From profinite words to profinite λ-terms

Vincent Moreau, IRIF, Université Paris Cité

This is joint work with Sam van Gool and Paul-André Melliès.

The aim of this work is to combine methods from profinite algebra and models of the λ-calculus to obtain a notion of *profinite λ-term*. Profinite algebraic structures have been used in automata theory for giving classifications of regular languages, see e.g. [1] for a survey. In particular, elements of the free profinite monoid are known as *profinite words*. They provide a way to speak about limiting behavior of finite words with respect to deterministic automata. In order to connect this notion of profiniteness to the λ-calculus, we generalize from usual automata models to the computation model of *higher-order automata* [2, 3]. Indeed, higher-order automata generalize automata of words and trees, in that they can process arbitrary simply-typed λ-terms as their input. Existing notions of automata on words and trees are then obtained as a special case through Church encoding.

Our main contribution here is the study of the notion of parametric λ-term, parametric in the sense of Reynolds [4] and relatively to any model of the simply-typed λ-calculus, and the establishment of a link with profinite words. We first consider a model constructed from finite sets and functions, which corresponds to deterministic automata. We prove that in that setting, the notion of parametric λ-term essentially coincides with the classical notion of profinite word. However, when we move for example from functions to relations, which corresponds to the move from deterministic to non-deterministic automata, we will obtain a new notion of 'non-deterministically profinite' λ-term. The abstract point of view that we introduce here will thus allow us, in future work, to investigate parametric λ-terms for other models, obtaining a new landscape of generalizations of profinite words.

In the remainder of this abstract, we will describe our work in more technical detail.

Let Σ be a finite alphabet. A *profinite word* over Σ is a family $u = (u_p)$, where p ranges over the surjective monoid homomorphisms $p\colon \Sigma^* \twoheadrightarrow M$, with M a finite monoid, such that, for every p, u_p is an element of the range of p, and for any monoid homomorphism $f : M \to N$, with N a finite monoid,

$$u_{f \circ p} = f(u_p).$$

Every finite word $w \in \Sigma^*$ yields a profinite word (w_p) by defining w_p to be $p(w)$, for each $p\colon \Sigma^* \twoheadrightarrow M$. However, there exist many profinite words that do not come from finite words: for example, for any letter $a \in \Sigma$, denote by u_p is the unique idempotent power of $p(a)$ in the finite monoid M, for each $p\colon \Sigma^* \twoheadrightarrow M$. Then (u_p) is a profinite word, which is usually denoted by "a^ω" and does not arise from a finite word.

The first step towards the definition of parametric λ-terms is the following alternative characterization of profinite words. Let us define a *monoidal relation* from M to N to be a submonoid of $M \times N$. It can be shown that, for any $p : \Sigma^* \twoheadrightarrow M$, $q : \Sigma^* \twoheadrightarrow N$, R a monoidal relation from M to N, and u a profinite word,

$$\text{if, for all } w \in \Sigma^*,\ p(w)\ R\ q(w),\ \text{then } u_p\ R\ u_q.$$

Conversely, any family (u_p) which verifies this condition is a profinite word.

The second step is to see how words may be encoded within the simply-typed λ-calculus. Recall that the untypes λ-terms are those generated by the grammar

$$M, N ::= x \mid \lambda x.M \mid MN.$$

A simple type (with one type variable $\mathfrak{o}$) is a term generated by the grammar

$$A, B ::= \mathfrak{o} \mid A \Rightarrow B.$$

Let Γ be a context, namely a set of pairs $x : A$ consisting of a variable and of a simple type. A λ-term M has type A in context Γ iff the judgment $\Gamma \vdash M : A$ can be derived in the system sith the three following typing rules

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash M : B \Rightarrow A \qquad \Gamma \vdash N : B}{\Gamma \vdash MN : A} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \Rightarrow B}$$

When a λ-term M is closed, i.e. when every occurring variable is bound by a λ-abstraction, we say that M has type A if $\varnothing \vdash M : A$ is derivable.

As an example, we consider the two letter alphabet $\{a, b\}$. For any word $w = a_1 \dots a_n \in \{a, b\}^*$, one can derive the typing judgment in the simply-typed λ-calculus

$$a : \mathfrak{o} \Rightarrow \mathfrak{o}, b : \mathfrak{o} \Rightarrow \mathfrak{o}, c : \mathfrak{o} \vdash a_1(\dots(a_n c)) : \mathfrak{o}.$$

It follows that the closed term $\lambda a.\lambda b.\lambda c.a_1(\dots(a_n c))$ has type

$$(\mathfrak{o} \Rightarrow \mathfrak{o}) \Rightarrow (\mathfrak{o} \Rightarrow \mathfrak{o}) \Rightarrow (\mathfrak{o} \Rightarrow \mathfrak{o}).$$

In general, any word $w \in \Sigma^*$ can be encoded in this way as a closed term having type

$$\mathbf{Church}_\Sigma := \underbrace{(\mathfrak{o} \Rightarrow \mathfrak{o}) \Rightarrow \dots \Rightarrow (\mathfrak{o} \Rightarrow \mathfrak{o})}_{|\Sigma| \text{ times}} \Rightarrow (\mathfrak{o} \Rightarrow \mathfrak{o}).$$

The third step is to consider logical relations in the sense of Reynolds [4]. As the category **FinSet** of finite sets and functions is cartesian closed, we can interpret

the simply-typed λ-calculus in it. Given a set P and defining $[\![o]\!]_P := P$, structural induction on simple types yields a set $[\![A]\!]_P$ for every simple type A. We extend this interpretation of types by defining, for every binary relation R between P and Q, the binary relation $[\![A]\!]_R$ between $[\![A]\!]_P$ and $[\![A]\!]_Q$ in the following way: we let $[\![o]\!]_R$ be R itself and define, for any simple types A and B, the set $[\![A \Rightarrow B]\!]_R$ to be

$$\{(f,g) \in [\![A \Rightarrow B]\!]_P \times [\![A \Rightarrow B]\!]_Q \mid \forall (p,q) \in [\![A]\!]_R,\ (f(p), g(q)) \in [\![B]\!]_R\}.$$

We now state the central definition of this work that we investigate in the case of finite models.

Definition. A *parametric λ-term of type A* is a family of elements (θ_Q), where Q ranges over all finite sets, such that $\theta_Q \in [\![A]\!]_Q$, and for any binary relation R between finite sets P and Q, we have $\theta_P \ [\![A]\!]_R \ \theta_Q$.

We note that we have only given the definition for the category of finite sets and relations, but this same definition applies in any cartesian closed double category $\mathbf{D}$.

When A is taken to be $\mathbf{Church}_\Sigma$, θ_Q can be seen as a function from $[\![(o \Rightarrow o)^{|\Sigma|}]\!]_Q$ to $[\![o \Rightarrow o]\!]_Q$, and the parametricity condition then becomes, for all families $(p_a)_{a\in\Sigma} : P \to P$ and $(q_a)_{a\in\Sigma} : Q \to Q$

$$\text{if, for all } a \in \Sigma,\ (p_a, q_a) \in [\![o \Rightarrow o]\!]_R, \text{ then } (\theta_P, \theta_Q) \in [\![o \Rightarrow o]\!]_R.$$

Now, any profinite word induces a parametric λ-term in **FinSet** by choosing the monoid M to be $Q \to Q$ with the composition. On the other hand, any parametric λ-term θ in **FinSet** induces a profinite word defined as

$$\begin{array}{ccc} [\Sigma^*, M] & \longrightarrow & M \\ p & \longmapsto & \theta_{(i_M \circ p)}(e_m) \end{array}$$

where i_M is the Cayley injection of M into $M \to M$. These two functions indeed yield profinite words and parametric λ-terms. Moreover, we prove the following, by making crucial use of the notion of parametricity with respect to relations.

Theorem. *The functions between profinite words and parametric λ-term in* **FinSet** *are mutually inverse.*

References

[1] Jean-Eric Pin. "Profinite Methods in Automata Theory". In: STACS 2009.

[2] Sylvain Salvati. "Recognizability in the Simply Typed Lambda-Calculus". In: *Logic, Language, Information and Computation*. Springer Berlin Heidelberg, 2009, pp. 48–60.

[3] Paul-André Melliès. "Higher-order parity automata". In: LICS 2017.

[4] John C. Reynolds. "Types, Abstraction and Parametric Polymorphism". In: IFIP Congress 1983.