

EDITORIAL PREFACE

Preface to the special issue on metric and differential semantics

Ugo Dal Lago¹ , Francesco Gavazzo² and Paolo Pistone¹

¹Department of Computer Science and Engineering, University of Bologna, Bologna, Italy and

²Department of Computer Science, University of Pisa, Pisa, Italy

Corresponding author: Ugo Dal Lago; Email: ugo.dallago@unibo.it

(Received 2 July 2023; accepted 3 July 2023; first published online 7 August 2023)

Programming language semantics traditionally deals with *qualitative* properties of programs, that is, properties that a program may either satisfy or not, like termination or correctness. Moreover, program semantics generally attribute programs a meaning, typically a function of some kind, so as to be able to identify programs which behave in the same way in all contexts (i.e., which have *the same* meaning). This can be done in many different ways, from observational equivalence – the coarsest adequate congruence – to various forms of formal systems in the style of equational logic, to denotational semantics.

Nevertheless, the past ten years have seen the introduction of a series of logical and semantic frameworks which go significantly beyond this picture: on the one hand, frameworks enabling the expression of *quantitative* properties, i.e., properties that a program may satisfy *to a certain extent* or *up to a certain error* (e.g., probabilistic termination or correctness up to some error probability or some approximation error). Moreover, the meaning attributed to programs may allow the latter to be compared in quantitative ways, that is, as behaving in a *similar*, although not exactly equivalent, way, or to analyze how *sensitive* programs are to variations in their input. We refer here, for example, to approaches like behavioral and program metrics, differential semantics, automatic differentiation, sensitivity analysis and its application to differential privacy. These frameworks have progressively led to integrate methods coming from probabilistic programming, approximate and incremental computing, as well as machine learning within several standard theoretical approaches to program semantics.

This special issue is meant to collect contributions along these lines and comprises the following five papers:

- “*Up-To Techniques for Behavioural Metrics via Fibrations*,” by Bonchi, König, and Petrisan. This deals with the problem of deriving enhancements to the metric analog of the bisimulation proof method in an abstract way and with how categorical fibrations turn out to be a powerful tool for that.
- “*Bisimulation and Behavioural Equivalences for Continuous-time Markov Processes*,” by Chen, Clerc, and Panangaden. This contribution gives a unified view of various notions of behavioral equivalence and bisimulation for probabilistic transition systems whose time evolution is continuous rather than discrete.
- “*Coherent Differentiation*,” by Ehrhard. This paper introduces a new categorical framework for higher order program differentiation. In contrast to usual approaches based on the differential λ -calculus, this new framework does not require additivity (hence nondeterminism), and is thus compatible with both deterministic and probabilistic computational models.

- “*CHAD for Expressive Total Languages*,” by Nunes and Vakař. This work studies forward and reverse mode combinatory homomorphic automatic differentiation (CHAD) in the setting of total functional programming languages with rich type systems and proves the correctness of CHAD relying on categorical semantic techniques.
- “*Divergences on Monads for Relational Program Logics*,” by Sato and Katsumata. This paper introduces divergences on monads as a way to measure the quantitative difference between computational effects, as well as a relational program logic to reason about such divergences.

We would like to thank all people involved in this special issue. We would first of all like to warmly thank the authors for agreeing to publish their exciting results here. We then feel indebted to the reviewers for their crucial work. Finally, special thanks go to the editor in chief, Pierre-Louis Curien, for his support and patience.