Linear Types and Resources for the Solos Calculus

Jim Laird
Department of Computer Science, University of Bath, UK

The solos calculus [5] is a low-level language for describing concurrent programs and systems. It is a calculus of pure mobility, with atomic primitives for communication of channel names, but no explicit sequencing, prefixing, guarding etc. Instead, these can be expressed using basic synchronization protocols.

We investigate the syntax and semantics of the solos calculus from this perspective, presenting a version of the calculus with explicit fusions (avoiding any global rules for unification of variable names) and a typing system (unidirectionality) based explicitly on linear logic. This has a denotational semantics in the category of free modules over any complete semiring, which captures a notion of testing equivalence based on summing the residues of homotopically distinct reduction paths in a complete semiring [4]. Underlying it is a particularly simple graphical representation of unidirectional terms as weighted *differential nets* [3].

Translation into the solos calculus thus provides a quantitative and graphical interpretation of higher-level languages, including the λ-calculus and the π-calculus (for which we reconstruct a sum-of-paths interpretation and relate it to the work of Beffara [1]). These vary according to the degree to which sequencing of events must be specified — this can be captured using a typing system for solos terms which associates them to event structures, which can also be used to establish the acyclicity property [2] (essentially, that a term does not attempt to fuse a name to itself, which in our interpretation endlessly consumes resources). The linear typing system can also be modified to impose bounds on reduction based on existing systems for implicit and explicit computational complexity and resource use.

References

[1] Emmanuel Beffara. Quantitative testing semantics for non-interleaving. *CoRR*, abs/0906.3994, 2009.

[2] T. Ehrhard and O. Laurent. Acyclic solos and differential interaction nets. *Logical Methods in Computer Science*, 6(3), 2010.

[3] T. Ehrhard and L. Regnier. Differential interaction nets. *Theoretical Computer Science*, 364(2):166–195, 2006.

[4] J. Laird. Weighted relational models for mobility. In *Formal Structures in Computation and Deduction (FSCD '16)*, number 52 in Leibniz International Proceedings in Computer Science, 2016.

[5] C. Laneve and B. Victor. Solos in concert. *Mathematical Structures in Computer Science*, 13(5), 2003.