

- Equivalence rules := β and η equivalences, equivalence by permutation, …

- Example:



$$
\begin{aligned}
\text{Left reduction:}\quad & a(\!|I(\!|\lambda x^t.P|\!), Q|\!); \beta\langle\!\langle \lambda x^t.M', N'\rangle\!\rangle \\
& \equiv \beta\langle\!\langle \lambda x^t.P, Q\rangle\!\rangle \\
& \equiv \beta\langle\!\langle \lambda x^t.M, N\rangle\!\rangle; (\lambda x^t.P)Q \\
& \equiv \beta\langle\!\langle \lambda x^t.M, N\rangle\!\rangle; P[x \mapsto Q] \qquad : \text{Right reduction.}
\end{aligned}
$$

What we get

An adjunction:

$$\mathbf{Sig} \underset{\mathcal{W}}{\overset{\mathcal{H}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{2\text{-}CCCat}$$

$\implies$ models for Σ : morphisms $\mathcal{H}(\Sigma) \to \mathbf{C}$ in **2-CCCat**

What does that mean?

$$\begin{array}{ccccc} & & t \times t & & \\ & \overset{I \times t}{\nearrow} & \Downarrow \beta & \overset{a}{\searrow} & \\ t^t \times t & & \xrightarrow[ev]{\hspace{6em}} & & t. \end{array}$$