Algebraic λ-calculus

On linear combinations of terms

Lionel Vaux

Institut de Mathématiques de Luminy, Marseille
vaux@iml.univ-mrs.fr

Rewriting Techniques and Applications, 2007, Paris

Background

There are models of linear logic where

- types are interpreted by vector spaces (more generally *modules over rigs*) ;
- proofs are interpreted by linear maps ;
- linear maps from $!A$ to B are analytic maps from A to B.

Thomas Ehrhard.
On Köthe sequence spaces and linear logic.
MSCS, 12 :579–623, 2001.

Thomas Ehrhard.
Finiteness spaces.
MSCS, 15(4) :615–646, 2005.

Differential calculus

- Usual translation of simple types : $A \Rightarrow B = !A \multimap B$.
- Ehrhard-Regnier's differential λ-calculus : terms as analytic functions between vector spaces.
  - Differentiation.
  - But also sums and linear combinations.
- Exists in pure and typed flavours.

Thomas Ehrhard and Laurent Regnier.
The differential lambda-calculus.
TCS, 309 :1–41, 2003.

Differential calculus

- Usual translation of simple types : $A \Rightarrow B = !A \multimap B$.
- Ehrhard-Regnier's differential λ-calculus : terms as analytic functions between vector spaces.
  - Differentiation.
  - But also sums and linear combinations.
- Exists in pure and typed flavours.
- *Reduction may behave strangely...*

Thomas Ehrhard and Laurent Regnier.
The differential lambda-calculus.
TCS, 309 :1–41, 2003.

Differential calculus

- Usual translation of simple types : $A \Rightarrow B = !A \multimap B$.
- Ehrhard-Regnier's differential λ-calculus : terms as analytic functions between vector spaces.
  - Differentiation.
  - But also sums and linear combinations.
- Exists in pure and typed flavours.
- *Reduction may behave strangely... only because of coefficients in linear combinations.*

Thomas Ehrhard and Laurent Regnier.
The differential lambda-calculus.
TCS, 309 :1–41, 2003.

Terms as functions between vector spaces

Basic ideas

- Extend the set of terms so that if forms a vector space.
- Mappings with values in a vector space form a vector space :

$$(a.f + b.g)[x] \quad = \quad a.f[x] + b.g[x]$$

In λ-ish words :

$$\begin{aligned} (a.s + b.t)\,u \quad &= \quad a.(s)\,u + b.(t)\,u \\ \lambda x\,(a.s + b.t) \quad &= \quad a.\lambda x\,s + b.\lambda x\,t \end{aligned}$$

Definitions

Definition (Rig)

We call *rig* any tuple $(\mathsf{R}, +, 0, \times, 1)$ where

- $(\mathsf{R}, +, 0)$ and $(\mathsf{R}, \times, 1)$ are commutative monoids ;
- $0 \times a = 0$ and $(a + b) \times c = (a \times c) + (b \times c)$.

Definition (Module)

A module over a rig R (or R-module) is a tuple $(V, +, \vec{0}, .)$ where

- $(V, +, \vec{0})$ is a commutative monoid ;
- external product (.) is left and right additive : $0.v = a.\vec{0} = \vec{0}$, $(a + b).v = a.v + b.v$ and $a.(v + w) = a.v + a.w$
- $1.v = v$ and $a.b.v = (a \times b).v$.

Morphology

Definition (Raw terms)

Let R be a fixed rig.

The set Λ_{R} of raw terms $(\sigma, \tau, \dots)$ is given by :

$$\sigma, \tau ::= x \mid \lambda x\, \sigma \mid (\sigma)\, \tau \mid \vec{0} \mid a.\sigma \mid \sigma + \tau \quad .$$

We will often write $\sum_{i=1}^{n} a_i.\sigma_i$ for

$$a_1.\sigma_1 + \cdots + a_n.\sigma_n + \vec{0} \quad .$$

Definition (Structural equality)

We write $\sim$ for the transitive closure of α-equivalence and AC of $+$.

Canonical forms

Definition

We define sets $\mathsf{A_R}$ of atomic terms $(s, t, \dots)$ and $\mathsf{C_R}$ of of canonical terms $(S, T, \dots)$ by :

- any variable x is an atomic term ;
- if $x \in \mathcal{V}$ and $s \in \mathsf{A_R}$ then $\lambda x\, s \in \mathsf{A_R}$;
- if $s \in \mathsf{A_R}$ and $T \in \mathsf{C_R}$ then $(s)\, T \in \mathsf{A_R}$;
- if $a_1, \dots, a_n \in \mathsf{R}^\bullet$ and $s_1, \dots, s_n \in \mathsf{A_R}$ are pairwise distinct ($\not\sim$) then $\sum_{i=1}^{n} a_i.s_i \in \mathsf{C_R}$.

If $s \in \mathsf{A_R}$ we write $\vec{s} = 1.s + \vec{0} \in \mathsf{C_R}$.

Canonization

Define $\mathsf{can} : \Lambda_{\mathsf{R}} \longrightarrow \mathsf{C}_{\mathsf{R}}$ inductively

- $\mathsf{can}(x) = \vec{x}$;
- if $\mathsf{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ then $\mathsf{can}(\lambda x\, \sigma) = \sum_{i=1}^{n} a_i.\lambda x\, s_i$;
- if $\mathsf{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ and $\mathsf{can}(\tau) = T$ then $\mathsf{can}((\sigma)\, \tau) = \sum_{i=1}^{n} a_i.(s_i)\, T$;
- $\mathsf{can}(\vec{0}) = \vec{0}$;
- if $\mathsf{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ and $\mathsf{can}(\tau) = \sum_{i=n+1}^{n+p} a_i.s_i$ then
$$\mathsf{can}(\sigma + \tau) = \mathsf{cansum}\left(\sum_{i=1}^{n+p} a_i.s_i\right) \quad ;$$
- if $\mathsf{can}(\sigma) = \sum_{i=1}^{n} a_i.s_i$ then $\mathsf{can}(a.\sigma) = \mathsf{cansum}\left(\sum_{i=1}^{n}(a \times a_i).s_i\right)$.

Implementation of algebraic identities

Representation

We write $\sigma =_{\mathsf{R}} \tau$ iff $\mathtt{can}(\sigma) \sim \mathtt{can}(\tau)$. Then :

- $=_{\mathsf{R}}$ is an equivalence relation ;
- $(\Lambda_{\mathsf{R}}/=_{\mathsf{R}}, +, \vec{0}, .)$ is an R-module ;
- each $=_{\mathsf{R}}$-class has a unique canonical representative (up to $\sim$).

Algebraic terms

We now consider terms up to $=_{\mathsf{R}}$:

- we write Δ_{R} for the set of all $=_{\mathsf{R}}$-classes of atomic terms (which we call simple terms) ;
- $\Lambda_{\mathsf{R}}/=_{\mathsf{R}}$ is generated by Δ_{R} ;
- we call algebraic terms the elements of $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle = (\Lambda_{\mathsf{R}}/=_{\mathsf{R}})$.

Extending β-reduction

Idea

Define a reduction relation on $\mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$ such that :

- if σ is simple then $(\lambda x\,\sigma)\,\tau \to \sigma\,[x := \tau]$;
- if σ is simple, $\sigma \to \sigma'$ and $a \neq 0$ then $a.\sigma + \tau \to a.\sigma' + \tau$.

Warning

It cannot be defined by induction on terms : if $a, b \in \mathsf{R}^{\bullet}$ with $a + b = 0$ then $\vec{0} =_{\mathsf{R}} a.\sigma + b.\sigma$ may reduce.

Extending β-reduction

Definition

We define $\rightarrow$ on algebraic terms by the following statements :

Reduction of simple terms :

- $(\lambda x\ s)\ T \rightarrow s\,[x := T]$;
- if $s \rightarrow S'$ then

$$\begin{array}{rcl} \lambda x\ s & \rightarrow & \lambda x\ S' \\ (s)\ T & \rightarrow & (S')\ T \\ a.s + T & \rightarrow & a.S' + T \text{ as soon as } a \neq 0 \end{array}$$

Extension to all terms : if $T \rightarrow T'$ then $(s)\ T \rightarrow (s)\ T'$.

Examples

- Every ordinary β-reduction is a valid reduction of algebraic λ-calculus : if $s \to_\Lambda t$ then $s \to t$.
- If $\mathsf{R} = \mathbf{Z}$, and $s \to S'$:

$$\vec{0} =_{\mathsf{R}} s - s \to S' - s \quad .$$

- More generally, if $\exists a \in \mathsf{R}$ s.t. $a + 1 = 0$:

$$S' =_{\mathsf{R}} (s + as) + S' \to s + (aS' + S') =_{\mathsf{R}} s \quad .$$

- If $\mathsf{R} = \mathbf{Q}$ and $s \to S'$:

$$s =_{\mathsf{R}} \frac{1}{2}s + \frac{1}{2}s \to \frac{1}{2}s + \frac{1}{2}S' \to \frac{1}{4}s + \frac{3}{4}S' \to \cdots$$

Confluence

Tait–Martin-Löf

Introduce parallel reduction $\rightrightarrows$ such that

$$\to \subset \rightrightarrows \subset \to^* .$$

Denote by $S\downarrow$ the term obtained by firing all redexes in S.

Lemma

For all terms S and S' such that $S \rightrightarrows S'$, we have $S' \rightrightarrows S\downarrow$.

This holds only thanks to the way we reduce linear combinations.

Theorem

Reduction $\to$ enjoys Church-Rosser.

Positive rig

Definition

A rig R is said to be positive if $a + b = 0$ implies $a = b = 0$.

Examples :

- The set $\mathbf{N}$ of natural integers.
- Sets $\mathbf{Q}^+$ and $\mathbf{R}^+$ of non negative numbers.
- The set $\mathsf{R}[\xi_0, \xi_1, \dots]$ of polynomials over indeterminates $\xi_0, \xi_1, \dots$, with coefficients taken in a positive rig R.

Conservativity

Assume R is positive.

Lemma

If $s \in \Lambda$ *and* $s \to^* S'$ *then there is* $t \in \Lambda$ *such that* $S' \to^* t$ *and* $s \to^*_\Lambda t$.

Theorem

If $s, s' \in \Lambda$, *then* $s \leftrightarrow s'$ *iff* $s \leftrightarrow_\Lambda s'$.

Corollary

Reductional equality is sound.

Indeterminate forms

Something is rotten in the state of Denmark.

- Let Y be a fixed point of ordinary λ-calculus.
  Write $\infty_\sigma = (Y)(\lambda x\,(x+\sigma))$:
  $$\infty_\sigma \to \sigma + \infty_\sigma \quad .$$
- Assume $-1 \in \mathsf{R}$:
  $$0 =_{\mathsf{R}} \infty_\sigma - \infty_\sigma \quad !$$

Indeterminate forms

Something is rotten in the state of Denmark.

- Let Y be a fixed point of ordinary λ-calculus.
  Write $\infty_\sigma = (Y)(\lambda x\,(x+\sigma))$:

$$\infty_\sigma \to \sigma + \infty_\sigma \ .$$

- Assume $-1 \in \mathsf{R}$:

$$0 =_{\mathsf{R}} \infty_\sigma - \infty_\sigma \ !$$

Theorem

Assume R *is not positive and* $a, b \in \mathsf{R}^\bullet$ *are such that* $a + b = 0$.
Then, for all $\sigma \in \mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$, $\vec{0} \to^* a.\sigma$ *and* $a.\sigma \to^* \vec{0}$.

Typing

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma, x : A \vdash \sigma : B}{\Gamma \vdash \lambda x\, \sigma : A \Rightarrow B}$$

$$\frac{\Gamma \vdash \sigma : A \Rightarrow B \qquad \Gamma \vdash \tau : A}{\Gamma \vdash (\sigma)\, \tau : B}$$

$$\frac{}{\Gamma \vdash \vec{0} : A} \qquad \frac{\Gamma \vdash \sigma : A}{\Gamma \vdash a.\sigma : A} \qquad \frac{\Gamma \vdash \sigma : A \qquad \Gamma \vdash \tau : A}{\Gamma \vdash \sigma + \tau : A}$$

Necessary conditions

Positivity

If R is not positive, every term reduces.
Moreover, typability isn't compatible with $=_{\mathrm{R}}$.

Finite splitting

If $\mathrm{R} = \mathbf{Q}^+$ and $s \to S'$:

$$s =_{\mathrm{R}} \frac{1}{2}s + \frac{1}{2}s \to \frac{1}{2}s + \frac{1}{2}s' \to \frac{1}{4}s + \frac{3}{4}s' \to \cdots$$

Hence, for all $a \in \mathrm{R}$,

$$\{(a_1, \ldots, a_n) \in (\mathrm{R}^\bullet)^n ;\ n \in \mathbf{N} \text{ and } a = a_1 + \cdots + a_n\}$$

must be finite.

Sufficient conditions

Definition (Width)

Define $w : \mathsf{R} \longrightarrow \mathbf{N}$ by :

$$w\,(a) = \max\left\{n \in \mathbf{N};\ \exists(a_1, \ldots, a_n) \in (\mathsf{R}^\bullet)^n;\ a = a_1 + \cdots + a_n\right\}.$$

Theorem

If w is a morphism of rigs, then all typable terms are SN.

Example

$\mathbf{N}[\xi_1, \xi_2, \ldots]$.

Sketch of proof

Adapt your own favourite proof by reducibily, using the following lemma.

Lemma

Write N_R *for the set of simple SN terms.*
Then the set of all SN terms is $\mathsf{R}\left\langle \mathsf{N}_\mathsf{R} \right\rangle$.

Proof: Let $S \in \mathsf{R}\left\langle \mathsf{N}_\mathsf{R} \right\rangle$.
One proves that S is SN by induction on

$$\sum_{s \in \mathsf{A}_\mathsf{R}} w\left(S_s\right)|s| \quad .$$

□

Weak normalization scheme

Assume R is positive and $\sigma \in \mathsf{R}\langle\Delta_{\mathsf{R}}\rangle$ is typable.

Algorithm

- Replace scalars occurring in $\mathtt{can}(\sigma)$ with formal pairwise distinct indeterminates $(\xi_1, \xi_2, \dots)$.
- The object τ thus obtained lies in $\mathsf{R}'\langle\Delta_{\mathsf{R}'}\rangle$, where $\mathsf{R}' = \mathbf{N}[\xi_1, \xi_2, \dots]$.
- τ is also typable and SN applies in $\mathsf{R}'\langle\Delta_{\mathsf{R}'}\rangle$.
- Replace indeterminates by their values in the NF of τ : this is the normal form of σ.

Failure is expected !

Assume :

- $(+, \vec{0}, .)$ are part of the syntax and make the set of terms a R-module ;
- reduction is contextual ;
- ordinary β-reductions are valid reductions.

Then, as soon as $-1 \in \mathrm{R}$:

$$(\lambda x\, x)\, \infty_y + (-1).\infty_y \to^* \begin{cases} \vec{0} \\ y \end{cases}$$

Hence the calculus is either unsound or non confluent.

Failure is expected !

Assume :

- $(+, \vec{0}, .)$ are part of the syntax and make the set of terms a R-module ;
- reduction is contextual ;
- ordinary β-reductions are valid reductions.

Then, as soon as $-1 \in \mathrm{R}$:

$$(\lambda x\, x)\, \infty_y + (-1).\infty_y \to^* \begin{cases} \vec{0} \\ y \end{cases}$$

Hence the calculus is either unsound or non confluent.

Something has to fail.

Drop confluence

Motto : Reduce only in canonical forms.

Details

- Consider only atomic and canonical terms.
- Define reduction in the natural way (canonize after each elementary reduction step).

Expected outcome

- Typable terms are SN.
- Conservative over ordinary λ-calculus.
- Can be somehow simulated in our setting as a *reduction strategy* (cf. weak normalization).
- Not confluent (whatever R).

Drop purity

Motto : Avoid fixed points by typing.

Details

Consider Church-style terms up to *typed* algebraic equality :
types are R-modules.

Expected outcome

- Should be SN, hence sound.
- Confluence ?

Drop contextuality

Motto : Perform algebraic rewriting only on values.

Details

Adapt Arrighi-Dowek's work on *linear* algebraic λ-calculus.

- Work on raw terms (equality : $\sim$).
- Introduce algebraic equality as part of reduction, allowing algebraic rewriting steps only on closed normal terms.

Pablo Arrighi and Gilles Dowek.
Linear-algebraic lambda-calculus : higher-order, encodings and confluence.
Manuscript, 2006.

Drop contextuality

Expected outcome

- Conservative over ordinary λ-calculus.
- Confluent and sound.
- The set of terms is not an R-module.
- Reduction is not contextual.
- Normalization properties ?

Drop soundness

Motto : β-reduction upto algebraic equality.

Details

This was the topic of the talk.

Outcome

- The set of terms is R-module w.r.t. syntactic sum and external product.
- Contextual and confluent reduction.
- Sometimes not conservative over ordinary λ-calculus, and even unsound.

THE END