Research Statement of Domenico Ruoppolo

research team Logique, Calcul et Raisonnement
Laboratoire d'Informatique de Paris Nord
Université Paris 13 - Sorbonne Paris Cité

April 16, 2017
General context

My research fits within the mathematical study of the ideas of computation and program. In particular, my work is situated at the interface between logic and computer science provided by the proofs-as-programs correspondence, also known as Curry-Howard isomorphism [How80]. Such correspondence is the realization that the many variants of Church’s λ-calculus overlap with a great variety of deductive systems issued from proof theory, the branch of mathematical logic giving a prominent role to the dynamics of rewriting of proofs [Glt89, SU06].

The untyped λ-calculus. Despite not being my only interest, most of my focus so far has been on the study of the untyped λ-calculus [Bar84], introduced by Alonzo Church in the early 30’s within an attempt to give mathematics a logical foundation [Chl32]. It is given by the terms M, N ::= x | λx.M | MN (where x ranges over countable variables) and the rewriting rule (λx.M)N →β M{N/x}. The system provides a model of computation, as it can represent all Turing-computable functions through the notion of λ-definability (Church’s thesis). Also, this system is the common core of all functional programming languages. Over the last five decades λ-calculi have played a prominent role in the conception, implementation and analysis of such languages, but also in a number of impressive theoretical insights into the concepts of computation, program and proof.

Equivalences of programs. The study of the untyped λ-calculus is not restricted to the sole β-rule. One is more often interested in λ-theories, which are congruences on λ-terms extending β-conversion. All λ-theories form a complete lattice of cardinality 2^{ℵ₀}, mostly still unexplored. From the point of view of computer science, observational equivalences have a certain relevance among λ-theories. Indeed, they provide an answer to a nontrivial question: when two programs are equivalent? The answer is behavioural: they are equivalent if they look to behave in the same way in every possible case of execution. Formally, two λ-terms M and N are observationally equivalent with respect to some fixed set O of observable terms when, for every possible context of evaluation C[−], the λ-term C[M] β-reduces to an observable in O if and only if C[N] β-reduces to an observable in O. The choice of O is not unique. The most studied instance is the one where the observables are λ-terms in head normal form. This λ-theory is denoted by H*. An alternative choice is to take as O the set of λ-terms in β-normal form. This last is called Morris’s observational equivalence, and denoted by H⁺ hereafter.

Denotational semantics. Mostly, my research concerns the denotational semantics of the λ-calculus. Dana Scott discovered the first denotational model [Sco72] in the late 60’s. Since then, a large number of such models, lying in many different categories, have been studied. In most of them λ-terms are interpreted as structure-preserving functions between some order-theoretic, algebraic or topological structures. A limitation of these traditional models is to abstract away from the execution process and overlook quantitative aspects such as the time, space, or energy consumed by a computation. My work fits in a wider research program aiming to overcome these limitations. Such a quantitative approach has its inspiration and technical roots in the semantics of Girard’s Linear Logic [Gir87]. More specifically, most of the results that I achieved so far concern relational semantics, which interprets λ-terms as relations where their inputs are grouped together in multisets. As a result of this usage of multisets, relational models are resource-sensitive, in that they represent explicitly the consumption of input resources during the execution of programs. The first concrete examples of relational models of λ-calculus were built in [BEM07, HNPR06].

Relational graph models

In my thesis [Ruo16] and related publications [MR14, BMPR16, BMR17] I studied a proper subclass of relational models, called relational graph models (rgm’s). On the one hand, the definition of an rgm is the relational analogue of the definition of a graph models à la Plotkin-Scott-Engeler [Plt93, Eng81, Lou83], a well-known kind of continuous model. In particular, rgm’s can be built by free completion and by forcing like the continuous ones. On the other hand, rgm’s can be seen as a resource-sensitive reformulation of filter models [BSD13]. The classical Stone duality between filter models and intersection type systems shows that some interesting classes of domain-based models can be described in logical form. The intuition is that a functional intersection type
$\alpha_1 \land \cdots \land \alpha_n \to \beta$ can be seen as a continuous step function sending the set \{\alpha_1, \ldots, \alpha_n\} to the element $\beta$. Our idea, already present in [dC09, PPRDR15], is that in the absence of idempotency and partial orders the functional type $\alpha_1 \land \cdots \land \alpha_n \to \beta$ can be seen as a relation associating the multiset $[\alpha_1, \ldots, \alpha_n]$ with the element $\beta$. As a consequence, even rgm’s can be presented in logical form. Precisely, as non-idempotent intersection type systems. This logical representation comes in handy when studying the quantitative features of these models.

**Full abstractions.** Every denotational model induces a $\lambda$-theory through the kernel of its interpretation function. In particular, a model is fully abstract when the induced $\lambda$-theory is an observational equivalence. In exploring the $\lambda$-theories induced by rgm’s my coauthors and I paid a particular attention to the full abstraction problem. Until recently, researchers were only able to prove full abstractions for individual models [Hyl76, Wad78, CDZ87], or at best to provide sufficient conditions for models living in some class to be fully abstract [Man09]. A substantial advance was made in [Bre14], where Breuvart was able to provide a characterization of all the models fully abstract for $\mathcal{H}^*$ living in a certain class. My coauthors and I achieved equally general full abstraction theorems for the class of rgm’s. We proved that an rgm is fully abstract for $\mathcal{H}^*$ iff it is extensional (it models $\eta$-conversion) and $\lambda$-König [BMPR16, BMR17]. Intuitively, a model is $\lambda$-König when every recursive tree has an infinite path that is witnessed by some element of the model, in a certain type-theoretical sense. By dualizing the notion of $\lambda$-König rgm we also proved a characterization for the other main observational equivalence: an rgm is fully abstract for $\mathcal{H}^*$ iff it is extensional and hyperimmune [BMR17].

**Other results on Morris’s $\lambda$-theory**

The observational equivalence $\mathcal{H}^*$ is maybe less ubiquitously studied in the literature than $\mathcal{H}^\circ$, but nevertheless important. For instance, its notion of observables is central in Böhm’s Theorem [Böhm68] and similar separabilities [CDR78]. This is why I focused on $\mathcal{H}^+$ during my PhD, and, together with my coauthors, proved some other notable results concerning it.

**Extensional Taylor expansion.** Ehrhard-Regnier’s Taylor expansion is a translation developing every $\lambda$-term as an infinite series of terms living in a resource-sensitive version of the $\lambda$-calculus, known as differential $\lambda$-calculus [ER03, ER08]. In [MR14] we defined the extensional Taylor expansion, a version of this notion taking $\eta$-reduction into account, and proved that it provides another model of $\mathcal{H}^+$. The $\omega$-rule. The $\omega$-rule is a strong form of extensionality defined as follows: for all $M, N$ $(MP = NP$ for all closed $P$) $\Rightarrow M = N$. In [BMPR16] we proved that $\mathcal{H}^+$ satisfies the $\omega$-rule. This solved a long-standing open question [Bar84, §17.4].

**Ongoing investigations**

A number of more or less precise open questions concerning rgm’s remain to investigate.

- Are all $\lambda$-theories in the interval $[\mathcal{H}^+, \mathcal{H}^\circ]$ relational graph theories? If it is not the case, is it possible to provide a characterization of the representable ones?
- Do all extensional rgm’s satisfy the $\omega$-rule?
- What is the extensional collapse [Ehr12] of the class of rgm’s?
- Is there a game semantics reading of rgm’s?

Finally, in a much more abstract perspective, a 2-categorical version of the relational semantics was contemplated in [FGHW08, Hyl10], where the categorical notion of profunctors takes the role of relations. An ambitious longterm aim is to explore the possibility of studying a profunctorial version of rgm’s.

**Operads of control**

A completely different ongoing investigation, in collaboration with Paul-André Melliès, is rooted in the higher-algebraic counterpart of denotational semantics: categorical logic, a branch of category theory interested in the interpretation of proofs [LS86, Mel09].

**The context.** Starting from [Gri90], researches have extended the proofs-as-programs correspondence beyond the limits of purely functional programming on one side, and of purely constructive proof systems on the other. In this perspective, the $\lambda\mu$-calculus introduced in [Par92]
is still a major reference. Its untyped version adds to the syntax of the \( \lambda \)-calculus a GOTO-like mechanism of control, by means of exception raising terms \([\alpha]M\) and exception handler abstractions \(\mu\alpha.M\). From a logical point of view, a simply typed fragment of the \(\lambda\mu\)-calculus is a deduction system for classical logic, that is the logic underlying ordinary mathematics.

The continuation-passing-style translations (CPS for short), traditionally providing implementation procedures for \(\lambda\)-calculi, extend also to the \(\lambda\mu\)-calculus. The idea behind CPS’s is to provide to (the translation of) a program \(M\) a parameter (itself a program, according to the functional paradigm) containing all the information about the environment of execution of \(M\) (the rest of the code surrounding \(M\)). CPS’s also have a logical meaning, which stresses the Curry-Howard isomorphism to the level of implementation mechanisms: they correspond to certain versions of double negation embeddings of classical logic into intuitionistic logic, whose pioneers in the 20’s were Glivenko, Gödel, Kolmogorov. Generalizing to a categorical setting a CPS appeared in [LRS93], Selinger axiomatized the algebraic nature of denotational models of the \(\lambda\mu\)-calculus in [Sel01]. The corresponding structure is named control category. In [MT08] and [Tab08] Melliès and Tabareau investigated what happens when one performs Selinger’s axiomatization in a linear context. Their analysis relies on dialogue categories, which can be considered as the basic categorical notion suitable to study from a resource-sensitive perspective the idea of response, and therefore the semantics of programs and environments interacting by mutual responses. The offspring of their work is a notion of linear control operads.

**Ongoing work.** I am focusing on the reconstruction of the non-linear setting from the aforementioned work of Melliès and Tabareau. Intuitively, the question is: can we find a semantic model for the \(\lambda\mu\)-calculus, or even more generally for any non-linear higher-order calculus with control, by adding some structure on top of a linear control operad? The emerging picture is one in which operads playing the role of linear multi-inputs programs interact with operads representing non-linear programs.
Bibliography


