From the finite to the transfinite: $\Lambda\mu$-terms and streams.

Fanny He and Alexis Saurin

Laboratoire PPS
University Paris Diderot
Paris, France
F.He@bath.ac.uk
alexis.saurin@pps.univ-paris-diderot.fr

April 16, 2014

Abstract

In this talk, we wish to discuss an ongoing work on the infinitary calculus arising from the $\Lambda\mu$-calculus. In contrast with usual λ-calculus which gives rise to infinite terms, the $\Lambda\mu$-calculus, as well as related calculi, gives rise to transfinite terms, that is, to infinite terms having subterms not only at finite positions, but also at infinite positions.

After some background on $\Lambda\mu$, we motivate the appearance of transfinite terms and compare (and contrast) with the work by Ketema *et al.* [7], the only work, to our knowledge, addressing rewriting theory for transfinite terms.

This talk proposal is concerned with ongoing work and, as such, it raises many more questions than it addresses. Still, we think it provides motivations for the study of transfinite calculi as well as some criteria for designing infinitary calculi.

1 Introduction

Infinite λ-terms arise naturally in the theory of (finite) λ-calculus, for instance from the consideration of Böhm trees or other Böhm-like trees [3] which can be understood as infinite normal forms for various notions of convergence. Infinitary λ-calculi [6, 1] thus extend traditional λ-calculus with infinite terms and possibly transfinite reductions [5].

The $\lambda\mu$-calculus [9], a λ-calculus originating in the study of the computational content of classical reasoning, has a somehow different connection with infinity. Indeed, μ-abstraction abstracts over evaluation contexts, or continuations, and can thus be viewed as potentially infinitary λ-abstractions. This is emphasized by reduction:

$$\boxed{\mu\alpha.y \longrightarrow^n_{\mathsf{fst}} \lambda x_1 \dots x_n.\mu\alpha.y \qquad x_1, \dots, x_n \neq y}$$

where n is arbitrary.

This viewpoint on μ-abstraction as *stream abstraction* [10, 12] leads to a natural infinitary extension of the $\Lambda\mu$-calculus where terms may have subterms at infinite positions, that is, to a *transfinite* calculus.

In their survey of infinitary λ-calculus, Barendregt and Klop [1] mention four main motivations for the studies of infinitary λ-calculi: *(i)* semantics of λ-calculus, *(ii)* pragmatics of computing with λ-terms, *(iii)* expressivity and *(iv)* theoretical coherence and transparency.

The same four motivational themes guide our work on a transfinite foundation for $\Lambda\mu$-calculi. Our talk could thus quite reasonably be organized along those four lines which motivate us to study transfinite λ-calculi in order to understand $\Lambda\mu$(-like) calculi[1]:

1. *Semantics.* Even though viewing $\Lambda\mu$-calculus as a stream calculus dates back to the separation proof [10], the true connection with transfinitary calculi really comes from the study of $\Lambda\mu$-Böhm trees (see below). Other recent semantics on $\Lambda\mu$-calculus also point towards transfinite calculi (See Nakazawa stream-models for $\Lambda\mu$ in [8] for instance);

2. *Pragmatics.* Transfinite reduction sequences already make much sense for λ-calculus. Here, we have a slight change of perspective on them: having stream-like abstractions, we have an abstraction to pass infinitely many arguments through a data-abstraction. Moreover, with $\Lambda\mu$-calculus, standard results from infinitary λ-calculi, such as the compression lemma, shall be reconsidered.

3. *Expressivity.* Input streams have no reason to be recursive (think that they can typically come from a physical device, activity on a network, ...) and can be truly infinite. Thus, they cannot be *represented* by finite $\Lambda\mu$-terms, but can be *treated* by a finite $\Lambda\mu$-term.

4. *Theoretical coherence and transparency.* $\Lambda\mu$-calculus results from an attempt to fix a problem with the $\lambda\mu$-calculus, namely its first notable negative result: the failure of Böhm theorem in Parigot's calculus as shown by David and Py [2]. $\Lambda\mu$-calculus constitutes a Böhm-complete calculus extending Parigot's calculus in a minimal way. The frontier between a calculus satisfying Böhm theorem and a calculus in which the property does not hold is often very tight and we hope that considering the calculi in a uniform framework of infinitary calculi can give us some tools to view more clearly what is going on there. The uniformity of the infinitary framework should indeed be useful to understand the expressiveness of various levels of the stream hierarchy [11], or if a separation theorem holds.

The remaining of this abstract consists in a brief review on $\Lambda\mu$ and $\Lambda\mu$-Böhm trees, followed by a recall on the approach of Ketema *et al.* approach to transfinite rewriting. In the final section we will list some questions we are willing to discuss at the workshop and contrast our approach with respect to usual approach to infinitary (and transfinitary) λ-calculi.

2 From (finite) $\Lambda\mu$ to transfinite terms via $\Lambda\mu$-Böhm trees

$\Lambda\mu$-calculus $\Lambda\mu$-calculus is a slight variant of Parigot's $\lambda\mu$-calculus introduced in the early 90's to provide a term calculus for classical proofs. It appeared at several occasions in the literature but was systematically studied and recognized as a proper calculus after it appeared that, contrarily to the original calculus [2], $\Lambda\mu$ satisfies Böhm theorem [10].

Definition 2.1. $\Lambda\mu$-terms are inductively defined by the following syntax:

$$\boxed{\Lambda\mu : t ::= x \mid \lambda x.t \mid (t)u \mid \mu\alpha.t \mid (t)\alpha}$$

(λ-variables are denoted by $x, y, h \dots$ and μ-variables by $\alpha, \gamma, \theta \dots$)

$\Lambda\mu$-calculus reductions contain usual β and η rules from λ-calculus, corresponding $\beta\eta$-rules for μ-abstraction as well as a rule connecting the two categories of objects, terms and streams (for which λ and μ-variables stand respectively), on which $\Lambda\mu$ is built:

$$\boxed{\mu\alpha.t \longrightarrow_{\mathsf{fst}} \lambda h.\mu\theta.t\{(u)h\theta/(u)\alpha\} \qquad h, \theta \notin \mathsf{FV}(t)}$$

[1]It should be noted, though, that the connections between $\Lambda\mu$-calculus and infinitary calculi are slightly different from that of ordinary λ-calculus.

This rule shall be understood as the rule for observing streams via pattern-matching, where h stands for the head of the stream and θ for the tail. As such, rule fst naturally suggests an infinitary interpretation of μ-abstraction as an infinitary λ-abstraction:

$$\mu\alpha.t \sim \lambda(x_i^\alpha)_{i\in\omega}.t\{(u)\ x_0^\alpha x_1^\alpha \dots /(u)\ \alpha\}$$

In $\Lambda\mu$, streams are only inhabited by variables, which makes it a bit trivial. It is however easy to inhabit them with constructed streams by slightly extending the calculus (see for instance [4] in this direction):

$$\begin{array}{l} \text{Terms } t ::= x \mid \lambda x.t \mid (t)u \mid \mu\alpha.t \mid (t)S \\ \text{Streams } S ::= \alpha \mid [t|S] \end{array}$$

$\Lambda\mu$-Böhm trees The above intuition is made more precise when considering Böhm trees (as well as Nakajima trees) for $\Lambda\mu$-calculus (see [12] for details):

Definition 2.2. Böhm trees for $\Lambda\mu$-calculus ($\mathfrak{B} \in \Lambda\mu\text{-}\mathfrak{BT}$) are (coinductively) defined as follows:

$$\mathfrak{B} ::= \Omega \mid \Lambda(x_i)_{i\in\mu\in\omega^2}.(y)(\mathfrak{B}_j)_{j\in\lambda\in\omega^2}$$

Nakajima trees for $\Lambda\mu$-calculus ($\mathfrak{N} \in \Lambda\mu\text{-}\mathfrak{NT}$) are (coinductively) defined as follows:

$$\mathfrak{N} ::= \Omega \mid \Lambda(x_i)_{i\in\omega^2}.(y)(\mathfrak{N}_j)_{j\in\omega^2}.$$

Example 2.3. Let $t = \mu\alpha.\lambda x.\mu\beta.\lambda y.((x)y\ ((\Delta)\Delta)\beta)\ \beta$. $\mathfrak{B} = \Lambda(z_i)_{i\in\omega\cdot 2+1}.(z_\omega)(\mathfrak{B}_j)_{j\in\omega}$ with

- $\mathfrak{B}_0 = z_{\omega\cdot 2}$,
- $\mathfrak{B}_1 = \Omega$ and
- $\mathfrak{B}_{j+1} = z_{\omega\cdot 2+j}$ for $1 \leq j < \omega$.

Böhm trees for the Stream hierarchy [11] uniformly generalize $\Lambda\mu$-Böhm trees with the following coinductive definition:

$$\mathfrak{B}^n ::= \Omega \mid \Lambda(x_i)_{i\in\mu\in\omega^{n+1}}.(y)(\mathfrak{B}_j^n)_{j\in\lambda\in\omega^{n+1}}$$

(λ corresponds to instantiating n with 0, $\Lambda\mu$ by instantiating n with 1.)

This shows that the ω^2 bound is only due to μ-abstraction, a contingent phenomenon...

Transfinite terms While usual infinitary systems allow terms having infinitely many subterms at arbitrarily large finite depth, they do not allow subterms at infinite depth. $\Lambda\mu$-Böhm trees, described above, clearly suggest that one should have subterms at infinite depth when interested in the $\Lambda\mu$-calculus.

To our knowledge, there is only one work which considers the possibility to have such transfinite terms, namely Ketema *et al.* paper on transfinite term rewriting [7]. We briefly recall the main characteristics of their approach. Essentially a *transfinite term* is a labelling function t of a set of transfinite positions, where transfinite positions are maps p from an ordinal α, its length, to the natural numbers. The labelling function shall satisfy several conditions, some being usual (t is prefix closed, the arity of a label at position p induces the number of positions immediately extending p where t is defined), and some are specific to transfinite terms, typically that if the length of p is a limit ordinal, then t is defined at p if t is defined at q, $\forall q < p$. We denote by $Pos(t)$ the domain of t. This allows to consider terms like $(f_1(f_2(f_3(\dots x))))$, which are not infinite terms but transfinite terms.

A *transfinite Term Rewriting System (tTRS)* is a pair (Σ', R), where Σ' is a set of symbols of finite arity and R is a set of transfinite rewrite rules, that is a set of pairs $l \to r$ of finite terms $\langle l, r\rangle$ such that l is not a variable and any variable occurring in r also appears in l. *Transfinite rewrite steps* can then be defined in a quite standard way rewriting a transfinite term $s = C[\sigma(l)]$ into $t = C[\sigma(r)]$ if $l \to r \in R$, $C[\Box]$ is a one-hole context and σ a substitution.

3 Transfinite terms and $\Lambda\mu$

With the previous definitions and remarks, it is not difficult to have an idea on how $\Lambda\mu$-calculus could fit in the picture. Still, there are several difficulties to be considered:

- The natural transfinite analog to $\mu\alpha.(x)\ \alpha$ is $\lambda x_0^\alpha x_1^\alpha \ldots.(x)\ x_0^\alpha x_1^\alpha \ldots$. One sees two cases of infinity embodied in each "…". The problem here is while the infinite sequence for abstraction variables goes down, as usual, the one for applied variables *goes up* and this is not well suited with the above definitions. This is however easily taken care of by the introduction of streams in the syntax and we will present a suitable notion of transfinite terms for $\Lambda\mu$.

Still, even at the level of the notion of transfinite terms, some differences with the modelling by Ketema *et al.* arise:

- Compared with usual infinite λ-calculi where infinite terms arise from the sole iteration of β-rules, we see here a possible interaction between β and fst. Since the computational content of fst is quite poor, it can be considered statically and should maybe be considered on different grounds than infinite normal forms arising from β-reductions. An additional hint that something is going on there is that the modelling of transfinite terms by Ketema *et al.* only works for our setting if we consider transfinite terms solely generated by fst: as soon as β enters the picture, one has to remove the restriction that a subterm at limit ordinal shall be defined as soon as all its (infinite) prefixes are defined. Think of $\mu\alpha.(Y)\ \lambda f\, x.f$.

Things are more open when dealing with reduction sequences:

- As noticed by Ketema *et al.*, the usual approach to the convergence of infinitary reduction sequences based on a suitable notion of metrics on terms or on the sequence of depths of fired redexes is problematic with transfinite terms. In our setting, we face an even more problematic situation since one is expecting that the transfinite counterpart of $\mu\beta.(\mu\alpha.x)\ \beta$ converges to $\lambda x_0^\beta x_1^\beta \ldots.x$ after ω steps occurring at depth... ω. No notion of strong convergence can be used here and it is easy to find variants of this example where the depth of the first difference between terms in the sequence stays at a constant depth all along the infinite reduction.

In addition to the problem with the topological modelling required to provide a good notion of limit on transfinite reduction sequences, it seems to us that Ketema *et al.* approach is not precisely what we need for modelling the $\Lambda\mu$-calculus:

- For instance, the *push down* and *pull up* properties certainly make sense in our setting, but in a slightly different meaning: an infinite reduction sequence of fst starting on $\mu\alpha.x$:

  $$\mu\alpha_0.x \longrightarrow_{\mathsf{fst}} \lambda x_0^\alpha.\mu\alpha_1.x \longrightarrow_{\mathsf{fst}} \lambda x_0^\alpha x_1^\alpha.\mu\alpha_2.x \longrightarrow_{\mathsf{fst}} \ldots \longrightarrow_{\mathsf{fst}} \lambda x_0^\alpha x_1^\alpha \ldots x_n^\alpha.\mu\alpha_{n+1}.x \longrightarrow_{\mathsf{fst}} \ldots$$

  converging to $\lambda x_0^\alpha x_1^\alpha \ldots x_n^\alpha \ldots.x$. This reduction certainly pushes down variable x: is it not at finite depth at all finite prefixes of the reduction while at depth ω at the limit?

  However, according to Ketema *et al.*'s definition of push-down/pull-up properties, we should reach after ω steps the term $\lambda x_0^\alpha x_1^\alpha \ldots x_n^\alpha \ldots.\mu\alpha.x$. In our case, μ-abstraction disappeared at the limit, or shall we say, it ran out of fuel.

4 Conclusion and perspectives

It is certainly not time to conclude: at the time being, our work raises many more questions than it addresses and so does this abstract. We precisely hope that discussions on the workshop will suggest us ideas to pursue our work and can in turn stimulate other participants to look in this direction.

To finish on a perspective, we think that understanding the structure of the infinitary calculi induced by calculi such as $\Lambda\mu$ can teach us a lot about properties such as separability. Already in the case of Böhm trees can we see intriguing phenomena: $\Lambda\mu$-$\mathfrak{BT}$ for $\lambda\mu$-terms have (up to finitely many η_S-expansions) the following structure:

$$\boxed{\mathfrak{B}^{\lambda\mu} ::= \Omega \mid x \mid \Lambda(x_i)_{i\in\omega}.(y)(\mathfrak{B}_j^{\lambda\mu})_{j\in\omega}}$$

One can observe that there is no freedom on the arity in these Böhm trees: the index is always ω and the arities are forced to match. This witnesses, directly in the structure of Böhm trees, the failure of separation theorem for $\lambda\mu$-calculus.

Acknowledgements The second author wants to thank Jeroen Ketema for pointing him to [7].

References

[1] Henk Barendregt and Jan Willem Klop. Applications of infinitary lambda calculus. *Inf. Comput.*, 207(5):559–582, May 2009.

[2] René David and Walter Py. $\lambda\mu$-calculus and Böhm's theorem. *Journal of Symbolic Logic*, 2001.

[3] Mariangiola Dezani-Ciancaglini and Elio Giovannetti. From Böhm's theorem to observational equivalences: an informal account. *Electr. Notes Theor. Comput. Sci.*, 50(2):83–116, 2001.

[4] Marco Gaboardi and Alexis Saurin. A foundational calculus for computing with streams. In *ICTCS*, 2010.

[5] Richard Kennaway, Jan Willem Klop, M. Ronan Sleep, and Fer-Jan de Vries. Transfinite reductions in orthogonal term rewriting systems. *Inf. Comput.*, 119(1):18–38, 1995.

[6] Richard Kennaway, Jan Willem Klop, M. Ronan Sleep, and Fer-Jan de Vries. Infinitary lambda calculus. *Theor. Comput. Sci.*, 175(1):93–125, 1997.

[7] Jeroen Ketema, Stefan Blom, Takahito Aoto, and Jakob Grue Simonsen. Rewriting transfinite terms. In *Liber Amicorum for Roel de Vrijer*, pages 129–144. Lulu, Raleigh, NC, September 2009.

[8] Koji Nakazawa and Shin ya Katsumata. Extensional models of untyped lambda-mu calculus. In Herman Geuvers and Ugo de'Liguoro, editors, *CL&C*, volume 97 of *EPTCS*, pages 35–47, 2012.

[9] Michel Parigot. Lambda-mu-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR*, pages 190–201, 1992.

[10] Alexis Saurin. Separation with streams in the $\Lambda\mu$-calculus. In *LICS*, pages 356–365. IEEE, 2005.

[11] Alexis Saurin. A hierarchy for delimited continuations in call-by-name. In *FOSSACS 2010*, LNCS. Springer, March 2010.

[12] Alexis Saurin. Böhm theorem and Böhm trees for the $\Lambda\mu$-calculus. *Theor. Comput. Sci.*, 435:106–138, 2012.