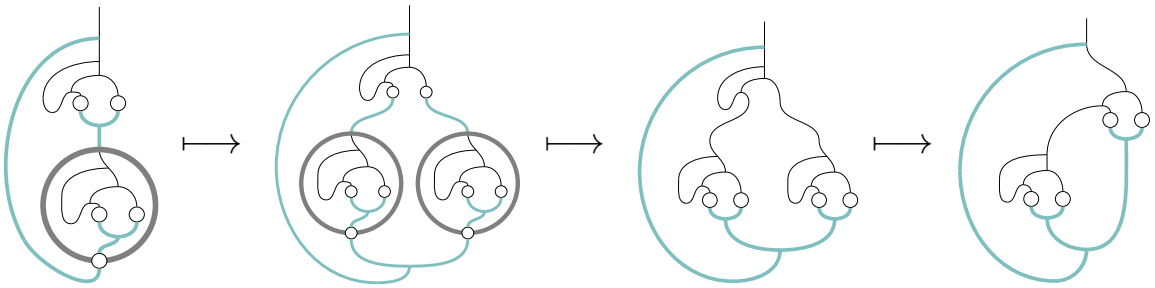
RESEARCH PROJECT: Programming in three dimensions

What is computation? The pioneering work of Gödel, Turing and Church in the 1930s organised the theory of computation around several different, but equivalent, definitions of the mathematical objects that we now call "computer programs". Recent developments in mathematics, computer science and physics are leading to a re-imagining of the nature these objects. For example: programs and their execution are often presented using string diagrams and transformations of diagrams, either for λ-terms in Church's approach to computation or in related approaches like linear logic, an abstract functional programming language introduced by Girard [2]. The following diagram from [6] depicts steps in a linear logic program:



These diagrams are manifestly 2-dimensional objects of a topological or geometric character. This makes one wonder: is there such a thing as 3-dimensional programs? How would one write them? What would they be good for? The research project I have in mind for Will Troiani aims to create and study such programs, in two steps:

(1) The first step will be to understand the category theory involved in making rigorous sense of the above two-dimensional diagrams [5] and the existing literature on *noncommutative linear logic* which extends these diagrams into the setting of braided monoidal categories [1, 2, 3, 4]. These extensions have a natural interpretation as string diagrams in three dimensions. Under the Curry-Howard correspondence these categorical logical systems can be interpreted as a kind of programming language. The upshot is that, at least formally, 3-dimensional programs exist. This leaves the question of what to do with them.

(2) The second step will be to write code to enable the creation of programs in this language in some 3D editing tool (here we will investigate modern cheap VR interfaces such as Samsung Gear) and then write further code to execute these programs. Ideally, this execution would be an animated and visual affair, showing by example the dynamic nature of program execution in diagrammatic form. Finally, there will be an exploration of "exotic" programs which can be naturally written in this language, using its 3D nature in an intrinsic way.

REFERENCES

[1] M. Barr, $\star$-autonomous categories and linear logic, Mathematical Structures in Computer Science **1(2)** (1991) 159–178.

[2] J.-Y. Girard, *Linear Logic*, Theoretical Computer Science **50** (1987), 1–102.

[3] J.-Y. Girard, Y. Lafont, and P. Taylor, *Proofs and Types*, Cambridge Tracts in Theoretical Computer Science 7 ,Cambridge University Press, 1989.

[4] J. Lambek and P. J. Scott, *Introduction to higher order categorical logic*, Cambridge Studies in Advanced Mathematics, vol. 7, Cambridge University Press, Cambridge, 1986.

[5] P-A. Melliès, *Categorical semantics of linear logic*, in : Interactive models of computation and program behaviour, Panoramas et Synthèses 27, Société Mathématique de France, 2009.

[6] D. Murfet, *Logic and linear algebra: an introduction*, preprint (2014) [arXiv: 1407.2650].