

# Profinite lambda-terms and parametricity

Sam van Gool<sup>a</sup> Paul-André Melliès<sup>b</sup> Vincent Moreau<sup>c</sup>

<sup>a</sup> *Université Paris Cité*

<sup>b</sup> *CNRS, Université Paris Cité, Inria*

<sup>c</sup> *Université Paris Cité, Inria*

---

## Abstract

Combining ideas coming from Stone duality and Reynolds parametricity, we formulate in a clean and principled way a notion of profinite  $\lambda$ -term which, we show, generalizes at every type the traditional notion of profinite word coming from automata theory. We start by defining the Stone space of profinite  $\lambda$ -terms as a projective limit of finite sets of usual  $\lambda$ -terms, considered modulo a notion of equivalence based on the finite standard model. One main contribution of the paper is to establish that, somewhat surprisingly, the resulting notion of profinite  $\lambda$ -term coming from Stone duality lives in perfect harmony with the principles of Reynolds parametricity. In addition, we show that the notion of profinite  $\lambda$ -term is compositional by constructing a cartesian closed category of profinite  $\lambda$ -terms, and we establish that the embedding from  $\lambda$ -terms modulo  $\beta\eta$ -conversion to profinite  $\lambda$ -terms is faithful using Statman's finite completeness theorem. Finally, we prove that the traditional Church encoding of finite words into  $\lambda$ -terms can be extended to profinite words, and leads to a homeomorphism between the space of profinite words and the space of profinite  $\lambda$ -terms of the corresponding Church type.

*Keywords:* higher-order automata, semantics of lambda-calculus, profinite monoids, Stone duality, regular languages

---

## 1 Introduction

In this paper, we formulate a notion of *profinite  $\lambda$ -term* which, as we will show, extends in a principled way, related to Reynolds parametricity, the important notion of *profinite word* found at the heart of automata theory.

Our starting point is provided by the Church encoding of finite words on a given finite alphabet  $\Sigma = \{a_1, \dots, a_n\}$  into simply typed  $\lambda$ -terms. The idea of the encoding is to view every letter  $a_i \in \Sigma$  as a variable  $a_i$  of type  $\circ \Rightarrow \circ$  where  $\circ$  is an arbitrary base type. Once a variable  $a_i : \circ \Rightarrow \circ$  has been declared in the context for each letter of  $\Sigma$ , a finite word  $w = a_{w_1} \cdots a_{w_k} \in \Sigma^*$  can be naturally viewed as the composite  $a_{w_k} \circ \cdots \circ a_{w_1}$  of type  $\circ \Rightarrow \circ$ . This composite is represented by the  $\lambda$ -term  $\lambda c.a_{w_k}(\cdots(a_{w_1}c))$ , which we note  $W$ , where  $c$  is a variable of type  $\circ$ . The finite word  $w$  is thus encoded as the  $\lambda$ -term defined as  $\lambda a_1 \dots \lambda a_n. \lambda c.a_{w_k}(\cdots(a_{w_1}c))$  which is of type  $\text{Church}_\Sigma$ , defined as

$$\underbrace{(\circ \Rightarrow \circ)}_{\text{type of } a_1} \Rightarrow \cdots \Rightarrow \underbrace{(\circ \Rightarrow \circ)}_{\text{type of } a_n} \Rightarrow \underbrace{\circ}_{\text{type of } c} \Rightarrow \circ,$$

where we have  $n$  occurrences of  $\circ \Rightarrow \circ$ , one for each letter  $a_i \in \Sigma$ , and one occurrence of  $\circ$  for the variable  $c$ , on the left of the base type  $\circ$ . Given a simple type  $A$  generated by the base type  $\circ$ , we write  $\Lambda_{\beta\eta}\langle A \rangle$  for

the set of closed  $\lambda$ -terms of simple type  $A$ , considered modulo  $\beta$ - and  $\eta$ -conversion. The Church encoding induces a one-to-one correspondence

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

between finite words on the alphabet  $\Sigma$  and simply typed  $\lambda$ -terms of type  $\text{Church}_\Sigma$  up to  $\beta\eta$ -equivalence. The correspondence allows us to think of finite words on the finite alphabet  $\Sigma$  as simply typed  $\lambda$ -terms of that specific type.

### *The finite set interpretation and deterministic automata*

The connection between the Church encoding of finite words and automata theory has been considered in syntactic [16, 22, 27] and semantic [13, 14, 20, 27] contexts.

Here, we follow the semantic track and focus on the finitary interpretation of the simply typed  $\lambda$ -calculus in the cartesian closed category **FinSet** of finite sets and functions between them, which, we claim, corresponds to deterministic finite state automata. In order to define this interpretation, we start by choosing a finite set  $Q$  which lets us define, for any simple type  $A$ , a finite set  $\llbracket A \rrbracket_Q$  in which we will interpret  $\lambda$ -terms of type  $A$ . This set  $\llbracket A \rrbracket_Q$  is inductively defined by

$$\llbracket \circ \rrbracket_Q := Q \quad \text{and} \quad \llbracket A \Rightarrow B \rrbracket_Q := \llbracket A \rrbracket_Q \Rightarrow \llbracket B \rrbracket_Q$$

where we interpret the functional type  $A \Rightarrow B$  as the finite set of set-theoretic functions from the set  $\llbracket A \rrbracket_Q$  to the set  $\llbracket B \rrbracket_Q$ . The interpretation then transports every simple type  $A$  to a finite set  $\llbracket A \rrbracket_Q$  and every simply typed  $\lambda$ -term  $M$  of type

$$a_1 : A_1, \dots, a_n : A_n \quad \vdash \quad M : B$$

to a function between finite sets

$$\llbracket M \rrbracket_Q \quad : \quad \llbracket A_1 \rrbracket_Q \times \dots \times \llbracket A_n \rrbracket_Q \longrightarrow \llbracket B \rrbracket_Q .$$

This interpretation in **FinSet** induces, on closed terms, a function  $\llbracket - \rrbracket_Q : \Lambda_{\beta\eta}\langle A \rangle \longrightarrow \llbracket A \rrbracket_Q$  which is called the *semantic bracket* and transports every closed  $\lambda$ -term  $M$  of type  $A$  to its interpretation  $\llbracket M \rrbracket_Q \in \llbracket A \rrbracket_Q$ . In order to understand the connection with finite automata, it is instructive to examine how the interpretation acts on the open  $\lambda$ -term  $W$  encoding the finite word  $w = a_{w_1} \dots a_{w_k} \in \Sigma^*$ . By construction, the  $\lambda$ -term  $W$  is of type

$$a_1 : \circ \Rightarrow \circ, \dots, a_n : \circ \Rightarrow \circ \quad \vdash \quad W \quad : \quad \circ \Rightarrow \circ$$

where each letter  $a_1, \dots, a_n \in \Sigma$  appears as a variable of type  $\circ \Rightarrow \circ$  in the context. The  $\lambda$ -term  $W$  is then interpreted as the functional

$$\llbracket W \rrbracket_Q \quad : \quad (Q \Rightarrow Q) \times \dots \times (Q \Rightarrow Q) \longrightarrow (Q \Rightarrow Q)$$

which transports an  $n$ -tuple  $f_1, \dots, f_n$  of endofunctions on the finite set  $Q$ , i.e. elements of the set  $Q \Rightarrow Q$ , to the composite endofunction  $f_{w_k} \circ \dots \circ f_{w_1}$  on the same finite set, that is,

$$\llbracket W \rrbracket_Q = f_1, \dots, f_n \mapsto f_{w_k} \circ \dots \circ f_{w_1} . \tag{1}$$

A *deterministic finite state automaton* on the alphabet  $\Sigma = \{a_1, \dots, a_n\}$  is defined as a tuple  $\mathcal{A} = (Q, \delta, q_0, \text{Acc})$  consisting of a finite set  $Q$  of *states*, a *transition function*  $\delta : \Sigma \times Q \rightarrow Q$ , an *initial state*  $q_0 \in Q$  and a set of *accepting states*  $\text{Acc} \subseteq Q$ . The transition function  $\delta$  gives rise to a family of transition functions

$$\delta_{a_1} = \delta(a_1, -), \dots, \delta_{a_n} = \delta(a_n, -) \quad : \quad Q \longrightarrow Q$$

where  $\delta_a(q) = q'$  means that the automaton  $\mathcal{A}$  in state  $q$  transitions to the state  $q'$  when it encounters the letter  $a \in \Sigma$ .

Now observe that, if we apply the interpretation (1) of the simply typed  $\lambda$ -term  $W$  in **FinSet** to these transition functions  $\delta_{a_1}, \dots, \delta_{a_n}$ , then we obtain the endofunction

$$\delta_w = \llbracket W \rrbracket_Q(\delta_{a_1}, \dots, \delta_{a_n}) \quad : \quad Q \longrightarrow Q$$

which transforms each input state  $q_0 \in Q$  into the output state  $q_f = \delta_w(q_0) \in Q$  obtained by running the deterministic automaton  $\mathcal{A}$  on the finite word  $w$  encoded by the simply typed  $\lambda$ -term  $W$ . This simple observation establishes the connection between deterministic automata and the interpretation of simply typed  $\lambda$ -terms of type  $\text{Church}_\Sigma$  in **FinSet**.

In this way, if  $\mathcal{A} = (Q, \delta, q_0, \text{Acc})$  is a deterministic finite state automaton, then the tuple  $(Q, \delta, q_0)$  induces an evaluation function

$$\text{eval}_{(\delta, q_0)} \quad : \quad \llbracket \text{Church}_\Sigma \rrbracket_Q \longrightarrow Q$$

which transports every functional  $F \in \llbracket \text{Church}_\Sigma \rrbracket_Q$  to the state  $F(\delta_{a_1}, \dots, \delta_{a_k})(q_0)$  in  $Q$ . Precomposing the evaluation function  $\text{eval}_{(\delta, q_0)}$  with the semantic bracket  $\llbracket - \rrbracket_Q$  induces a composite function

$$\Sigma^* \cong \Lambda_{\beta\eta}\langle \text{Church}_\Sigma \rangle \longrightarrow \llbracket \text{Church}_\Sigma \rrbracket_Q \longrightarrow Q \tag{2}$$

which associates a finite word  $w \in \Sigma^*$  with the final state  $q_f = \delta_w(q_0)$  returned by the automaton. The inverse image of the set  $\text{Acc} \subseteq Q$  under this composite function is, by definition, the *regular language*  $L_{\mathcal{A}}$  of finite words recognized by the deterministic automaton  $\mathcal{A}$ .

#### *The Boolean algebra $\text{Reg}\langle A \rangle$ of regular languages*

The regular language  $L_{\mathcal{A}}$  described above is an element of the Boolean algebra  $\text{Reg}_Q\langle \text{Church}_\Sigma \rangle$  of regular languages of  $\lambda$ -terms of type  $\text{Church}_\Sigma$  recognized by the finite set  $Q$ . This algebra may be defined as the image of the Boolean algebra homomorphism  $\llbracket - \rrbracket_Q^{-1}$  from  $\wp(\llbracket \text{Church}_\Sigma \rrbracket_Q)$  to  $\wp(\Sigma^*)$ , obtained by applying the contravariant power set functor  $\wp : \mathbf{FinSet}^{\text{op}} \rightarrow \mathbf{BA}$  to the semantic bracket  $\llbracket - \rrbracket_Q$ . In the theory of regular languages of simply typed  $\lambda$ -terms developed by Salvati [27, 28], this point of view is extended to any type. The Boolean algebra  $\text{Reg}_Q\langle A \rangle$  of regular languages of  $\lambda$ -terms of higher-order type  $A$  recognizable by a finite set  $Q$  of states is defined as the image of the Boolean algebra homomorphism

$$\llbracket - \rrbracket_Q^{-1} \quad : \quad \wp(\llbracket A \rrbracket_Q) \longrightarrow \wp(\Lambda_{\beta\eta}\langle A \rangle).$$

In other words, a set  $L$  of  $\lambda$ -terms of type  $A$  is recognizable by the finite set  $Q$  precisely when it is of the form  $\llbracket - \rrbracket_Q^{-1}(\text{Acc}) = \{M \in \Lambda_{\beta\eta}\langle A \rangle \mid \llbracket M \rrbracket_Q \in \text{Acc}\}$  for some choice  $\text{Acc} \subseteq \llbracket A \rrbracket_Q$  of a set of accepting elements. Now, letting  $Q$  range over all finite sets, the collection  $\text{Reg}\langle A \rangle \subseteq \wp(\Lambda_{\beta\eta}\langle A \rangle)$  of regular languages of  $\lambda$ -terms of type  $A$  is defined in [27, Def. 1] as

$$\text{Reg}\langle A \rangle = \bigcup \{ \text{Reg}_Q\langle A \rangle \mid Q \text{ a finite set} \}.$$

Salvati [27, Thm. 8] then establishes that  $\text{Reg}\langle A \rangle$  is a Boolean algebra, which boils down to the fact that  $\text{Reg}\langle A \rangle$  is closed under intersection. The proof relies on a presentation of higher-order automata based on intersection types, and on the construction of a product higher-order automaton.

#### *Profinite words in automata theory*

The monoid  $\widehat{\Sigma}^*$  of profinite words on a finite alphabet  $\Sigma$  plays an important role in automata theory, where profinite words encode the limiting behaviour of finite words with respect to deterministic finite automata [23]. For example, one can define an **idempotent power operator** over profinite words, using the fact that the automata are finite, appearing in Equation 14 within §7 where we explain its construction. The monoid  $\widehat{\Sigma}^*$  is the free profinite monoid generated by  $\Sigma$  and can be constructed as the limit, computed in the category **Mon** of monoids, of the codirected (also known as *projective*) system of finite monoid homomorphisms

$$\left( \Sigma^* / \phi \longrightarrow \Sigma^* / \phi' \right)_{\phi \subseteq \phi'} \tag{3}$$

where  $\phi$  and  $\phi'$  range over the finite index congruences on  $\Sigma^*$ , subject to the condition that  $\phi \subseteq \phi'$ . Note that every such finite index congruence  $\phi$  can be seen equivalently as a surjective homomorphism  $h : \Sigma^* \rightarrow M$  to the finite monoid  $M = \Sigma^*/\phi$  whose elements are the equivalence classes of the congruence  $\phi$ . The surjectivity condition on  $h$  can be relaxed in order to show that the monoid  $\widehat{\Sigma^*}$  of profinite words is in fact the codirected limit of the composite functor

$$\Sigma^*/\mathbf{FinMon} \xrightarrow{\pi} \mathbf{FinMon} \hookrightarrow \mathbf{Mon} \quad (4)$$

where  $\mathbf{FinMon}$  denotes the category of finite monoids. Here, we use the notation  $\Sigma^*/\mathbf{FinMon}$  for the slice category whose objects  $(M, h)$  are the pairs consisting of a finite monoid  $M$  and of a (not necessarily surjective) homomorphism of the form  $h : \Sigma^* \rightarrow M$ , and whose morphisms  $(M, h) \rightarrow (M', h')$  are the homomorphisms  $f : M \rightarrow M'$  making the triangle

$$\begin{array}{ccc} & \Sigma^* & \\ h \swarrow & & \searrow h' \\ M & \xrightarrow{f} & M' \end{array}$$

commute. The projection functor  $\pi$  in (4) transports  $(M, h)$  to the underlying finite monoid  $M$ . One obtains in this way  $\widehat{\Sigma^*}$  as the limit of a codirected diagram of finite monoid homomorphisms

$$\left( M \longrightarrow M' \right)_{(M,h) \rightarrow (M',h')} \quad (5)$$

which extends the diagram (3) from finite index congruences  $\phi$  on  $\Sigma^*$  to all homomorphisms  $h : \Sigma^* \rightarrow M$  to a finite monoid  $M$ .

To explain the relationship with automata, recall that every homomorphism  $h : \Sigma^* \rightarrow M$  to a finite monoid  $(M, \cdot_M, e_M)$  induces a deterministic finite automaton, by letting  $Q := M$  be the set of states, and defining  $\delta(a, q) := q \cdot_M h(a)$  for every letter  $a \in \Sigma$  and state  $q \in M$ . This establishes that every monoid homomorphism  $h : \Sigma^* \rightarrow M$  to a finite monoid  $M = \{q_1, \dots, q_m\}$  induces a decomposition of  $\Sigma^*$  into  $m$  components  $L_{q_i} = h^{-1}(q_i)$  for  $1 \leq i \leq m$  where each  $L_{q_i}$  is a regular language. We will denote by  $\mathbf{Reg}_{(M,h)}\langle \Sigma \rangle$  the Boolean algebra of languages generated by the regular languages of the form  $L_q = h^{-1}(q)$ , as  $q$  ranges over the elements of  $M$ . One obtains in this way a functor

$$\mathbf{Reg}_{(-)}\langle \Sigma \rangle \quad : \quad (\Sigma^*/\mathbf{FinMon})^{\text{op}} \longrightarrow \mathbf{BA} \quad (6)$$

to the category  $\mathbf{BA}$  of Boolean algebras, which maps every pair  $(M, h)$  to the Boolean algebra  $\mathbf{Reg}_{(M,h)}\langle \Sigma \rangle$ . Note that this Boolean algebra  $\mathbf{Reg}_{(M,h)}\langle \Sigma \rangle$  coincides with the image of the Boolean algebra homomorphism  $h^{-1} : \wp(M) \rightarrow \wp(\Sigma^*)$  obtained by applying the contravariant powerset functor  $\wp$  to the map  $h : \Sigma^* \rightarrow M$ . An important insight of [10, Sec. 4.2] is that the following directed diagram in  $\mathbf{BA}$ , associated to the functor (6),

$$\left( \mathbf{Reg}_{(M',h')}\langle \Sigma \rangle \longleftarrow \mathbf{Reg}_{(M,h)}\langle \Sigma \rangle \right)_{(M,h) \rightarrow (M',h')}$$

may be obtained more directly by applying  $\wp$  to the codirected diagram of finite sets underlying (4) and (5). Since the colimit in  $\mathbf{BA}$  of this diagram coincides with  $\mathbf{Reg}\langle \Sigma \rangle$ , one establishes in this way that the monoid of profinite words is in fact the Stone dual of the Boolean algebra  $\mathbf{Reg}\langle \Sigma \rangle$  of regular sets, see [10] as well as §2 below for details.

#### *From profinite words to profinite $\lambda$ -terms*

In order to define the notion of *profinite  $\lambda$ -term* at an arbitrary simple type  $A$ , we combine this general scheme with ideas coming from Reynolds parametricity. We have seen that, given a finite set  $Q$ , we can interpret any simple type  $A$  as a finite set  $\llbracket A \rrbracket_Q$ . To relate elements belonging to two different

interpretations  $\llbracket A \rrbracket_Q$  and  $\llbracket A \rrbracket_{Q'}$  one can construct, given a relation  $R \subseteq Q \times Q'$  between the finite sets used for the interpretation, a relation  $\llbracket A \rrbracket_R \subseteq \llbracket A \rrbracket_Q \times \llbracket A \rrbracket_{Q'}$  between the two interpretations of the simple type  $A$ . Such inductively-defined relations are called *logical relations*. A fundamental fact is that  $\lambda$ -terms are parametric, that is, for any  $\lambda$ -term  $M$  of type  $A$  and any relation  $R \subseteq Q \times Q'$ ,

$$(\llbracket M \rrbracket_Q, \llbracket M \rrbracket_{Q'}) \in \llbracket A \rrbracket_R .$$

In particular, we will recall in Proposition 2.3 the well-known fact that every *partial surjection*  $f : Q \rightarrow Q'$ , seen as a relation, induces a *partial surjection*  $\llbracket A \rrbracket_f : \llbracket A \rrbracket_Q \rightarrow \llbracket A \rrbracket_{Q'}$  such that for every  $\lambda$ -term  $M$  of type  $A$ , its interpretation  $\llbracket M \rrbracket_Q$  is in the domain of  $\llbracket A \rrbracket_f$  and the *partial surjection*  $\llbracket A \rrbracket_f$  sends the interpretation of  $M$  in the finite set  $\llbracket A \rrbracket_Q$  to its interpretation in  $\llbracket A \rrbracket_{Q'}$ , that is,

$$\llbracket A \rrbracket_f(\llbracket M \rrbracket_Q) = \llbracket M \rrbracket_{Q'} . \quad (7)$$

An easy argument, given in Lemma 2.4 below, shows that, as a consequence, every *partial surjection*  $f : Q \rightarrow Q'$  induces an inclusion of Boolean algebras  $\text{Reg}_{Q'}\langle A \rangle \subseteq \text{Reg}_Q\langle A \rangle$ . We note **FinPSurj** the category whose objects are finite sets and whose morphisms are *partial surjections*. For every simple type  $A$ , we then have a functor

$$\text{Reg}_{(-)}\langle A \rangle : \mathbf{FinPSurj}^{\text{op}} \rightarrow \mathbf{BA}$$

which sends each *partial surjection* on the associated inclusion of Boolean algebras. This leads us to the first main result of the paper, established in §2.

**Theorem A.** *The diagram of Boolean algebras  $\text{Reg}_{(-)}\langle A \rangle : \mathbf{FinPSurj}^{\text{op}} \rightarrow \mathbf{BA}$ , i.e.*

$$\left( \text{Reg}_{Q'}\langle A \rangle \longleftarrow \text{Reg}_Q\langle A \rangle \right)_{f: Q \rightarrow Q' \in \mathbf{FinPSurj}} ,$$

is directed, and its colimit in **BA** coincides with the Boolean algebra  $\text{Reg}\langle A \rangle$  of regular languages of higher-order type  $A$ .

At this stage, a key observation coming from Stone duality is that, for each finite set  $Q$ , the finite Boolean algebra  $\text{Reg}_Q\langle A \rangle$  is join-generated by its finite set of atoms, which, as we will show in Proposition 3.1 below, is in bijection with the set

$$\llbracket A \rrbracket_Q^\bullet = \left\{ \llbracket M \rrbracket_Q \mid M \in \Lambda_{\beta\eta}\langle A \rangle \right\} \subseteq \llbracket A \rrbracket_Q$$

of *definable elements* in  $\llbracket A \rrbracket_Q$ . Moreover, using (7), we see that, for every *partial surjection*  $f : Q \rightarrow Q'$ , there exists a unique (total) surjection  $\llbracket A \rrbracket_f^\bullet : \llbracket A \rrbracket_Q^\bullet \rightarrow \llbracket A \rrbracket_{Q'}^\bullet$  making the following diagram commute

$$\begin{array}{ccc} \llbracket A \rrbracket_Q^\bullet & \xrightarrow{\llbracket A \rrbracket_f^\bullet} & \llbracket A \rrbracket_{Q'}^\bullet \\ \downarrow & & \downarrow \\ \llbracket A \rrbracket_Q & \xrightarrow{\llbracket A \rrbracket_f} & \llbracket A \rrbracket_{Q'} \end{array} ,$$

in the category **FinPSet** of finite sets and partial functions. We are now ready to define the set  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  of *profinite  $\lambda$ -terms* of type  $A$  as the limit in the category **Set** of the codirected diagram of finite sets

$$\left( \llbracket A \rrbracket_f^\bullet : \llbracket A \rrbracket_Q^\bullet \longrightarrow \llbracket A \rrbracket_{Q'}^\bullet \right)_{f: Q \rightarrow Q' \in \mathbf{FinPSurj}}$$

indexed by *partial surjections* between finite sets. This diagram is dual to the directed diagram in **BA** defining the Boolean algebra  $\text{Reg}\langle A \rangle$  in Theorem A. Moreover, by Stone duality, the set  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  of *profinite  $\lambda$ -terms* of type  $A$  is not just a set, but a Stone space, dual to the Boolean algebra  $\text{Reg}\langle A \rangle$ .

The conceptual definition of **profinite  $\lambda$ -term** which we have just given is nice but probably a little bit abstract to a reader with expertise in the  $\lambda$ -calculus but not necessarily in Stone duality. A more pedestrian way to understand it is to think of a **profinite  $\lambda$ -term**  $\theta \in \widehat{\Lambda}_{\beta\eta}\langle A \rangle$  of type  $A$  as a family of **definable elements**  $\theta_Q \in \llbracket A \rrbracket_Q^\bullet$  indexed by finite sets  $Q$ , such that the family  $\theta$  is moreover *natural* with respect to finite **partial surjections**, in the expected sense that the equality  $\llbracket A \rrbracket_f^\bullet(\theta_Q) = \theta_{Q'}$  holds for every **partial surjection**  $f : Q \twoheadrightarrow Q'$  between finite sets.

#### *Profinite $\lambda$ -terms and Reynolds parametricity*

The pedestrian definition of **profinite  $\lambda$ -terms** just given requires that the family of definable elements  $\theta_Q \in \llbracket A \rrbracket_Q^\bullet$  is natural with respect to finite **partial surjections**, instead of asking the stronger property that the family  $\theta$  is *parametric* in the traditional sense of Reynolds. We establish in §4 the important property that every **profinite  $\lambda$ -term** may be equivalently defined using parametricity instead of partial surjections, as follows:

**Theorem B.** *A profinite  $\lambda$ -term  $\theta \in \widehat{\Lambda}_{\beta\eta}\langle A \rangle$  of type  $A$  may be equivalently defined as a family of definable elements  $\theta_Q \in \llbracket A \rrbracket_Q^\bullet$  indexed by finite sets  $Q$ , such that the family  $\theta$  is moreover parametric with respect to any **logical relation**, in the sense that  $(\theta_Q, \theta_{Q'}) \in \llbracket A \rrbracket_R$  for every relation  $R \subseteq Q \times Q'$ .*

As we will see in §4, the fact that the notion based on parametricity is stronger than the notion based on naturality is easy to show. What is more difficult to establish is that the two notions are in fact equivalent.

#### *The cartesian closed category **ProLam** of profinite lambda-terms*

We establish that the resulting notion of **profinite  $\lambda$ -term** is compositional by constructing a cartesian closed category **ProLam** of **profinite  $\lambda$ -terms**. There is a functor

$$\text{idonobj} \quad : \quad \mathbf{Lam} \longrightarrow \mathbf{ProLam}$$

which is faithful by Statman's theorem and which embeds the simply typed  $\lambda$ -terms into **profinite  $\lambda$ -terms**. It associates to a simply typed  $\lambda$ -term  $M$  the **profinite  $\lambda$ -term** whose component at the finite set  $Q$  is the interpretation  $\llbracket M \rrbracket_Q$ .

Another interesting fact is that there exists, for every simple type  $A$ , a **profinite  $\lambda$ -term** defining a fixpoint operator

$$\Omega_A \in \widehat{\Lambda}_{\beta\eta}\langle (A \Rightarrow A) \Rightarrow (A \Rightarrow A) \rangle$$

which thus defines a morphism

$$\Omega_A \quad : \quad (A \Rightarrow A) \longrightarrow (A \Rightarrow A)$$

in the category **ProLam** of **profinite  $\lambda$ -terms**. The fixpoint operator  $\Omega_A$  is similar in spirit but different in practice from the usual fixpoint operators  $Y_A : (A \Rightarrow A) \Rightarrow A$  of Scott domain semantics, and one interesting direction for future work will be to understand how the two fixpoint operators  $\Omega_A$  and  $Y_A$  are related.

We also establish at the end of the paper (see §7) that **profinite  $\lambda$ -terms** of type  $\text{Church}_\Sigma$  are the same thing as profinite words over the alphabet  $\Sigma$  in the traditional sense.

**Theorem C.** *For every finite set  $\Sigma$ , there is a homeomorphism between the space of profinite  $\lambda$ -terms of type  $\text{Church}_\Sigma$  and the space of profinite words over  $\Sigma$ , that is,*

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

#### *Related works*

As explained in the introduction, our present definition of **profinite  $\lambda$ -term** relies on the notion of *regular language* of simply typed  $\lambda$ -terms introduced by Salvati [27]. Interestingly, the notion of regular language is formulated by Salvati in two different but equivalent ways. The first definition of regular language is based on the interpretation of  $\lambda$ -terms in the finite standard model **FinSet** of the simply typed  $\lambda$ -calculus. This

is the definition which we recall and develop in the introduction and in the paper. The second equivalent definition given by Salvati relies on the construction of an intersection type system in direct correspondence with the finite monotone model of the simply typed  $\lambda$ -calculus constructed in the category **FinScott** of finite lattices and monotone maps between them, see [28] for a discussion. Aware of this correspondence with Scott semantics, Salvati and Walukiewicz actively promoted a semantic approach to higher-order model checking [29] which would complement the intersection type approach developed by Kobayashi and Ong [18, 19]. However, besides the fascinating connections to Krivine environment machines and collapsible pushdown automata [6, 15, 30], it took several years to develop a precise connection between Scott semantics and intersection type systems for higher-order model checking, with the emergence of a notion of higher-order parity automaton [20] founded on the discovery of an unexpected relationship with linear logic [7, 8, 13, 14] combined with a comonadic translation designed by Melliès of the simply typed  $\lambda Y$ -calculus into a  $\lambda Y_{\mu\nu}$ -calculus with inductive and coinductive fixpoints [20], or into a  $\lambda Y$ -calculus with priorities [32].

One fundamental idea which emerged from these works, also apparent in the work by Colcombet and Petrişan [9], is that there exists a correspondence between the specific category used for the semantic interpretation and a specific class of automata of interest. Typically, the interpretation of the simply typed  $\lambda$ -calculus in **FinSet** corresponds to the class of deterministic automata, while the interpretation in **FinScott** corresponds to the class of non-deterministic automata. In the present paper, we focus on the finite standard model in **FinSet**, and leave the investigation of the finite monotone lattice model in **Scott** for future works.

Another important line of work at the interface of automata theory and  $\lambda$ -calculus was initiated by Hillebrand and Kanellakis [16] with a purely syntactic description of regular languages of finite words using the Church encoding in the simply typed  $\lambda$ -calculus. This alternative approach is extremely promising and has seen a recent revival with the works by Nguyèn and Pradic on implicit automata theory [21, 22]. Our definition of **profinite  $\lambda$ -term** is formulated using the finite standard model, but it is largely independent of it, and it would thus be interesting to recast our definition of **profinite  $\lambda$ -term** in this purely syntactic framework.

In the study of regular languages and profinite monoids, the potential role of Stone duality was identified early on by Pippenger [24], and can also already be recognized in the “implicit operations” which were introduced by Reiterman [25] and play a role in Almeida’s important work on profinite semigroups [3]. It is also interesting to note in this context that monoidal relations, under the name of “relational morphisms”, have long played an important role in (pro)finite semigroup theory, as exemplified for example by Rhodes and Steinberg [26], and our crucial use of logical relations in this paper opens up potential new connections with that theory.

The specific methodology of understanding profinite algebraic structure by applying Stone duality to a lattice of regular languages that we closely follow in Sections 2 and 3 of this paper emerged from an influential series of works by Gehrke, Grigorieff and Pin [11, 12], culminating in Gehrke’s [10], which contains the most general account to date of that line of research. In a direction that is related to, but different from, the one pursued in this paper, Bojańczyk [5] generalized these profinite ideas to the category of algebras given by an arbitrary monad, also see the more recent work by Adámek et al. [1] pursuing a similar direction. While these works were always based in an algebraic setting, a novel contribution of this paper is to show how these ideas extend to the setting of the simply typed  $\lambda$ -calculus and cartesian closed categories.

### *Overview of the paper*

We start by recalling in §2 the notion of regular language of  $\lambda$ -terms induced by the finite standard model of the simply typed  $\lambda$ -calculus. Then, as explained in the introduction, we establish in §2 that the Boolean algebra  $\text{Reg}\langle A \rangle$  of regular languages of simply typed  $\lambda$ -terms of type  $A$  formulated by Salvati can be equivalently expressed (**Theorem A**) as a colimit in **BA** of a specific directed diagram of finite Boolean algebras  $\text{Reg}_Q\langle A \rangle$ . This leads us to introduce in §3 the set  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  of **profinite  $\lambda$ -terms** of type  $A$ , which we define as the limit in **Set** of a specific codirected diagram of finite sets  $\llbracket A \rrbracket_Q^\bullet \subseteq \llbracket A \rrbracket_Q$ . We also show that, by construction, the set of **profinite  $\lambda$ -terms** can be equipped with a natural topology which turns  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  into the Stone space dual to the Boolean algebra  $\text{Reg}_Q\langle A \rangle$ . We establish in the next section §4 that **profinite**

$\lambda$ -terms can be defined in an alternative and more direct way as families of **definable elements**  $\theta_Q \in \llbracket A \rrbracket_Q^\bullet$  satisfying a parametricity property with respect to any binary relation  $R \subseteq Q \times Q'$ . This is the essence of **Theorem B** mentioned in the introduction. We then show in §5 that the resulting notion of **profinite  $\lambda$ -term** is compositional in the technical sense that it defines a cartesian closed category **ProLam** whose objects are the simply types and whose morphisms are **profinite  $\lambda$ -terms**. Using Statman's theorem, we establish in §6 that the canonical functor from the category **Lam** of simply typed  $\lambda$ -terms to the category **ProLam** of **profinite  $\lambda$ -terms** is a faithful embedding. Finally, we establish in §7 our theorem (**Theorem C**) that given a finite alphabet  $\Sigma$  of letters, the notion of **profinite  $\lambda$ -terms** of type  $\text{Church}_\Sigma$  coincides with the usual notion of profinite words over  $\Sigma$ . We conclude and give a number of perspectives for future work in §8.

## 2 Regular languages of $\lambda$ -terms

In this section, we define the collection  $\text{Reg}\langle A \rangle$  of **regular languages** at an arbitrary type  $A$ , and establish **Theorem A** of the introduction, showing how  $\text{Reg}\langle A \rangle$  can be built as the colimit of a directed diagram in the category of Boolean algebras.

**Definition 2.1** Let  $Q$  be a finite set and  $A$  a type. We say that a subset  $L \subseteq \Lambda_{\beta\eta}\langle A \rangle$  is a **regular language** of type  $A$  recognized by  $Q$  if there exists a subset  $\text{Acc}$  of  $\llbracket A \rrbracket_Q$  such that  $L = \llbracket - \rrbracket_Q^{-1}(\text{Acc})$ , that is,

$$\text{for any } M \in \Lambda_{\beta\eta}\langle A \rangle, \quad M \in L \iff \llbracket M \rrbracket_Q \in \text{Acc} .$$

We denote the Boolean algebra of **regular languages** of type  $A$  recognized by  $Q$  by  $\text{Reg}_Q\langle A \rangle \subseteq \wp(\Lambda_{\beta\eta}\langle A \rangle)$  and we write

$$\text{Reg}\langle A \rangle = \bigcup \{ \text{Reg}_Q\langle A \rangle \mid Q \text{ a finite set} \}$$

for the collection of **regular languages** of type  $A$ .

While it is clear that, for each individual finite set  $Q$ , the set  $\text{Reg}_Q\langle A \rangle$  is closed under the Boolean operations, since it is defined as the image of the Boolean homomorphism  $\llbracket - \rrbracket_Q^{-1}$ , it is not immediately apparent that the union  $\text{Reg}\langle A \rangle$  is also closed under the Boolean operations.

To this end, we will use **logical relations**. If  $S \subseteq P \times P'$  and  $R \subseteq Q \times Q'$  are two set-theoretic relations between finite sets, then one can define their exponential, which is  $S \Rightarrow R \subseteq (P \Rightarrow Q) \times (P' \Rightarrow Q')$ , as

$$S \Rightarrow R \quad := \quad \{ (g, h) \mid \text{for all } x \ R \ y, \text{ we have } g(x) \ R' \ h(y) \} .$$

Therefore, for any relation  $R \subseteq Q \times Q'$  between two finite sets  $Q$  and  $Q'$ , we construct the relation  $\llbracket A \rrbracket_R \subseteq \llbracket A \rrbracket_Q \times \llbracket A \rrbracket_{Q'}$  by induction on the simple type  $A$  as

$$\llbracket \circ \rrbracket_R := R \quad \text{and} \quad \llbracket A \Rightarrow B \rrbracket_R := \llbracket A \rrbracket_R \Rightarrow \llbracket B \rrbracket_R .$$

The **fundamental lemma of logical relations** then states that for all  $M \in \Lambda_{\beta\eta}\langle A \rangle$  and any  $R \subseteq Q \times Q'$ , the interpretations of  $M$  at  $Q$  and  $Q'$  are related in the sense that

$$\llbracket M \rrbracket_Q \llbracket A \rrbracket_R \llbracket M \rrbracket_{Q'} .$$

In particular, we will make extensive use of **partial surjections**, i.e. relations which are graphs of surjective partial functions. We note  $f : Q \twoheadrightarrow Q'$  such a **partial surjection**. We first prove the following lemma, which states that **partial surjections** are stable by exponential.

**Lemma 2.2** *If  $e : P \twoheadrightarrow P'$  and  $f : Q \twoheadrightarrow Q'$  are **partial surjections**, then so is the relation  $e \Rightarrow f$ .*

**Proof.** We first remark that a **partial surjection**  $f : Q \twoheadrightarrow Q'$  may equivalently be described as a span

$$Q \xleftarrow{\pi_1} f \xrightarrow{\pi_2} Q'$$



where  $\pi_1$  is injective and  $\pi_2$  is surjective. We adopt this viewpoint during this proof.

Let  $e : P \rightrightarrows P'$  and  $f : Q \rightrightarrows Q'$  be two **partial surjections**. Note that two functions  $g : P \rightarrow Q$  and  $h : P' \rightarrow Q'$  are related by  $e \Rightarrow f$  if and only if there exists a function  $c : e \rightarrow f$  such that the following diagram commutes:

$$\begin{array}{ccccc} P & \xleftarrow{\pi_1} & e & \xrightarrow{\pi_2} & P' \\ g \downarrow & & \downarrow c & & \downarrow h \\ Q & \xleftarrow{\pi'_1} & f & \xrightarrow{\pi'_2} & Q' \end{array} . \quad (8)$$

First, we show that the relation  $e \Rightarrow f$  is a partial function. Let  $g : P \rightarrow Q$  and  $h_1, h_2 : P' \rightarrow Q'$  such that  $(g, h_i) \in e \Rightarrow f$  for  $i = 1, 2$ . We thus have two maps  $c_i : e \rightarrow f$  for  $i = 1, 2$  such that

$$g \circ \pi_1 = \pi'_1 \circ c_i \quad \text{and} \quad h_i \circ \pi_2 = \pi'_2 \circ c_i \quad \text{for } i = 1, 2 .$$

The maps all fit in the diagram

$$\begin{array}{ccccc} P & \xleftarrow{\pi_1} & e & \xrightarrow{\pi_2} & P' \\ g \downarrow & & c_1 \downarrow & c_2 \downarrow & h_1 \downarrow & h_2 \downarrow \\ Q & \xleftarrow{\pi'_1} & f & \xrightarrow{\pi'_2} & Q' \end{array} .$$

By injectivity of  $\pi'_1$ , we get that  $c_1 = c_2$ . Therefore,

$$h_1 \circ \pi_2 = \pi'_2 \circ c_1 = \pi'_2 \circ c_2 = h_2 \circ \pi_2 .$$

By surjectivity of  $\pi_2$ , we get that  $h_1 = h_2$ . This proves that the relation  $e \Rightarrow f$  is a partial function.

We now show that the relation  $e \Rightarrow f$  is surjective. Let  $h : P' \rightarrow Q'$  be any function. As  $\pi'_2$  is surjective, it has a section  $s : Q' \rightarrow f$ , that is,  $\pi'_2 \circ s = \text{id}_{Q'}$ . We then define the function  $c : e \rightarrow f$  as  $s \circ h \circ \pi_2$ . Note that  $\pi'_2 \circ c = h \circ \pi_2$ , so we obtain the commuting diagram

$$\begin{array}{ccccc} P & \xleftarrow{\pi_1} & e & \xrightarrow{\pi_2} & P' \\ & & \downarrow c & & \downarrow h \\ Q & \xleftarrow{\pi'_1} & f & \xrightarrow[\leftarrow s]{\pi'_2} & Q' \end{array} .$$

As  $\pi_1$  is injective, it has a retraction  $r : P \rightarrow e$ , that is,  $r \circ \pi_1 = \text{id}_e$ . We define the function  $g : Q \rightarrow Q'$  as  $\pi'_1 \circ c \circ r$  and note that

$$g \circ \pi_1 = \pi'_1 \circ c .$$

Thus, the diagram (8) commutes for this choice of  $g$ ,  $c$ , and  $h$ , which means that  $(g, h)$  is in the relation  $e \Rightarrow f$ , as required. This shows that  $e \Rightarrow f$  is surjective.  $\square$

Using Proposition 2.2, we get a proof of the following proposition by induction on simple types.

**Proposition 2.3** *If  $f : Q \rightrightarrows Q'$  is a **partial surjection**, then the relation  $\llbracket A \rrbracket_f \subseteq \llbracket A \rrbracket_Q \times \llbracket A \rrbracket_{Q'}$  is a **partial surjection** for any simple type  $A$ .*

The following lemma contains the crucial argument needed to prove the fact that  $\text{Reg}\langle A \rangle$  is a Boolean algebra.

**Lemma 2.4** *Let  $f : Q \rightrightarrows Q'$  be a **partial surjection**. Then, for any simple type  $A$ , we have an inclusion of Boolean algebras  $\text{Reg}_{Q'}\langle A \rangle \subseteq \text{Reg}_Q\langle A \rangle$ .*

**Proof.** Let  $A$  be any simple type and let  $\text{Acc}' \subseteq \llbracket A \rrbracket_{Q'}$ , recognizing the language  $L = \llbracket - \rrbracket_{Q'}^{-1}(\text{Acc}')$  in  $\text{Reg}_{Q'}\langle A \rangle$ . We define the subset  $\text{Acc}$  of  $\llbracket A \rrbracket_Q$  as  $\llbracket A \rrbracket_f^{-1}(\text{Acc}')$ , that is  $\{x \in \llbracket A \rrbracket_Q \mid x \llbracket A \rrbracket_f y \text{ for some } y \in \text{Acc}'\}$ . By the **fundamental lemma of logical relations**, for any term  $M$  of simple type  $A$ , we have  $\llbracket M \rrbracket_Q \in$

Acc if and only if  $\llbracket M \rrbracket_{Q'} \in \text{Acc}'$  as  $\llbracket A \rrbracket_f$  is a partial function. We conclude that  $L$  is equal to  $\llbracket - \rrbracket_Q^{-1}(\text{Acc})$ , so that  $L$  is also recognized by  $Q$ .  $\square$

In order to prove that the union  $\text{Reg}\langle A \rangle$  of the Boolean algebras  $\text{Reg}_Q\langle A \rangle$  is again a Boolean algebra, we will apply the following general principle from universal algebra in the case where  $\mathbf{V}$  is the variety of Boolean algebras, see for example [2, Rem. 3.4.4(iii) on p. 136].

**Proposition 2.5** *For any finitary variety of algebras  $\mathbf{V}$ , the forgetful functor  $\mathbf{V} \rightarrow \mathbf{Set}$  creates directed colimits.*

We are now ready to prove our first main result, **Theorem A** of the introduction.

**Theorem A.** *The diagram of Boolean algebras  $\text{Reg}_{(-)}\langle A \rangle : \mathbf{FinPSurj}^{\text{op}} \rightarrow \mathbf{BA}$ , i.e.*

$$\left( \text{Reg}_{Q'}\langle A \rangle \longleftarrow \text{Reg}_Q\langle A \rangle \right)_{f:Q \rightarrow Q' \in \mathbf{FinPSurj}} ,$$

is directed, and its colimit in  $\mathbf{BA}$  coincides with the Boolean algebra  $\text{Reg}\langle A \rangle$  of regular languages of higher-order type  $A$ .

**Proof.** We first show that the diagram of inclusions of Boolean algebras is directed. Indeed, for any finite sets  $Q_1$  and  $Q_2$ , we have, for  $i = 1, 2$ , the **partial surjection**  $f_i: Q_1 + Q_2 \rightarrow Q_i$  defined by  $q f_i q'$  if and only if  $q \in Q_i$  and  $q = q'$ . Thus, Lemma 2.4 gives that  $\text{Reg}_{Q_i}\langle A \rangle \subseteq \text{Reg}_{Q_1+Q_2}\langle A \rangle$ . Now, by Proposition 2.5, applied in the case  $\mathbf{V} = \mathbf{BA}$ , the union  $\text{Reg}\langle A \rangle$  of the sets in the diagram is again a Boolean algebra, and it is the colimit of the diagram in  $\mathbf{BA}$ .  $\square$

We end this section by showing explicitly how we recover in this context the result of [27, Thm. 8] that  $\text{Reg}\langle A \rangle$  is closed under binary intersection.

**Proposition 2.6** *For any simple type  $A$ , the set of regular languages  $\text{Reg}\langle A \rangle \subseteq \wp(\Lambda_{\beta\eta}\langle A \rangle)$  is closed under binary intersection.*

**Proof.** Suppose that  $L_1 \in \text{Reg}_{Q_1}\langle A \rangle$  and  $L_2 \in \text{Reg}_{Q_2}\langle A \rangle$ . By the argument given in the proof of **Theorem A**, both  $L_1$  and  $L_2$  are in  $\text{Reg}_{Q_1+Q_2}\langle A \rangle$  which is a Boolean algebra, so their intersection  $L_1 \cap L_2$  is also in  $\text{Reg}_{Q_1+Q_2}\langle A \rangle$ .  $\square$

### 3 The space of profinite $\lambda$ -terms

The aim of this section is to define **profinite  $\lambda$ -terms** of an arbitrary simple type  $A$  as special **parametric families of semantic elements**, and to show that they form a Stone space dual to the Boolean algebra  $\text{Reg}\langle A \rangle$ .

Throughout this section, we fix a simple type  $A$ . We saw in the previous section that  $\text{Reg}\langle A \rangle$  is a Boolean algebra which is the colimit of a directed diagram of inclusions between the Boolean algebras  $\text{Reg}_Q\langle A \rangle$ . As  $\text{Reg}_Q\langle A \rangle$  is finite for every  $Q$ , it is isomorphic to  $\wp(X_Q(A))$ , where  $X_Q(A)$  is the set of atoms of  $\text{Reg}_Q\langle A \rangle$ . Applying discrete Stone duality to the directed diagram of inclusions of finite Boolean algebras, we thus obtain a codirected diagram of maps  $X_Q(A) \rightarrow X_{Q'}(A)$ , still indexed by **partial surjections**  $Q \rightarrow Q'$ . We now first give a more concrete description of that diagram.

**Proposition 3.1** *For every finite set  $Q$ , the set of atoms  $X_Q(A)$  of  $\text{Reg}_Q\langle A \rangle$  is in a bijection with the set  $\llbracket A \rrbracket_Q^\bullet$  of definable elements of simple type  $A$ , given by the function*

$$\begin{aligned} \llbracket A \rrbracket_Q^\bullet &\longrightarrow X_Q(A) \\ q &\longmapsto \llbracket - \rrbracket_Q^{-1}(\{q\}) . \end{aligned}$$

**Proof.** The Boolean algebra  $\text{Reg}_Q\langle A \rangle$  is, by definition, the image of the Boolean algebra homomorphism  $\llbracket - \rrbracket_Q^{-1} : \wp(\llbracket A \rrbracket_Q) \rightarrow \wp(\Lambda_{\beta\eta}\langle A \rangle)$ . Thus,  $\text{Reg}_Q\langle A \rangle$  arises as the following epi-mono factorization of Boolean algebras

$$\begin{array}{ccc}
 \wp(\llbracket A \rrbracket_Q) & \xrightarrow{\llbracket - \rrbracket_Q^{-1}} & \wp(\Lambda_{\beta\eta}\langle A \rangle) \\
 & \searrow \llbracket - \rrbracket_Q^{-1} & \swarrow p \\
 & \text{Reg}_Q\langle A \rangle & 
 \end{array}$$

Applying the discrete duality functor  $\mathbf{At}: \mathbf{CABA} \rightarrow \mathbf{Set}$  to this diagram, we get the dual epi-mono factorization of sets

$$\begin{array}{ccc}
 \llbracket A \rrbracket_Q & \xleftarrow{\llbracket - \rrbracket_Q} & \Lambda_{\beta\eta}\langle A \rangle \\
 & \swarrow & \nwarrow \\
 & X_Q(A) & 
 \end{array}$$

Since  $\llbracket A \rrbracket_Q^\bullet$  is by definition the image of  $\llbracket - \rrbracket_Q$  in  $\llbracket A \rrbracket_Q$ , the result follows by the uniqueness up to isomorphism of epi-mono factorizations in  $\mathbf{CABA}$ .  $\square$

We now show that **logical relations** induced by **partial surjections**, when restricted to **definable elements**, all yield the same total function.

**Proposition 3.2** *For any partial surjection  $f: Q \rightrightarrows Q'$  and for any simple type  $A$ , the set  $\llbracket A \rrbracket_Q^\bullet \subseteq \llbracket A \rrbracket_Q$  of definable elements is contained in the domain of  $\llbracket A \rrbracket_f$ , and the restriction of  $\llbracket A \rrbracket_f$  to  $\llbracket A \rrbracket_Q^\bullet$  is the unique function  $p_{Q,Q'}$  that makes the following diagram commute:*

$$\begin{array}{ccc}
 & \Lambda_{\beta\eta}\langle A \rangle & \\
 \llbracket - \rrbracket_Q^\bullet \swarrow & & \searrow \llbracket - \rrbracket_{Q'}^\bullet \\
 \llbracket A \rrbracket_Q^\bullet & \xrightarrow{p_{Q,Q'}} & \llbracket A \rrbracket_{Q'}^\bullet
 \end{array}$$

**Proof.** By the **fundamental lemma of logical relations**, for any term  $M$  of simple type  $A$ , we have  $\llbracket M \rrbracket_Q \llbracket A \rrbracket_f \llbracket M \rrbracket_{Q'}$ , so that any definable element is in the domain of  $\llbracket A \rrbracket_f$ . By Proposition 2.3,  $\llbracket A \rrbracket_f$  is in particular a partial function, so that it makes the diagram commute. For the uniqueness, simply note that any  $q \in \llbracket A \rrbracket_Q^\bullet$  can by definition be written as  $\llbracket M \rrbracket_Q$  for some term  $M$  of simple type  $A$ , and must therefore be sent to  $\llbracket M \rrbracket_{Q'}$  by any function making the diagram commute.  $\square$

Note that Proposition 3.2 in particular implies that, if  $f, g: Q \rightrightarrows Q'$  are two **partial surjections**, then, while their semantic interpretations  $\llbracket A \rrbracket_f, \llbracket A \rrbracket_g: \llbracket A \rrbracket_Q \rightrightarrows \llbracket A \rrbracket_{Q'}$  are in general distinct, their restrictions to the set  $\llbracket A \rrbracket_Q^\bullet$  of definable elements must both be equal to the function  $p_{Q,Q'}$ .

We are now ready to define **profinite  $\lambda$ -terms** of a given simple type  $A$ .

**Definition 3.3** Let  $A$  be any simple type. We define the set of **profinite  $\lambda$ -terms** as the limit  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  in  $\mathbf{Set}$  of the diagram

$$\left( \llbracket A \rrbracket_f^\bullet : \llbracket A \rrbracket_Q^\bullet \longrightarrow \llbracket A \rrbracket_{Q'}^\bullet \right)_{f: Q \rightrightarrows Q' \in \mathbf{FinPSurj}} .$$

It follows from the way that one calculates limits in  $\mathbf{Set}$  that, concretely, a **profinite  $\lambda$ -term** in  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  is a family  $\theta$  of definable elements  $\theta_Q \in \llbracket A \rrbracket_Q^\bullet$  where  $Q$  ranges over all finite sets such that

$$\text{for every partial surjection } f: Q \rightarrow Q', \quad \text{we have } \llbracket A \rrbracket_f^\bullet(\theta_Q) = \theta_{Q'} . \quad (9)$$

By Proposition 3.2, the condition (9) on the family  $\theta$  is equivalent to the condition that, for any term  $M$  of simple type  $A$  and any finite sets  $Q$  and  $Q'$ ,

$$\text{if } \theta_Q = \llbracket M \rrbracket_Q \text{ and } |Q| \geq |Q'| \text{ then } \theta_{Q'} = \llbracket M \rrbracket_{Q'} . \quad (10)$$

We conclude this section by equipping the set  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  with a natural topology, and showing that this topology turns  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  into the Stone dual space of the Boolean algebra  $\text{Reg}\langle A \rangle$ . The easiest way to define the topology of  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  is to say that it is the subspace topology inherited from the inclusion

$$\widehat{\Lambda}_{\beta\eta}\langle A \rangle \hookrightarrow \prod_Q \llbracket A \rrbracket_Q^\bullet$$

into the product space  $\prod_Q \llbracket A \rrbracket_Q^\bullet$  computed in the category **Top** of topological spaces, where each component  $\llbracket A \rrbracket_Q^\bullet$  is considered as a topological space equipped with the discrete topology. More concretely, for any finite set  $Q$  and  $q \in \llbracket A \rrbracket_Q^\bullet$ , let us write  $U_{Q,q}$  for the set of **profinite  $\lambda$ -terms** that take value  $q$  at  $Q$ , that is,

$$U_{Q,q} := \left\{ \theta \in \widehat{\Lambda}_{\beta\eta}\langle A \rangle \mid \theta_Q = q \right\}.$$

The topology on  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  is now defined by taking the collection of sets  $U_{Q,q}$  as a basis, where  $Q$  ranges over all finite sets and  $q$  ranges over all the elements of  $Q$ . The following result is proved via an argument similar to the one given in [10, Sec. 4.2] for profinite algebras.

**Proposition 3.4** *The space  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$  is the Stone dual space of the Boolean algebra  $\text{Reg}\langle A \rangle$ . In particular,  $\text{Reg}\langle A \rangle$  is isomorphic to the Boolean algebra of clopen sets of  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$ .*

**Proof.** Stone duality arises from the dual equivalence between **FinSet** and **FinBA** by taking the projective and inductive completions, respectively. Therefore, we have in particular that  $\widehat{\Lambda}_{\beta\eta}\langle A \rangle$ , which is defined as the codirected limit of the diagram of finite discrete spaces  $\llbracket A \rrbracket_Q^\bullet$  in **Top**, is the dual space of the directed colimit of the diagram of finite Boolean algebras  $\text{Reg}_Q\langle A \rangle$ , which is the Boolean algebra  $\text{Reg}\langle A \rangle$  by **Theorem A**. The second statement now follows because any Boolean algebra is isomorphic to the collection of clopen sets of its dual space.  $\square$

## 4 Profinite $\lambda$ -terms and parametricity

Let  $A$  be any simple type. A *parametric family* is a family of points  $\theta_Q \in \llbracket A \rrbracket_Q$ , where  $Q$  ranges over all finite sets, such that for any relation  $R \subseteq Q \times Q'$ , we have  $\theta_Q \llbracket A \rrbracket_R \theta_{Q'}$ .

Every **parametric family**  $\theta$  is in particular parametric with respect to **partial surjections**. Therefore, a **parametric family** whose components are **definable elements** is a **profinite  $\lambda$ -term**. We now show that the converse holds.

**Theorem B.** *A profinite  $\lambda$ -term  $\theta \in \widehat{\Lambda}_{\beta\eta}\langle A \rangle$  of simple type  $A$  may be equivalently defined as a **parametric family of definable elements**  $\theta_Q \in \llbracket A \rrbracket_Q^\bullet$ .*

**Proof.** Let  $\theta$  be a **profinite  $\lambda$ -term**, viewed as a family of definable elements which is parametric with respect to every partial surjection, or equivalently, satisfying condition (10). Let  $Q_1$  and  $Q_2$  be any two finite sets and let  $R \subseteq Q_1 \times Q_2$  be any relation. Pick any finite set  $Q$  of cardinality  $\max(|Q_1|, |Q_2|)$ . Since  $\theta_Q$  is in particular definable, pick a  $\lambda$ -term  $M$  in  $\Lambda_{\beta\eta}\langle A \rangle$  such that  $\theta_Q$  is  $\llbracket M \rrbracket_Q$ . Since  $|Q| \geq |Q_i|$  for  $i = 1, 2$ , by (10) we now also have  $\theta_{Q_i}$  is equal to  $\llbracket M \rrbracket_{Q_i}$ . By the **fundamental lemma of logical relations**, we obtain that  $\llbracket M \rrbracket_{Q_1} \llbracket A \rrbracket_R \llbracket M \rrbracket_{Q_2}$  which proves that  $\theta$  is a **parametric family**.  $\square$

## 5 The cartesian closed category of profinite $\lambda$ -terms

We now show that **profinite  $\lambda$ -terms** assemble into a cartesian closed category **ProLam** which thus provides an interpretation of the simply typed  $\lambda$ -calculus. In order to construct the category **ProLam**, we find it convenient to use a general construction introduced by Jacq and Mellès [17] in a more general monoidal and 2-categorical setting. Suppose given a cartesian closed category **C** and a functor

$$\mathcal{P} : \mathbf{C} \longrightarrow \mathbf{Set}$$

which is *cartesian product preserving* in the sense that the canonical functions

$$\begin{aligned} \langle \mathcal{P}(\pi_1), \mathcal{P}(\pi_2) \rangle & : \mathcal{P}(A \times B) \longrightarrow \mathcal{P}(A) \times \mathcal{P}(B) \\ !_{\mathcal{P}(1)} & : \mathcal{P}(1) \longrightarrow 1 \end{aligned}$$

are bijections for all objects  $A$  and  $B$  of the category  $\mathbf{C}$ . We denote by

$$\begin{aligned} m_{A,B} &: \mathcal{P}(A) \times \mathcal{P}(B) \longrightarrow \mathcal{P}(A \times B) \\ m_1 &: 1 \longrightarrow \mathcal{P}(1) \end{aligned}$$

the inverse functions. In that situation, one defines the category  $\mathbf{C}[\mathcal{P}]$  whose objects are the objects of  $\mathbf{C}$  and whose hom-sets are defined as follows:

$$\mathbf{C}[\mathcal{P}](A, B) := \mathcal{P}(A \Rightarrow B)$$

using the internal hom-object  $A \Rightarrow B$  of the cartesian closed category  $\mathbf{C}$ . Equivalently,  $\mathbf{C}[\mathcal{P}]$  is the result of seeing the cartesian closed category  $\mathbf{C}$  as enriched in itself, and then changing the base along  $\mathcal{P}$ . One establishes that

**Proposition 5.1** *The category  $\mathbf{C}[\mathcal{P}]$  is cartesian closed and comes equipped with a cartesian closed identity-on-object functor*

$$\text{idonobj}_{\mathbf{C},\mathcal{P}} : \mathbf{C} \longrightarrow \mathbf{C}[\mathcal{P}]$$

which strictly preserves the cartesian product as well as the internal hom.

Now, in order to obtain the category **ProLam** of **profinite  $\lambda$ -terms** using this categorical construction, we start by recalling the definition of the cartesian closed category **Lam** freely generated by the terminal category.

**Definition 5.2** The category **Lam** has as objects the simple types of the  $\lambda$ -calculus and its hom-sets are defined as

$$\mathbf{Lam}(A, B) := \Lambda_{\beta\eta}\langle A \Rightarrow B \rangle$$

for all pairs  $A$  and  $B$  of simple types.

At this stage, we are ready to consider the functor  $\mathcal{P} : \mathbf{Lam} \longrightarrow \mathbf{Set}$  which transports every simple type  $A$  to the set of **profinite  $\lambda$ -terms**

$$\mathcal{P}(A) = \widehat{\Lambda}_{\beta\eta}\langle A \rangle \tag{11}$$

and every  $\lambda$ -term  $M$  of simple type  $A \Rightarrow B$  to the set-theoretic function sending a **profinite  $\lambda$ -term**  $\theta$  of simple type  $A$  on the family  $(\llbracket M \rrbracket_Q(\theta_Q))$  which can be shown to be a **profinite  $\lambda$ -term** of simple type  $B$  using the **fundamental lemma of logical relations**. It is interesting to observe that the functor  $\mathcal{P}$  is **cartesian product preserving** and that we have canonical bijections

$$\mathcal{P}(A \times B) \cong \mathcal{P}(A) \times \mathcal{P}(B) \quad \mathcal{P}(1) \cong 1$$

for every pair of simple types  $A$  and  $B$ . By applying the construction, we obtain a cartesian closed category

$$\mathbf{ProLam} := \mathbf{Lam}[\mathcal{P}]$$

whose objects are the simple types of the  $\lambda$ -calculus and whose hom-sets are defined as follows:

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

**Remark 5.3** Note that the functors  $\mathcal{P}$  are chosen to be valued in **Set**, but we could choose any cartesian category  $\mathbf{S}$  as long as  $\mathcal{P}$  still is **cartesian product preserving** relatively to the cartesian structure of  $\mathbf{S}$ . The construction will then yield a cartesian closed category  $\mathbf{C}[\mathcal{P}]$  enriched over  $\mathbf{S}$ . As a matter of fact, the functor  $\mathcal{P} : \mathbf{Lam} \rightarrow \mathbf{Set}$  used in (11) to construct  $\mathbf{ProLam} = \mathbf{Lam}[\mathcal{P}]$  happens to factor through the category **Stone** of Stone spaces, in the following way:

$$\mathbf{Lam} \xrightarrow{\widehat{\Lambda}_{\beta\eta}\langle - \rangle} \mathbf{Stone} \xrightarrow{\text{forget}} \mathbf{Set}$$

This shows that the cartesian closed category **ProLam** of **profinite  $\lambda$ -terms** may be also considered as enriched over the category **Stone** of Stone spaces.

## 6 A faithful embedding from $\lambda$ -terms to profinite $\lambda$ -terms

By construction, the category **ProLam** comes equipped with a cartesian closed identity-on-object functor

$$\text{idonobj} \quad : \quad \mathbf{Lam} \longrightarrow \mathbf{ProLam} \quad (12)$$

which may also be derived from the fact that **Lam** is the free cartesian closed category. We now establish that

**Proposition 6.1** *The functor  $\text{idonobj}$  is faithful.*

Towards proving Proposition 6.1, we first claim that the category **ProLam** can be obtained as the limit of a codirected diagram of cartesian closed categories, described in the following way. Given a finite set  $Q$  and a simple type  $A$ , consider the equivalence relation

$$\sim_Q^A \quad \subseteq \quad \Lambda_{\beta\eta}\langle A \rangle \times \Lambda_{\beta\eta}\langle A \rangle$$

on the set of simply typed  $\lambda$ -terms of type  $A$  modulo  $\beta\eta$ -conversion, defined as:

$$M \sim_Q^A N \quad \stackrel{\text{def}}{\iff} \quad \llbracket M \rrbracket_Q = \llbracket N \rrbracket_Q .$$

When we fix the finite set  $Q$ , the family  $\sim_Q$  of equivalence relations  $\sim_Q^A$  parametrized by simple types  $A$  defines a congruence relation on the category **Lam**, in the expected sense that

$$\text{if } f \sim_Q^{A \Rightarrow B} f' \text{ and } g \sim_Q^{B \Rightarrow C} g', \text{ then } g \circ f \sim_Q^{A \Rightarrow C} g' \circ f'$$

for any tuple of morphisms  $f, f', g, g'$  of the form:

$$A \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{f'} \end{array} B \begin{array}{c} \xrightarrow{g} \\ \xrightarrow{g'} \end{array} C \quad .$$

From this, it follows that we can define the category

$$\mathbf{Lam}_Q := \mathbf{Lam} / \sim_Q$$

obtained by considering the morphisms of the free cartesian closed category **Lam** modulo the congruence relation  $\sim_Q$  in the expected sense that

$$\mathbf{Lam}_Q(A, B) = \Lambda_{\beta\eta}\langle A \Rightarrow B \rangle / \sim_Q^{A \Rightarrow B} . \quad (13)$$

We then establish that

**Proposition 6.2** *For every finite set  $Q$ , the category  $\mathbf{Lam}_Q$  is cartesian closed and comes equipped with a cartesian closed identity-on-object functor*

$$\pi_Q \quad : \quad \mathbf{Lam} \longrightarrow \mathbf{Lam}_Q .$$

Moreover, every partial surjection  $f : Q \rightarrow Q'$  in the category **FinPSurj** induces a cartesian closed identity-on-object functor

$$\mathbf{Lam}_f \quad : \quad \mathbf{Lam}_Q \longrightarrow \mathbf{Lam}_{Q'}$$

making the diagram of cartesian closed functors commute:

$$\begin{array}{ccc} & \xrightarrow{\pi_Q} & \mathbf{Lam}_Q \\ \mathbf{Lam} & \searrow & \downarrow \mathbf{Lam}_f \\ & \xrightarrow{\pi_{Q'}} & \mathbf{Lam}_{Q'} \end{array}$$

From this observation, it is not too difficult to show that

**Proposition 6.3** *The category **ProLam** is the codirected limit of the diagram of cartesian closed categories **Lam**<sub>Q</sub> indexed by finite sets and partial surjections. The projection functor*

$$\pi_Q \quad : \quad \mathbf{ProLam} \longrightarrow \mathbf{Lam}_Q$$

is defined by transporting every morphism  $\theta : A \rightarrow B$  defined as a family  $\theta$  of definable elements in

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

to the instance  $\theta_Q$  in

$$\mathbf{Lam}_Q(A, B) = \Lambda_{\beta\eta}\langle A \Rightarrow B \rangle / \sim_Q^{A \Rightarrow B} \cong \llbracket A \Rightarrow B \rrbracket_Q^\bullet.$$

One also establishes that the canonical functor (12) is also characterized by the fact that it is the unique cartesian closed functor making the diagram below commute:

$$\begin{array}{ccccc}
 & & \pi_Q & \xrightarrow{\quad} & \mathbf{Lam}_Q \\
 & \searrow & & \nearrow \pi_Q & \downarrow \mathbf{Lam}_f \\
 \mathbf{Lam} & \xrightarrow{\text{idonobj}} & \mathbf{ProLam} & & \\
 & \swarrow & \searrow \pi_{Q'} & & \downarrow \\
 & & \pi_{Q'} & \xrightarrow{\quad} & \mathbf{Lam}_{Q'}
 \end{array}$$

This observation provides us with a clean proof that the functor  $\text{idonobj}$  is faithful. Indeed, by Statman's finite completeness theorem [31], for every pair of morphisms  $f, g : A \Rightarrow B$  in the category **Lam** which is (by definition) a pair of  $\lambda$ -terms  $M$  and  $N$  modulo  $\beta\eta$ -conversion, either  $M$  and  $N$  are equal modulo  $\beta\eta$ -conversion or there exists a finite set  $Q$  such that the interpretations  $\llbracket M \rrbracket_Q = \pi_Q(M)$  and  $\llbracket N \rrbracket_Q = \pi_Q(N)$  are different. In particular,  $\text{idonobj}(f)$  and  $\text{idonobj}(g)$  differ in the second case. This establishes that the canonical functor  $\text{idonobj} : \mathbf{Lam} \rightarrow \mathbf{ProLam}$  is faithful, as claimed in Proposition 6.1.

## 7 Profinite $\lambda$ -terms and profinite words

The higher-order language theory on simply typed  $\lambda$ -terms is designed to extend the traditional language theory on words on a given finite alphabet  $\Sigma$ . The idea is that a finite word on the alphabet  $\Sigma$  is the same thing as a  $\lambda$ -term of simple type  $\text{Church}_\Sigma$  modulo  $\beta\eta$ -conversion. In particular, we recall below a folklore result which states that the Boolean algebra  $\text{Reg}\langle \text{Church}_\Sigma \rangle$  of regular higher-order languages on  $\text{Church}_\Sigma$  coincides with the Boolean algebra  $\text{Reg}\langle \Sigma \rangle$  of regular languages on the finite alphabet  $\Sigma$ .

**Proposition 7.1** *For every finite alphabet  $\Sigma$ , one has an isomorphism of Boolean algebra*

$$\text{Reg}\langle \text{Church}_\Sigma \rangle \quad \cong \quad \text{Reg}\langle \Sigma \rangle$$

given by the Church encoding.

**Proof.** The Church encoding provides a one-to-one correspondence between subsets  $L \subseteq \Sigma^*$  of words over the alphabet  $\Sigma$  and subsets  $\mathcal{L} \subseteq \Lambda_{\beta\eta}\langle \text{Church}_\Sigma \rangle$  of  $\lambda$ -terms of simple type  $\text{Church}_\Sigma$  closed modulo  $\beta\eta$ -conversion. We show that a subset  $L \subseteq \Sigma^*$  is regular if and only if the associated subset  $\mathcal{L} \subseteq \Lambda_{\beta\eta}\langle \text{Church}_\Sigma \rangle$  is an element of  $\text{Reg}\langle \text{Church}_\Sigma \rangle$ .

In one direction, suppose that  $L \subseteq \Sigma^*$  is a language of words recognized by a DFA  $\mathcal{A} = (Q, \delta, q_0, \text{Acc})$ . We recall from the introduction that the associated set  $\mathcal{L} \subseteq \Lambda_{\beta\eta}\langle \text{Church}_\Sigma \rangle$  of  $\lambda$ -terms is the inverse image by the semantic bracket

$$\llbracket - \rrbracket_Q \quad : \quad \Lambda_{\beta\eta}\langle \text{Church}_\Sigma \rangle \longrightarrow \llbracket \text{Church}_\Sigma \rrbracket_Q$$

of the set of functionals in  $\llbracket \text{Church}_\Sigma \rrbracket_Q$  defined as follows

$$\text{eval}_{(\delta, q_0)}^{-1}(\text{Acc}) = \{F \in \llbracket \text{Church}_\Sigma \rrbracket_Q \mid F(\delta_{a_1}, \dots, \delta_{a_n})(q_0) \in \text{Acc}\}.$$

By definition,  $\mathcal{L} \subseteq \Lambda_{\beta\eta}\langle \text{Church}_\Sigma \rangle$  is thus an element of  $\text{Reg}_Q\langle \text{Church}_\Sigma \rangle$  and thus an element of  $\text{Reg}\langle \text{Church}_\Sigma \rangle$ .

Conversely, by definition of  $\text{Reg}\langle \text{Church}_\Sigma \rangle$ , it is sufficient to establish, for every finite set  $Q$ , that every subset  $\mathcal{L} \in \text{Reg}_Q\langle \text{Church}_\Sigma \rangle$  has its corresponding subset  $L \subseteq \Sigma^*$  a regular language. By definition of  $\text{Reg}_Q\langle \text{Church}_\Sigma \rangle$ , we may suppose without loss of generality that  $\mathcal{L} \in \text{Reg}_Q\langle \text{Church}_\Sigma \rangle$  is of the form

$$\mathcal{L} = \llbracket - \rrbracket_Q^{-1}(\{F\}) = \{M \in \Lambda_{\beta\eta}\langle \text{Church}_\Sigma \rangle \mid \llbracket M \rrbracket_Q = F\}$$

where  $F$  is a functional in  $\llbracket \text{Church}_\Sigma \rrbracket_Q$ . The corresponding set  $L \subseteq \Sigma^*$  is the finite intersection of all the regular languages recognized by the DFAs of the form  $\mathcal{A} = (Q, \delta, q_0, \{q_f\})$  where the unique final state  $q_f$  is equal to  $q_f = F(\delta_{a_1}, \dots, \delta_{a_n})(q_0)$ . As a finite intersection of regular languages, the set  $L \subseteq \Sigma^*$  is itself regular.  $\square$

We use this result in order to establish our [Theorem C](#).

**Theorem C.** *For every finite alphabet  $\Sigma$ , there is a homeomorphism*

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

between the space  $\widehat{\Lambda}_{\beta\eta}\langle \text{Church}_\Sigma \rangle$  of *profinite  $\lambda$ -terms* of type  $\text{Church}_\Sigma$  and the space  $\widehat{\Sigma}^*$  of *profinite words*.

**Proof.** By Proposition 3.4, the space  $\widehat{\Lambda}_{\beta\eta}\langle \text{Church}_\Sigma \rangle$  is the Stone dual of the Boolean algebra  $\text{Reg}\langle \text{Church}_\Sigma \rangle$  which is isomorphic to the Boolean algebra  $\text{Reg}\langle \Sigma \rangle$  by Proposition 7.1. From this follows that the space  $\widehat{\Lambda}_{\beta\eta}\langle \text{Church}_\Sigma \rangle$  is homeomorphic to the Stone dual of  $\text{Reg}\langle \Sigma \rangle$  which coincides with the space  $\widehat{\Sigma}^*$  of profinite words by an important result of Stone duality, see [23].  $\square$

One main benefit of extending finite words into profinite words is that a new class of implicit operations become available [4]. In particular, there exists an *idempotent power operator*  $u \mapsto u^\omega$  which turns every profinite word  $u$  into another profinite word noted  $u^\omega$ , and defines a continuous function

$$u \longmapsto u^\omega \quad : \quad \widehat{\Sigma}^* \longrightarrow \widehat{\Sigma}^* \tag{14}$$

see for example [23, Prop. 2.5]. The construction is based on the observation that for any element  $x$  of a finite monoid  $M$ , there exists a unique power  $x^n$  of  $x$ , for  $n \geq 1$ , which is idempotent. This unique power is obtained when  $n$  is the factorial of the cardinality of  $M$ , and is also usually written  $x^\omega$ . The continuous function (14) is obtained by taking the profinite limit of this operation on monoids. We show the construction generalizes from profinite words to *profinite  $\lambda$ -terms* at every type  $A$ .

**Proposition 7.2** *For every simple type  $A$ , there exists a *profinite  $\lambda$ -term**

$$\Omega_A \quad : \quad (A \Rightarrow A) \Rightarrow A \Rightarrow A \tag{15}$$

which, given any  $M \in \widehat{\Lambda}_{\beta\eta}\langle A \Rightarrow A \rangle$ , satisfies the idempotency equation

$$(\Omega_A M) \circ (\Omega_A M) = \Omega_A M$$

between *profinite  $\lambda$ -terms*, where  $g \circ f$  is notation for  $\lambda x.f(gx)$  where  $f$  and  $g$  are *profinite  $\lambda$ -terms*.

We have seen in [Theorem C](#) that one recovers the traditional notion of profinite words on a finite alphabet  $\Sigma$  by considering the *profinite  $\lambda$ -terms* of type  $\text{Church}_\Sigma$ . Accordingly, the continuous operation (14)



can be recovered as the **profinite  $\lambda$ -term**

$$\lambda u. \lambda f_1 \dots f_n. \Omega_{\circ}(u f_1 \dots f_n) \quad : \quad \text{Church}_{\Sigma} \Rightarrow \text{Church}_{\Sigma}$$

where  $\Omega_{\circ} : (\circ \Rightarrow \circ) \Rightarrow (\circ \Rightarrow \circ)$  denotes the idempotent power operator  $\Omega_A$  at type  $A = \circ$ . Note that we use the compositional calculus provided by the cartesian closed category **ProLam** in order to see the expression  $\lambda u. \lambda f_1 \dots f_n. \Omega_{\circ}(u f_1 \dots f_n)$  as a **profinite  $\lambda$ -term**.

## 8 Conclusion

In this paper, we introduce the notion of *profinite  $\lambda$ -term* of a given simple type which we define in a clean and principled way by establishing in **Theorem A** and **Theorem B** that the definitions based on duality theory and on parametricity coincide. We also establish in **Theorem C** that the Church encoding of finite words as  $\lambda$ -terms extends to profinite words, in the sense that the usual notion of profinite word on a finite alphabet  $\Sigma$  coincides with the notion of **profinite  $\lambda$ -term** on the type  $\text{Church}_{\Sigma}$  encoding the alphabet  $\Sigma$ . We also construct a cartesian closed category **ProLam** of **profinite  $\lambda$ -terms**, and construct a cartesian closed functor

$$\text{idonobj} \quad : \quad \mathbf{Lam} \longrightarrow \mathbf{ProLam}$$

from the cartesian closed category of usual simply typed  $\lambda$ -terms. We also show that this embedding functor from simply typed  $\lambda$ -terms to **profinite  $\lambda$ -terms** is faithful, using Statman's theorem. The construction shows that simply typed  $\lambda$ -terms can be considered as particular **profinite  $\lambda$ -terms**, and that **profinite  $\lambda$ -terms** can be manipulated in the same compositional way as usual simply typed  $\lambda$ -terms.

## References

- [1] Adámek, J., L.-T. Chen, S. Milius and H. Urbat, *Reiterman's theorem on finite algebras for a monad*, ACM Transactions on Computational Logic (TOCL) **22**, pages 1–48 (2021).
- [2] Adámek, J. and J. Rosický, *Locally Presentable and Accessible Categories*, London Mathematical Society Lecture Note Series, Cambridge University Press (1994).  
<https://doi.org/10.1017/CB09780511600579>
- [3] Almeida, J., *Profinite semigroups and applications*, in: V. B. Kudryavtsev, I. G. Rosenberg and M. Goldstein, editors, *Structural Theory of Automata, Semigroups, and Universal Algebra*, pages 1–45, Springer Netherlands, Dordrecht (2005). Notes taken by Alfredo Costa.
- [4] Almeida, J., A. Costa, R. Kyriakoglou and D. Perrin, *Profinite Semigroups and Symbolic Dynamics* (2020), ISBN 978-3-030-55214-5.  
<https://doi.org/10.1007/978-3-030-55215-2>
- [5] Bojanczyk, M., *Recognisable languages over monads (full version)*, CoRR **abs/1502.04898** (2015). 1502.04898.  
<http://arxiv.org/abs/1502.04898>
- [6] Broadbent, C. H., A. Carayol, C. L. Ong and O. Serre, *Higher-order recursion schemes and collapsible pushdown automata: Logical properties*, ACM Trans. Comput. Log. **22**, pages 12:1–12:37 (2021).  
<https://doi.org/10.1145/3452917>
- [7] Bucciarelli, A. and T. Ehrhard, *On phase semantics and denotational semantics in multiplicative-additive linear logic*, Ann. Pure Appl. Log. **102**, pages 247–282 (2000).  
[https://doi.org/10.1016/S0168-0072\(99\)00040-8](https://doi.org/10.1016/S0168-0072(99)00040-8)
- [8] Bucciarelli, A. and T. Ehrhard, *On phase semantics and denotational semantics: the exponentials*, Ann. Pure Appl. Log. **109**, pages 205–241 (2001).  
[https://doi.org/10.1016/S0168-0072\(00\)00056-7](https://doi.org/10.1016/S0168-0072(00)00056-7)
- [9] Colcombet, T. and D. Petrişan, *Automata minimization: a functorial approach*, Logical Methods in Computer Science **16**, page 32:1–32:28 (2020).  
<https://lmcs.episciences.org/6213/pdf>
- [10] Gehrke, M., *Stone duality, topological algebra, and recognition*, Journal of Pure and Applied Algebra **220**, pages 2711–2747 (2016).

- [11] Gehrke, M., S. Grigorieff and J.-E. Pin, *Duality and equational theory of regular languages*, in: L. Aceto and al., editors, *ICALP 2008, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 246–257, Springer, Berlin (2008).
- [12] Gehrke, M., S. Grigorieff and J.-E. Pin, *A Topological Approach to Recognition*, in: S. A. et al., editor, *Automata, Languages and Programming*, volume 6199 of *Lecture Notes in Computer Science*, pages 151–162, Springer (2010). 37th International Colloquium (ICALP 2010).
- [13] Grellois, C. and P. Melliès, *Finitary semantics of linear logic and higher-order model-checking*, in: G. F. Italiano, G. Pighizzini and D. Sannella, editors, *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part I*, volume 9234 of *Lecture Notes in Computer Science*, pages 256–268, Springer (2015).  
[https://doi.org/10.1007/978-3-662-48057-1\\_20](https://doi.org/10.1007/978-3-662-48057-1_20)
- [14] Grellois, C. and P. Melliès, *Relational semantics of linear logic and higher-order model checking*, in: S. Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, volume 41 of *LIPICs*, pages 260–276, Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015).  
<https://doi.org/10.4230/LIPICs.CSL.2015.260>
- [15] Hague, M., A. S. Murawski, C. L. Ong and O. Serre, *Collapsible pushdown automata and recursion schemes*, *ACM Trans. Comput. Log.* **18**, pages 25:1–25:42 (2017).  
<https://doi.org/10.1145/3091122>
- [16] Hillebrand, G. and P. Kanellakis, *On the expressive power of simply typed and let-polymorphic lambda calculi*, in: *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*, pages 253–263 (1996).  
<https://doi.org/10.1109/LICS.1996.561337>
- [17] Jacq, C. and P.-A. Melliès, *Categorical combinatorics for non deterministic strategies on simple games*, in: C. Baier and U. Dal Lago, editors, *Foundations of Software Science and Computation Structures*, pages 39–70, Springer International Publishing, Cham (2018), ISBN 978-3-319-89366-2.
- [18] Kobayashi, N., *Types and higher-order recursion schemes for verification of higher-order programs*, *SIGPLAN Not.* **44**, page 416–428 (2009), ISSN 0362-1340.  
<https://doi.org/10.1145/1594834.1480933>
- [19] Kobayashi, N. and C. L. Ong, *Complexity of model checking recursion schemes for fragments of the modal mu-calculus*, *Log. Methods Comput. Sci.* **7** (2011).  
[https://doi.org/10.2168/LMCS-7\(4:9\)2011](https://doi.org/10.2168/LMCS-7(4:9)2011)
- [20] Melliès, P.-A., *Higher-order parity automata*, in: *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '17*, IEEE Press (2017), ISBN 9781509030187.
- [21] Nguyễn, L. T. D., C. Noûs and C. Pradic, *Implicit automata in typed  $\lambda$ -calculi II: streaming transducers vs categorical semantics*, *CoRR abs/2008.01050* (2020). 2008.01050.  
<https://arxiv.org/abs/2008.01050>
- [22] Nguyễn, L. T. D. and C. Pradic, *Implicit automata in typed  $\lambda$ -calculi I: aperiodicity in a non-commutative logic*, in: A. Czumaj, A. Dawar and E. Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPICs*, pages 135:1–135:20, Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020).  
<https://doi.org/10.4230/LIPICs.ICALP.2020.135>
- [23] Pin, J.-E., *Profinite Methods in Automata Theory*, in: S. Albers and J.-Y. Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science*, volume 3 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31–50, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2009), ISBN 978-3-939897-09-5, ISSN 1868-8969.  
<https://doi.org/10.4230/LIPICs.STACS.2009.1856>
- [24] Pippenger, N., *Regular languages and Stone duality*, *Theory of Computing Systems* **30**, pages 121–134 (1997).  
<https://doi.org/10.1007/BF02679444>
- [25] Reiterman, J., *The Birkhoff theorem for finite algebras*, *Algebra Universalis* **14**, pages 1–10 (1982).
- [26] Rhodes, J. and B. Steinberg, *The q-theory of Finite Semigroups*, Springer Publishing Company, 1st edition (2008), ISBN 9780387097800.
- [27] Salvati, S., *Recognizability in the Simply Typed Lambda-Calculus*, in: *16th Workshop on Logic, Language, Information and Computation*, volume 5514 of *Lecture Notes in Computer Science*, pages 48–60, Springer, Tokyo Japan (2009).
- [28] Salvati, S., *Lambda-calculus and formal language theory*, *Habilitation à diriger des recherches*, Université de Bordeaux (2015).  
<https://hal.science/tel-01253426>

- [29] Salvati, S. and I. Walukiewicz, *Krivine machines and higher-order schemes*, in: L. Aceto, M. Henzinger and J. Sgall, editors, *Automata, Languages and Programming*, pages 162–173, Springer Berlin Heidelberg, Berlin, Heidelberg (2011), ISBN 978-3-642-22012-8.
- [30] Salvati, S. and I. Walukiewicz, *Krivine machines and higher-order schemes*, *Inf. Comput.* **239**, pages 340–355 (2014).  
<https://doi.org/10.1016/j.ic.2014.07.012>
- [31] Statman, R., *Completeness, invariance and lambda-definability*, *J. Symb. Log.* **47**, pages 17–26 (1982).  
<https://doi.org/10.2307/2273377>
- [32] Walukiewicz, I., *Lambda  $\eta$ -calculus with priorities*, in: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13, IEEE (2019).  
<https://doi.org/10.1109/LICS.2019.8785674>