Finitary semantics and profinite λ-terms

Vincent Moreau, j.w.w. Sam van Gool, Paul-André Melliès and Tito Nguyễn

LoVe seminar, LIPN

November the 9th, 2023

IRIF, Université Paris Cité, Inria Paris

Motivations: on automata and λ-calculus

Automata and λ-calculus: the syntactic side

Any finite word can be encoded as a λ-term through the Church encoding:

$$abb \in \{a, b\}^* \quad \rightsquigarrow \quad \lambda(a : \mathtt{o} \Rightarrow \mathtt{o}).\lambda(b : \mathtt{o} \Rightarrow \mathtt{o}).\lambda(c : \mathtt{o}).\, b\,(b\,(a\,c)) \ .$$

More generally, any finite ranked tree can be encoded as a λ-term:

$$\begin{tikzcd} & c & c \\ a \arrow[r] \arrow[ur] & a \arrow[r] \arrow[ur] & c \end{tikzcd} \quad \rightsquigarrow \quad \lambda(a : \mathtt{o} \Rightarrow \mathtt{o} \Rightarrow \mathtt{o}).\lambda(c : \mathtt{o}).a\,c\,(a\,c\,c)$$

1. The simply typed λ-calculus generalizes finite words and trees.

Automata and λ-calculus: the semantic side

If Q is a finite set, $\delta_a : Q \to Q$, $\delta_b : Q \to Q$ and $q_0 \in Q$, then

$$[\![\lambda a.\lambda b.\lambda c.\, b\,(b\,(a\,c))]\!]_Q(\delta_a, \delta_b, q_0) \quad = \quad \delta_b(\delta_b(\delta_a(q_0))) \ ,$$

so interpreting the encoding of a words amounts to running it in an automaton.

The same observation holds for finite ranked trees: if $\delta_a : Q \times Q \to Q$, then

$$[\![\lambda a.\lambda c.a\,c\,(a\,c\,c)]\!]_Q(\delta_a, q_0) \quad = \quad \delta_a(q_0, \delta_a(q_0, q_0)) \ .$$

2. Semantics of λ-calculus generalize the interpretation in finite automata.

A topological approach to automata theory

If w and w' are two finite words on a same alphabet, then

$$d(w, w') = 2^{-r(w,w')} \quad \text{where } r(w, w') = \min\{|\mathcal{A}| \mid \mathcal{A} \text{ distinguishes } w \text{ and } w'\}$$

is a metric on finite words. Intuitively, two words w and w' are

- far if there is a small automaton distinguishing them, e.g. a^{2n} and a^{2n+1}.
- close if only a large automaton can distinguish them, e.g. $a^{n!}$ and $a^{(n+1)!}$.

The metric completion $\widehat{\Sigma^*}$ of Σ^* is the space of profinite words. In particular, if u is a profinite run and $\mathcal{A} = (Q, \delta, q_0)$ is an automaton, then we get a state $q_u \in Q$.

3. Profinite words form a syntax for interpretation into finite automata.

The topic of this talk

4. **We define profinite λ-terms and show some of their properties.**

- They assemble into a CCC ProLam.
- The Church encoding can be extended to a homeomorphism

$$\begin{tikzcd} \widehat{\Sigma^*} \arrow[r, "\sim"] & \widehat{\Lambda}_{\beta\eta}\langle \mathrm{Church}_\Sigma \rangle \\ \Sigma^* \arrow[u, hook] \arrow[r, "\sim"] & \Lambda_{\beta\eta}\langle \mathrm{Church}_\Sigma \rangle \arrow[u, hook] \end{tikzcd}$$

- They provide a structure to understand some operations on profinite words.
- They are the Stone dual of regular languages of λ-terms.
- They live in harmony with the principles of Reynolds parametricity.

Languages

Regular languages of words

Let Σ be a finite alphabet, M be a finite monoid and $p : \Sigma \to M$ a set-theoretic function. We write $\bar{p}$ for the associated monoid homomorphism $\Sigma^* \to M$.

For each subset $F \subseteq M$, the set

$$L_F \quad := \quad \{w \in \Sigma^* \mid \bar{p}(w) \in F\}$$

is a regular language. These sets assemble into the Boolean algebra

$$\mathrm{Reg}_M\langle\Sigma\rangle \quad := \quad \{L_F : F \subseteq M\} \ .$$

When M ranges over all finite monoids, we get in this way all regular languages:

$$\mathrm{Reg}\langle\Sigma\rangle \quad = \quad \bigcup_M \mathrm{Reg}_M\langle\Sigma\rangle \ .$$

The Church encoding for words

Any natural number n can be encoded in the simply typed λ-calculus as

$$\lambda(s : o \Rightarrow o).\lambda(z : o).\,\underbrace{s\,(\ldots(s\,z))}_{n \text{ applications}} \quad : \quad (o \Rightarrow o) \Rightarrow o \Rightarrow o\,.$$

A natural number is just a word over a one-letter alphabet.

For any alphabet $\Sigma = \{a_1, \ldots, a_N\}$, any word $w = a_{i_1} \ldots a_{i_n} \in \Sigma^*$ can be encoded as

$$\lambda(a_1 : o \Rightarrow o) \ldots \lambda(a_N : o \Rightarrow o).\lambda(c : o).a_{i_n}\,(\ldots(a_{i_1}\,c))$$

which is a closed λ-term of type

$$\mathrm{Church}_\Sigma \quad := \quad \underbrace{(o \Rightarrow o) \Rightarrow \ldots \Rightarrow (o \Rightarrow o)}_{N \text{ times}} \Rightarrow o \Rightarrow o\,.$$

Categorical interpretation

Let C be a cartesian closed category and c be one of its objects.

For any simple type A built from o, we define the object $[\![A]\!]_c$ by induction as

$$[\![\mathrm{o}]\!]_c \;:=\; c \qquad \text{and} \qquad [\![A \Rightarrow B]\!]_c \;:=\; [\![A]\!]_c \Rightarrow [\![B]\!]_c \;.$$

Using the cartesian closed structure, one defines an interpretation function

$$[\![-]\!]_c \quad : \quad \Lambda_{\beta\eta}\langle A\rangle \;\longrightarrow\; \mathsf{C}(1, [\![A]\!]_c)\;.$$

Given an object c of C used to interpret o, every word w over the alphabet $\Sigma = \{a, b\}$, seen as a λ-term, is interpreted as a morphism

$$[\![w]\!]_c \quad \in \quad \mathsf{C}(1, (c \Rightarrow c) \Rightarrow (c \Rightarrow c) \Rightarrow c \Rightarrow c)$$

which describes how the word will interact with a deterministic automaton.

Regular languages of λ-terms

The notion of regular language of λ-terms has been introduced by Salvati.

For any object c and any subset $F \subseteq \mathsf{C}(1, [\![A]\!]_c)$, we define the language

$$L_F \quad := \quad \{t \in \Lambda_{\beta\eta}\langle A\rangle \mid [\![t]\!]_c \in F\} \ .$$

All the languages recognized by c assemble into a Boolean algebra

$$\mathrm{Reg}_c\langle A\rangle \quad := \quad \{L_F \mid F \subseteq \mathsf{C}(1, [\![A]\!]_c)\} \ .$$

We can then make c range over all objects of C, and we get the definition

$$\mathrm{Reg}\langle A\rangle \quad := \quad \bigcup_c \mathrm{Reg}_c\langle A\rangle \ .$$

Notice that $\mathrm{Reg}\langle A\rangle$ has no reason to be a Boolean algebra for the moment.

Salvati generalizes Kleene in FinSet

The Church encoding induces an isomorphism of Boolean algebras

$$\mathrm{Reg}\langle \mathrm{Church}_\Sigma \rangle_{\mathsf{FinSet}} \quad \cong \quad \mathrm{Reg}\langle \Sigma \rangle \ .$$

In the CCC FinSet, maps $1 \to Q$ are elements $q \in Q$.

Indeed, every automaton $(Q, \delta, q_0, \mathrm{Acc})$ induces a subset

$$F \quad := \quad \{ q \in [\![\mathrm{Church}_\Sigma]\!]_Q \mid q(\delta, q_0) \in \mathrm{Acc} \}$$

On the other hand, every $q \in [\![\mathrm{Church}_\Sigma]\!]_Q$ induces a finite family of automata

$$(Q, \delta, q_0, \{q(\delta, q_0)\}) \qquad \text{for all } \delta : \Sigma \times Q \to Q \text{ and } q_0 \in Q$$

which determines the behavior of q, and from which one gets finite monoids.

About FinSet and other CCCs

We have seen that FinSet gives the usual notion of regular language for Church types.

If C is a CCC, we say that it is finitely pointable if there exists a faithful product-preserving functor

$$P \quad : \quad \mathsf{C} \longrightarrow \mathsf{FinSet}\ .$$

Finitely pointable CCCs are the ones which can be embedded into locally finite well-pointed ones. Example: $[\mathsf{J}, \mathsf{FinSet}]$, where J is a finite category.

Theorem. If C is finitely pointable[1], then $\mathrm{Reg}\langle A\rangle_{\mathsf{C}} = \mathrm{Reg}\langle A\rangle_{\mathsf{FinSet}}$.

As a corollary, we get that the set $\mathrm{Reg}\langle A\rangle$ is a Boolean algebra.

[1] and has two distinct parallel morphisms

Entering the profinite world

The monoid of profinite words

A **profinite word** u is a family (u_p) of elements

$$u_p \in M \qquad \text{where} \qquad \begin{array}{r} M \text{ ranges over all finite monoids} \\ p : \Sigma \to M \text{ ranges over all functions} \end{array}$$

such that for every function $p : \Sigma \to M$ and homomorphism $\varphi : M \to N$, with M and N finite monoids, we have $u_{\varphi \circ p} = \varphi(u_p)$.

The monoid $\widehat{\Sigma^*}$ of profinite words contains Σ^* as a submonoid, since any word $w = w_1 \dots w_n$, where each $w_i \in \Sigma$, induces a profinite word with components

$$p(w_1) \dots p(w_n) \qquad \text{for all } p : \Sigma \to M.$$

A profinite word which is not a word

For any finite monoid M there exists $n(M) \geq 1$ such that for all elements m of M, the element $m^{n(M)}$ is the idempotent power of m, which is unique.

Let a be any letter in Σ. The family of elements

$$u_p \quad := \quad p(a)^{n(M)} \qquad \text{for all } p : \Sigma \to M$$

is an idempotent profinite word written a^ω which is not a finite word.

There is a more general construction: if u is a profinite word, then one can build another profinite word u^ω which is idempotent.

From the metric viewpoint, it is the limit of the Cauchy sequence $u^{n!}$.

Profinite natural numbers

What does $\widehat{\{a\}^*} \cong \widehat{\mathbb{N}}$, the monoid of profinite natural numbers, look like?

$$0 \text{ ——— } 1 \text{ ——— } 2 \text{ ——— } 3 \cdots\cdots\cdots \qquad ?$$

Profinite natural numbers

What does $\widehat{\{a\}^*} \cong \widehat{\mathbb{N}}$, the monoid of profinite natural numbers, look like?

$$0 \;\text{———}\; 1 \;\text{———}\; 2 \;\text{———}\; 3 \;\cdots\cdots\cdots \qquad \ldots\omega\ldots$$

Profinite natural numbers

What does $\widehat{\{a\}^*} \cong \widehat{\mathbb{N}}$, the monoid of profinite natural numbers, look like?

$$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 3 \cdots\cdots\cdots \qquad \prod_{p \text{ prime}} \mathbb{Z}_p$$

where $\mathbb{Z}_p$ is the ring of p-adic numbers, i.e. $\mathbb{Z}_p \cong \lim_n \mathbb{Z}/p^n\mathbb{Z}$.

Duality: words

Stone spaces, i.e. compact and totally separated spaces, and continuous maps form a category Stone. Boolean algebras and their homomorphisms form a category BA.

There is an equivalence of categories

$$\mathrm{Stone} \quad \cong \quad \mathrm{BA}^{\mathrm{op}}$$

which associates to every Stone space its algebra of clopens and to every Boolean algebra its space of ultrafilters.

In particular, the monoid of profinite words $\widehat{\Sigma^*}$ has a natural topology such that

$$\widehat{\Sigma^*} \qquad \text{is the Stone dual of} \qquad \mathrm{Reg}\langle\Sigma\rangle\ .$$

Duality: λ-terms

For any simple type A and finite set Q, we consider the subset

$$[\![A]\!]^\bullet_Q \quad := \quad \{[\![t]\!]_Q \mid t \in \Lambda_{\beta\eta}\langle A\rangle\} \;\subseteq\; [\![A]\!]_Q$$

of definable elements of $[\![A]\!]_Q$. Equivalently, it is the quotient

$$\Lambda_{\beta\eta}\langle A\rangle/\simeq_Q \qquad \text{with } t \simeq_Q s \text{ if and only if } [\![t]\!]_Q = [\![s]\!]_Q \;.$$

The finite set of definable elements is related to regular languages as

$$[\![A]\!]^\bullet_Q \qquad \text{is the Stone dual of} \qquad \mathrm{Reg}_Q\langle A\rangle$$

A first observation using logical relations

If Q and Q' are two finite sets and $R \subseteq Q \times Q'$, for any simple type A we have

$$[\![A]\!]_R \quad \subseteq \quad [\![A]\!]_Q \times [\![A]\!]_{Q'}$$

In particular, if $f : Q \twoheadrightarrow Q'$ is a partial surjection, then so is $[\![A]\!]_f : [\![A]\!]_Q \twoheadrightarrow [\![A]\!]_{Q'}$.

Using the fundamental lemma of logical relations, one can deduce that

$$\text{if} \quad |Q| \ \geq \ |Q'| \ , \qquad \text{then} \quad \mathrm{Reg}_{Q'}\langle A \rangle \ \subseteq \ \mathrm{Reg}_Q\langle A \rangle \ .$$

This shows that the diagram

$$\Big(\mathrm{Reg}_{Q'}\langle A \rangle \longrightarrow \mathrm{Reg}_Q\langle A \rangle \Big)_{f : Q \twoheadrightarrow Q'}$$

is directed so we have

$$\mathrm{Reg}\langle A \rangle \quad = \quad \mathrm{colim}_Q \, \mathrm{Reg}_Q\langle A \rangle \ .$$

Dualizing the diagram

$$
\begin{tikzcd}[column sep=small]
 & & & & \mathrm{Reg}\langle A\rangle & & & & \\
\mathrm{Reg}_{Q_4}\langle A\rangle \arrow[urrrr, hook, dashed] & & \cdots & & \mathrm{Reg}_{Q_5}\langle A\rangle \arrow[u, hook, dashed] & & \cdots & & \mathrm{Reg}_{Q_6}\langle A\rangle \arrow[ullll, hook, dashed] \\
 & & \mathrm{Reg}_{Q_2}\langle A\rangle \arrow[ull, hook] \arrow[urr, hook] & & \cdots & & \mathrm{Reg}_{Q_3}\langle A\rangle \arrow[ull, hook] \arrow[urr, hook] & & \\
 & & & & \mathrm{Reg}_{Q_1}\langle A\rangle \arrow[ull, hook] \arrow[urr, hook] & & & &
\end{tikzcd}
$$

Dualizing the diagram

$$\begin{tikzcd}
 & & & \text{profinite } \lambda\text{-terms of type } A \arrow[dlll, dashed, two heads] \arrow[d, dashed, two heads] \arrow[drrr, dashed, two heads] & & & \\
{[\![A]\!]^\bullet_{Q_4}} \arrow[dr, two heads] & \cdots & & {[\![A]\!]^\bullet_{Q_5}} \arrow[dll, two heads] \arrow[drr, two heads] & & \cdots & {[\![A]\!]^\bullet_{Q_6}} \arrow[dl, two heads] \\
 & {[\![A]\!]^\bullet_{Q_2}} \arrow[drr, two heads] & & \cdots & & {[\![A]\!]^\bullet_{Q_3}} \arrow[dll, two heads] & \\
 & & & {[\![A]\!]^\bullet_{Q_1}} & & &
\end{tikzcd}$$

Definition of profinite λ-terms

By dualizing the diagram defining $\mathrm{Reg}\langle A\rangle$, we obtain a codirected diagram

$$\left([\![A]\!]^{\bullet}_{f} \;:\; [\![A]\!]^{\bullet}_{Q} \longrightarrow\!\!\!\!\!\rightarrow [\![A]\!]^{\bullet}_{Q'} \right)_{f : Q \twoheadrightarrow Q'}$$

and we define $\widehat{\Lambda}_{\beta\eta}\langle A\rangle$ as its limit. As expected,

$$\widehat{\Lambda}_{\beta\eta}\langle A\rangle \qquad \text{is the Stone dual of} \qquad \mathrm{Reg}\langle A\rangle \;.$$

Concretely: a **profinite λ-term** θ of type A is a family of elements $\theta_Q \in [\![A]\!]^{\bullet}_{Q}$ s.t.

$$[\![A]\!]^{\bullet}_{f}(\theta_Q) \;=\; \theta_{Q'} \qquad \text{for every partial surjection } f : Q \twoheadrightarrow Q'.$$

The CCC of profinite λ-terms

Theorem. The profinite λ-terms assemble into a CCC ProLam such that

$$\mathrm{ProLam}(A, B) \quad := \quad \widehat{\Lambda}_{\beta\eta}\langle A \Rightarrow B\rangle \ .$$

This means that we a compositional notion of profinite λ-calculus.

The interpretation of the simply typed λ-calculus into ProLam yields a functor

$$\mathrm{Lam} \ \longrightarrow \ \mathrm{ProLam}$$

which sends a simply typed λ-term t of type A on the profinite λ-term

$$[\![t]\!]_Q \qquad \text{where } Q \text{ ranges over all finite sets.}$$

This assignment is injective thanks to Statman's finite completeness theorem.

Profinite λ-terms of Church type are profinite words

The Church encoding gives a bijection

$$\Lambda_{\beta\eta}\langle \mathsf{Church}_\Sigma \rangle \quad \cong \quad \Sigma^* \ .$$

This extends to the profinite setting. Indeed, profinite λ-terms of simple type Church_Σ are exactly profinite words as we have a homeomorphism

$$\widehat{\Lambda}_{\beta\eta}\langle \mathsf{Church}_\Sigma \rangle \quad \cong \quad \widehat{\Sigma^*} \ .$$

The profinite λ-term Ω

We consider the profinite λ-term Ω of type $(\oplus \Rightarrow \oplus) \Rightarrow \oplus \Rightarrow \oplus$ such that

$$\Omega_Q \quad : \quad f \longmapsto \underbrace{f \circ \cdots \circ f}_{n \text{ times}}$$

where f^n is the idempotent power of the element f of the finite monoid $Q \Rightarrow Q$.

Using Ω, for any Σ of cardinal n, one gets the profinite λ-term

$$\lambda u \lambda a_1 \ldots \lambda a_n . \Omega\,(u\, a_1\ \ldots\ a_n) \quad : \quad \mathrm{Church}_\Sigma \Rightarrow \mathrm{Church}_\Sigma$$

which is the representation in the profinite λ-calculus of the operator

$$(-)^\omega \quad : \quad \widehat{\Sigma^*} \longrightarrow \widehat{\Sigma^*}$$

on profinite words.

Parametric families

Let A be a simple type. A **parametric family** θ is a family of elements $\theta_Q \in [\![A]\!]_Q$ s.t.

$$(\theta_Q, \theta_{Q'}) \;\in\; [\![A]\!]_R \qquad \text{for all relations } R \subseteq Q \times Q'.$$

Two differences with profinite λ-terms:

- the element θ_Q is not asked to be definable...
- ...but the family is parametric with respect to all relations.

A theorem and its partial converse

We first have a general theorem at every type.

Theorem. Every profinite λ-term is a parametric family.

This theorem admits the following converse at Church types.

Theorem. Every parametric family of type Church_Σ is a profinite λ-term.

The proof of the converse uses the λ-terms

$$\lambda s \lambda z. z \ : \ \mathsf{Nat} \quad \text{and} \quad \lambda n \lambda s \lambda z. s\,(n\,s\,z) \ : \ \mathsf{Nat} \Rightarrow \mathsf{Nat}$$

which behave like constructors of the simple type $\mathsf{Nat} := \mathsf{Church}_1$.

Conclusion

Current work:

- show that the natural categorical definition of profinite trees coincides with profinite λ-terms of tree-Church types.

Future work:

- generalize the parametricity theorem to any simple type;
- investigate the universal property of ProLam.

Conclusion

Current work:

- show that the natural categorical definition of profinite trees coincides with profinite λ-terms of tree-Church types.

Future work:

- generalize the parametricity theorem to any simple type;
- investigate the universal property of ProLam.

Thank you for your attention!

Any questions?

Bibliography

[Geh16] Mai Gehrke. **"Stone duality, topological algebra, and recognition".** In: *Journal of Pure and Applied Algebra* 220.7 (2016), pp. 2711–2747. ISSN: 0022-4049. DOI: https://doi.org/10.1016/j.jpaa.2015.12.007.

[Mel17] Paul-André Melliès. **"Higher-order parity automata".** In: *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, 2017*. 2017, pp. 1–12.

[Pin] Jean-Eric Pin. **"Profinite Methods in Automata Theory".** In: *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009*. IBFI Schloss Dagstuhl. URL: https://hal.inria.fr/inria-00359677.

[Sal09] Sylvain Salvati. **"Recognizability in the Simply Typed Lambda-Calculus".** In: *16th Workshop on Logic, Language, Information and Computation*. Vol. 5514. Lecture Notes in Computer Science. Tokyo Japan: Springer, 2009, pp. 48–60.