Profinite $\lambda$-terms and parametricity

Vincent Moreau, joint work with Sam van Gool and Paul-André Melliès

MFPS XXXIX, Indiana University Bloomington
June the 22th, 2023

IRIF, Université Paris Cité, Inria Paris
Context of the talk

Regular languages have a central place in theoretical computer science. Profinite methods are well established for words using finite monoids.

Salvati proposed a notion of regular language of \( \lambda \)-terms using semantic tools.

Contribution: definition of profinite \( \lambda \)-terms using the CCC \textbf{FinSet} such that

\[
\text{profinite words are in bijection with profinite } \lambda\text{-terms of Church type}
\]

and living in harmony with Stone duality and the principles of Reynolds parametricity.
Languages
Regular languages of words

Let $\Sigma$ 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

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

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

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

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

$$\operatorname{Reg} \langle \Sigma \rangle = \bigcup_M \operatorname{Reg}_M \langle \Sigma \rangle.$$
The Church encoding for words

Any natural number $n$ can be encoded in the simply typed $\lambda$-calculus as

$$s : \emptyset \Rightarrow \emptyset, \quad z : \emptyset \vdash s(\ldots(s\,z)) : \emptyset. \quad \text{n applications}$$

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

For example, the word $abba$ over the two-letter alphabet $\{a, b\}$

$$a : \emptyset \Rightarrow \emptyset, \quad b : \emptyset \Rightarrow \emptyset, \quad c : \emptyset \vdash a(b(b(a\,c))) : \emptyset.$$ is encoded as the closed $\lambda$-term

$$\lambda a.\lambda b.\lambda c.a(b(b(a\,c))) : (\emptyset \Rightarrow \emptyset) \Rightarrow (\emptyset \Rightarrow \emptyset) \Rightarrow \emptyset \Rightarrow \emptyset.$$

For any alphabet $\Sigma$, we define $\text{Church}_\Sigma$ as

$$\big(\emptyset \Rightarrow \emptyset\big) \Rightarrow \ldots \Rightarrow \big(\emptyset \Rightarrow \emptyset\big) \Rightarrow \emptyset \Rightarrow \emptyset. \quad |\Sigma| \times \text{times}$$
Categorical interpretation

Let $\mathbf{C}$ be a cartesian closed category and $Q$ be one of its objects.

For any simple type $A$ built from $\varnothing$, we define the object $[A]_Q$ by induction as

$$
[A]_Q := Q 
$$

and

$$
[A \to B]_Q := [A]_Q \to [B]_Q.
$$

Using the cartesian closed structure, one defines an interpretation function

$$
[-]_Q : \Lambda_{\beta\eta}(A) \to \mathbf{C}(1, [A]_Q).
$$

In $\text{FinSet}$ which is cartesian closed, given a finite set $Q$ used to interpret $\varnothing$, every word $w$ over the alphabet $\Sigma = \{a, b\}$, seen as a $\lambda$-term, is interpreted as a point

$$
[w]_Q \in (Q \to Q) \to (Q \to Q) \to Q \to Q
$$

which describes how the word will interact with a deterministic automaton.
Regular languages of $\lambda$-terms

The notion of regular language of $\lambda$-terms has been introduced by Salvati. For any finite set $Q$ and any subset $F \subseteq [[A]]_Q$, we define the language

$$L_F := \{ M \in \Lambda_{\beta\eta}(A) \mid [[M]]_Q \in F \}.$$ 

All the languages recognized by $Q$ assemble into a Boolean algebra

$$\text{Reg}_Q(A) := \{ L_F \mid F \subseteq [[A]]_Q \}.$$ 

We can then make $Q$ range over all finite sets, and we get the definition

$$\text{Reg}(A) := \bigcup_Q \text{Reg}_Q(A).$$ 

Notice that $\text{Reg}(A)$ has no reason to be a Boolean algebra for the moment.
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

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

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

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

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

This shows that the diagram

$$\left( \text{Reg}_{Q'} \langle A \rangle \hookrightarrow \text{Reg}_Q \langle A \rangle \right)_{f : Q \rightarrow Q'}$$

is directed so we have

$$\text{Reg} \langle A \rangle = \text{colim}_Q \text{Reg}_Q \langle A \rangle.$$
Entering the profinite world
The monoid of profinite words

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

$$u_p \in M$$

where

$M$ ranges over all finite monoids

$p : \Sigma \rightarrow M$ ranges over all functions

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

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

$$p(w_1) \ldots p(w_n) \quad \text{for all } p : \Sigma \rightarrow M.$$
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 $\Sigma$. The family of elements

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

is an idempotent profinite word written $a^\omega$ 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 $u^\omega$ which is idempotent.
What does \( \{a\}^* \cong \hat{\mathbb{N}} \), the monoid of profinite natural numbers, look like?

\[
0 \quad 1 \quad 2 \quad 3 \quad \ldots \quad ?
\]
Profinite natural numbers

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

0 ---- 1 ---- 2 ---- 3 ............ ...\(\omega\)....
Profinite natural numbers

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

0 —— 1 —— 2 —— 3  ..........  $\prod_{p \text{ prime}} \mathbb{Z}_p$

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 $\text{Stone}$. Boolean algebras and their homomorphisms form a category $\text{BA}$.

There is an equivalence of categories

$$\text{Stone} \cong \text{BA}^{\text{op}}$$

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

$$\widehat{\Sigma}^* \text{ is the Stone dual of } \text{Reg}(\langle \Sigma \rangle).$$
Duality: $\lambda$-terms

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

$$\llbracket A \rrbracket_Q^* := \{ \llbracket M \rrbracket_Q \mid M \in \Lambda_{\beta\eta}\langle A \rangle \} \subseteq \llbracket A \rrbracket_Q$$

of definable elements of $\llbracket A \rrbracket_Q$. Equivalently, it is the quotient

$$\Lambda_{\beta\eta}\langle A \rangle / \approx_Q \quad \text{with} \quad M \approx_Q N \text{ if and only if } \llbracket M \rrbracket_Q = \llbracket N \rrbracket_Q.$$

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

$$\llbracket A \rrbracket_Q^* \quad \text{is the Stone dual of} \quad \text{Reg}_Q\langle A \rangle$$

and the inclusion $\text{Reg}_Q'\langle A \rangle \hookrightarrow \text{Reg}_Q\langle A \rangle$ induced by a partial surjection $f : Q \twoheadrightarrow Q'$ dualizes to the surjection $\llbracket A \rrbracket_f^* : \llbracket A \rrbracket_Q^* \rightarrow \llbracket A \rrbracket_Q'^*$, which is the restriction of $\llbracket A \rrbracket_f$. 
Dualizing the diagram

\[ \text{Reg} \langle A \rangle \]

\[ \text{Reg}_{Q_1} \langle A \rangle \]
\[ \text{Reg}_{Q_2} \langle A \rangle \]
\[ \text{Reg}_{Q_3} \langle A \rangle \]
\[ \text{Reg}_{Q_4} \langle A \rangle \]
\[ \text{Reg}_{Q_5} \langle A \rangle \]
\[ \text{Reg}_{Q_6} \langle A \rangle \]
Dualizing the diagram

profinite $\lambda$-terms of type $A$

$\llbracket A \rrbracket_{Q_1} \quad \ldots \quad \llbracket A \rrbracket_{Q_5} \quad \ldots \quad \llbracket A \rrbracket_{Q_6}$
Definition of profinite $\lambda$-terms

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

$$
\left( [A]_f^\bullet : [A]_Q^\bullet \rightarrow [A]_{Q'}^\bullet \right)_{f : Q \rightarrow Q'}
$$

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

$$
\hat{\Lambda}_{\beta\eta}\langle A \rangle \quad \text{is the Stone dual of} \quad \text{Reg}\langle A \rangle.
$$

Concretely: a **profinite** $\lambda$-term $\theta$ of type $A$ is a family of elements $\theta_Q \in [A]^\bullet_Q$ s.t.

$$
[A]_f^\bullet(\theta_Q) = \theta_{Q'} \quad \text{for every partial surjection } f : Q \rightarrow Q'.
$$
The CCC of profinite $\lambda$-terms

**Theorem.** The profinite $\lambda$-terms assemble into a CCC $\text{ProLam}$ such that

$$\text{ProLam}(A, B) := \hat{\lambda}_\beta \langle A \Rightarrow B \rangle.$$ 

This means that we a compositional notion of profinite $\lambda$-calculus.

The interpretation of the simply typed $\lambda$-calculus into $\text{ProLam}$ yields a functor

$$\text{Lam} \rightarrow \text{ProLam}$$

which sends a simply typed $\lambda$-term $M$ of type $A$ on the profinite $\lambda$-term

$$[M]_Q$$

where $Q$ ranges over all finite sets.

This assignment is injective thanks to Statman’s finite completeness theorem.
Profinite $\lambda$-terms of Church type are profinite words

The Church encoding gives a bijection

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

This extends to the profinite setting. Indeed, profinite $\lambda$-terms of simple type Church$_\Sigma$ are exactly profinite words as we have a homeomorphism

$$\hat{\Lambda}_{\beta\eta}\langle\text{Church}_\Sigma\rangle \cong \hat{\Sigma}^*.$$
The profinite $\lambda$-term $\Omega$

We consider the profinite $\lambda$-term $\Omega$ of type $(\varnothing \Rightarrow \varnothing) \Rightarrow \varnothing \Rightarrow \varnothing$ such that

$\Omega_Q : f \mapsto f \circ \cdots \circ f$

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

Using $\Omega$, for any $\Sigma$ of cardinal $n$, one gets the profinite $\lambda$-term

$\lambda u \lambda a_1 \ldots \lambda a_n \cdot \Omega (u \ a_1 \ldots a_n) : \text{Church}_\Sigma \Rightarrow \text{Church}_\Sigma$

which is the representation in the profinite $\lambda$-calculus of the operator

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

on profinite words.
Profinite $\lambda$-terms and Reynolds parametricity
Parametric families

Let $A$ be a simple type. A **parametric family** $\theta$ is a family of elements $\theta_Q \in \llbracket A \rrbracket_Q$ s.t.

$$(\theta_Q, \theta_{Q'}) \in \llbracket A \rrbracket_R$$

for all relations $R \subseteq Q \times Q'$.

Two differences with profinite $\lambda$-terms:

- the element $\theta_Q$ 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 $\lambda$-term is a parametric family.

This theorem admits the following converse at Church types.

**Theorem.** Every parametric family of type Church$\Sigma$ is a profinite $\lambda$-term.

The proof of the converse uses the unfolding terms, which generalize the constructors

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

of the simple type $\text{Nat} := \text{Church}_1$ to any Church type.
Conclusion

Future work:

- study the situation in other locally finite CCCs;
- generalize the notion of unfolding term to any simple type;
- investigate a generalization of logic on words, which uses monadic second-order logic (MSO), to a logic on $\lambda$-terms.

Thank you for your attention!

Any questions?
Conclusion

Future work:

• study the situation in other locally finite CCCs;
• generalize the notion of unfolding term to any simple type;
• investigate a generalization of logic on words, which uses monadic second-order logic (MSO), to a logic on $\lambda$-terms.

Thank you for your attention!

Any questions?


Salvati generalizes Kleene

The Church encoding induces an isomorphism of Boolean algebras

\[ \text{Reg} \langle \text{Church}_\Sigma \rangle \cong \text{Reg} \langle \Sigma \rangle. \]

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

\[ F := \{ q \in [\text{Church}_\Sigma]_Q \mid q(\delta, q_0) \in \text{Acc} \}. \]

On the other hand, every \(q \in [\text{Church}_\Sigma]_Q\) induces a finite family of automata

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

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