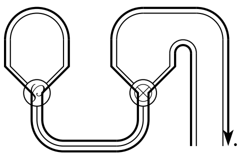
A Categorical Approach to the Geometry of Synchronisation

Ugo Dal Lago* Naohiko Hoshino†

Abstract

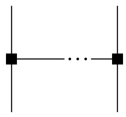
In [3, 7], a proof net system equipped with a specific node modelling synchronisation is introduced so as to give interaction-based semantics for computational systems that support parallelism and synchronisation mechanism, such as quantum programming languages. The interaction-based semantics is obtained by associating proof nets with multi-token machines, which enjoy some nice properties: confluence, uniformity and uniqueness of normal forms. However, these properties are shown in a concrete setting, which is not compositional. The aim of this talk is to give a general perspective and compositional reasoning principles for multi-token machines in terms of traced monoidal categories and the **Int**-construction.

Girard's Geometry of Interaction (GoI) [5] can be seen as a programming language semantics where execution of a program is modelled as an interaction process between some mathematical objects called *token machines*. The way token machines interact with each other in the original GoI semantics [8, 4] is called *particle-style* because the interactive process happening between token machines can be modeled by the trajectory of a single particle called a *token* as follows:

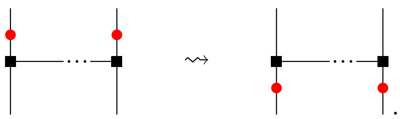


Such local interaction process in GoI semantics can capture execution of sequential programs such as those of PCF, and we can apply particle-style GoI semantics to give a correct-by-construction low-level implementation of (higher-order) programming languages [8].

What about programs with some parallelism or even with some synchronisation mechanism between multiple processes? For example, in quantum computation, we may have several computation processes in parallel, and because of quantum entanglement and measurement, these computation processes may interact with each other. The extension of particle-style GoI proposed in [3, 7], called the Geometry of Synchronisation, is a way to natively model computation with parallelism and synchronisation mechanisms. The novelty of these works consists in the introduction of synchronisation nodes



that behaves as follows: any token (a red bullet in the following picture) on an input wire can cross the synchronisation node *only when* we have a token on every such input wire:



*University of Bologna & INRIA Sophia Anitpolis, ugo.dallago@unibo.it

†Kyoto University, naophiko@kurims.kyoto-u.ac.jp

Since the synchronisation process only makes sense in presence of multiple tokens, we need to keep track of these multiple tokens along the token machine's execution. As reasoning principle for interaction process with multiple tokens, it is shown in [3, 7] that multi-token machines enjoy some nice properties: confluence, uniqueness of normal forms and uniformity. However, their work is done in a concrete setting, which lacks compositionality.

The aim of this talk is to give a general perspective and some compositional reasoning principle for multi-token machines in terms of traced monoidal categories and **Int**-construction. In [2], it is shown that we can understand particle-style GoI by traced monoidal categories and **Int**-construction, and the generality of these categorical notions provides many other categorical examples of "GoI." One of such GoI is obtained by the category **Cppo** of pointed dcpos and continuous functions. Our contributions are as follows::

- We show that we can capture the synchronisation node and multiple tokens running in parallel by the traced symmetric monoidal category **Cppo**.
- We prove uniqueness of normal forms in a general setting as a consequence of the existence of least fixed points of continuous functions in **Cppo**.

Related Works There can be other approaches towards understanding synchronisation mechanism in terms of traced monoidal categories. For example, using profunctors as in [6] may be a good candidate. Abramsky introduced a model of classical linear logic using CCS, based on the idea of GoI [1], and it would be possible to embed the synchronisation mechanism in [3, 7] into process realisability. Understanding process realisability in terms of traced monoidal categories is an interesting, although challenging problem.

References

[1] Samson Abramsky. Process realizability: (abstract). *Electronic Notes in Theoretical Computer Science*, 23(1):1–2, 1999.

[2] Samson Abramsky, Esfandiar Haghverdi, and Philip Scott. Geometry of interaction and linear combinatory algebras. *Mathematical. Structures in Comp. Sci.*, 12(5):625–665, oct 2002.

[3] Ugo Dal Lago, Claudia Faggian, Ichiro Hasuo, and Akira Yoshimizu. The geometry of synchronization. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, pages 35:1–35:10, New York, NY, USA, 2014. ACM.

[4] Vincent Danos and Laurent Regnier. Reversible, irreversible and optimal λ-machines. *Theoretical Computer Science*, 227(1):79–97, 1999.

[5] Jean-Yves Girard. Geometry of interaction 1: Interpretation of system F. In R. Ferro, C. Bonotto, S. Valentini, and A. Zanardo, editors, *Logic Colloquium '88*, volume 127 of *Studies in Logic and the Foundations of Mathematics*, pages 221–260. Elsevier, 1989.

[6] Thomas T. Hildebrandt, Prakash Panangaden, and Glynn Windskel. A relational model of non-deterministic dataflow. *Mathematical Structures in Computer Science*, 14(5):613–649, 2004.

[7] Ugo Dal Lago, Claudia Faggian, Benoit Valiron, and Akira Yoshimizu. Parallelism and synchronization in an infinitary context. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS '15, pages 559–572, Washington, DC, USA, 2015. IEEE Computer Society.

[8] Ian Mackie. The geometry of interaction machine. In Ron K. Cytron and Peter Lee, editors, *Conference Record of POPL'95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Francisco, California, USA, January 23-25, 1995*, pages 198–208. ACM Press, 1995.