Semantics and syntactic characterizations of Morris's equivalence*

Giulio Manzonetto and Domenico Ruoppolo

Laboratoire LIPN, CNRS UMR7030 Université Paris-Nord
giulio.manzonetto@lipn.univ-paris13.fr
domenico.ruoppolo@lipn.univ-paris13.fr

Introduction. In λ-calculus, two programs M and N are considered equivalent whenever they are *contextually equivalent* with respect to some fixed set $\mathcal{O}$ of *observables*. This means that we can plug either M or N into any context $C(-)$, that is any program with a *hole*, without noticing any difference in the global behaviour: $C(M)$ reduces to an observable in $\mathcal{O}$ exactly when $C(N)$ does. Two notable examples are $\equiv^{\text{hnf}}$ and Morris's equivalence $\equiv^{\text{nf}}$ obtained by taking as observables the head normal forms and the β-normal forms, respectively.

Working with these definitions is difficult because of the quantification over all possible contexts, so researchers have found alternative characterisations of these program equivalences based on syntactic trees or denotational models. For instance, they proved that two programs are equivalent with respect to $\equiv^{\text{hnf}}$ whenever they have the same Nakajima tree [8] or, equivalently, when their interpretations coincide in Scott's model $\mathcal{D}_\infty$ [9]. Similarly, $\equiv^{\text{nf}}$ is captured by extensional Böhm trees [5] and Coppo, Dezani and Zacchi's filter model $\mathcal{D}_{\text{cdz}}$ [2].

The idea behind Böhm trees, and their extensional versions, is to extract the computational content of a program by representing its output as a possibly infinite tree — the continuity of this representation allows to infer properties of the whole tree by studying its finite approximants. For this reason Böhm-like trees and continuous models relied to them via approximation theorems constituted for over forty years the main tools to reason about the behaviour of a program. A limitation of these methods is that they abstract away from the execution process and overlook quantitative aspects such as the time, space, or energy consumed by a computation.

Contributions. Our work [7] fits in a wider research programme whose aim is to rebuild the traditional theory of program approximations, by replacing it with a mathematical model of resource consumption. The starting point is [4], where Ehrhard and Regnier propose to analyse the behaviour of a program via its *Taylor expansion*, which is a generally infinite series of "resource approximants". Such approximants are terms of a *resource calculus* corresponding to a finitary fragment of the differential λ-calculus [3]. Each resource approximant t of a λ-term M captures a particular choice of the number of times M must call its sub-routines during its execution.

* This work is partly supported by ANR JCJC Project Coquas 12JS0200601.

Both the differential λ-calculus and the Taylor expansion can be naturally interpreted in the relational semantics of linear logic. The first author *et al.* built a relational model $\mathcal{D}_\omega$ living in such a semantics [1] and proved, using standard techniques, that the induced equality is exactly $\equiv^{\mathrm{hnf}}$ [6], just like for Scott's model $\mathcal{D}_\infty$. In our work [7] we provide syntactical and denotational methods based on Taylor expansion that allow to characterise Morris's equivalence $\equiv^{\mathrm{nf}}$.

We introduce the class of *relational graph models* (rgms) of λ-calculus, which are the relational analogous of graph models, and describe them as non-idempotent intersection type systems. This class is general enough to encompass all relational models individually introduced in the literature, including $\mathcal{D}_\omega$. We then show that: (*i*) all rgms satisfy an approximation theorem for resource approximants; (*ii*) in any rgm preserving the polarities of its "empty type" ω, β-normalisable λ-terms can be easily characterized. As a consequence, we get that all extensional rgms preserving ω-polarities induce as order-theory Morris's observational pre-order, and hence $\equiv^{\mathrm{nf}}$ as equality. As an instance, we provide the rgm $\mathcal{D}_\star$ generated by $\star \to \star \simeq \star$ where $\star$ is the only atom.

Finally, we introduce a notion of *extensional Taylor expansion* characterising, like extensional Böhm trees, Morris's equivalence while keeping the quantitative information. Intuitively, the extensional Taylor expansion of a λ-term is the η-normal form of its resource approximants. The definition is tricky because the η-reduction is a *global* operation — one should look at the whole series of approximants to decide whether an element should reduce or not. Our solution is to define a labeling as a global operation on the series of approximants, and then a local η-reduction on labeled terms. Two programs are then $\equiv^{\mathrm{nf}}$-equivalent exactly when they have the same extensional Taylor expansion.

References

1. A. Bucciarelli, T. Ehrhard, and G. Manzonetto. Not enough points is enough. In *CSL'07*, volume 4646 of *LNCS*, pages 298–312. Springer, 2007.
2. M. Coppo, M. Dezani, and M. Zacchi. Type theories, normal forms and D_∞-lambda-models. *Inf. Comput.*, 72(2):85–116, 1987.
3. T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1):1–41, 2003.
4. T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008.
5. J.-J. Levy. Le lambda calcul - notes du cours, 2005. In French, `http://pauillac.inria.fr/~levy/courses/X/M1/lambda/dea-spp/jjl.pdf`.
6. G. Manzonetto. A general class of models of $\mathcal{H}^\star$. In *MFCS 2009*, volume 5734 of *LNCS*, pages 574–586. Springer, 2009.
7. G. Manzonetto and D. Ruoppolo. Relational graph models, taylor expansion and extensionality. In *MFPS'14*, 2014. To appear.
8. R. Nakajima. Infinite normal forms for the lambda calculus. In *Lambda-Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, pages 62–82. Springer, 1975.
9. D. Scott. Continuous lattices. In Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, volume 274 of *Lecture Notes in Math.*, pages 97–136. Springer, 1972.