Linear Realisability and Cobordism: Understanding the Trefoil Property

Thomas Seiller
CNRS

Valentin Maestracci
Aix-Marseille University

May 20, 2025

Linear realisability[4] is a research program in the continuation of Geometry of Interaction, where logic is studied from the point of view of models of computation. Many such models of computation have been defined (ludics, flows, stellar resolution...), among which the interaction graphs models. Interaction graphs traditionally do not suffice to define an interesting enough ∗-autonomous category, because of the disparition of cycles between axioms and cuts during execution. We take inspiration in the category of cobordism to try to overcome this problem and define a category generalizing interaction graphs, namely the category of interaction semi-categories.

1. Categorical perspective on Interaction Graphs

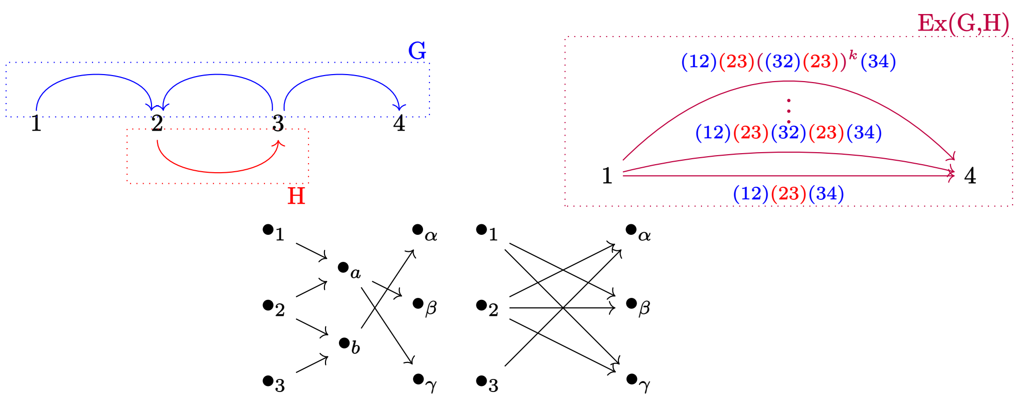
Interactions Graphs (IG) are a model of computation introduced by Seiller in [3]. They are usually presented directly as graphs, equipped with a notion of execution corresponding to computing paths, as in figure 1. They are used to construct dynamical semantics for various fragments of linear logic.

But they can be understood as a form of Int-construction on the following span-like category (illustrated in figure 1) that admits a categorical trace (it is a proof relevant generalisation of usual GoI categories, Rel_+, partial functions etc...):

Definition 1 (2*Graph*) *The category* 2*Graph has finite sets as objects and a morphism* $G : A \to B$ *is an oriented, bipartite graph with nodes* $A \sqcup B$ *and whose edges have their source in* A *and target in* B*. Note the disjoint union of the vertices. Morphisms are composed as spans, which corresponds to taking paths of length* 2*.*

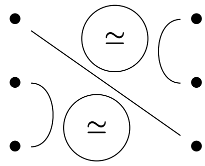
This notion of graph does not suffice, unfortunately, to internally express the notion of cycle, which is important to characterize the correctness criterion of linear logic. Indeed, in figure 1, the cycle between 2 and 3 disappears during execution. (Worse, there isn't even a trace left of completely internal cycles). To overcome this, Girard and Seiller introduce a notion of "wager" counting cycles. This extends the model, giving rise to a category **Project** in which objects are a pair of a graph and a wager. The extended model still has an associative execution, due to a property called the *trefoil property* [2].

This phenomenon does not appear in the category of cobordism, which led us to investigate it.

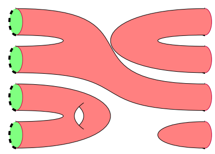


(a) Two morphisms (b) Paths

Figure 1: Two graphs and their execution, Two morphisms and their (elementary) composition



(a) A 1-cobordism



(b) A 2-cobordism

Figure 2: Examples of cobordisms

2. Cobordism

Cobordisms are geometrical objects, such as those illustrated in figure 2. They are the morphism of their own compact closed category, in which objects are boundaries and composition is defined by glueing ends together. This category has a rich structure of scalars: a sphere or a torus have no boundaries and therefore are morphisms $\emptyset \to \emptyset$. The glueing is reminiscent of how our graphs compose, but is not exactly similar: obordisms form a Cospan category[1], not a Span category.

3. Lesson from Cob

It seems the reason internal cycles do not disappear in the case of cobordisms comes from their geometric nature. We would thus like to keep both a geometric nature (glueing), and keep a notion of dynamics which is the original goal of GoI.

Categorically, we represented interaction graphs as Spans. But one could also represent interaction graphs as a form of "cospan":

$$\begin{array}{ccccccccc} & & & & G\square H & & & & \\ & & & \nearrow & & \nwarrow & & & \\ & & G & & & & H & & \\ & \nearrow & & \nwarrow & & \nearrow & & \nwarrow & \\ X & & & & Y & & & & Z \end{array}$$

where the composition of two cospans happens in 3 steps: taking a pushout (glueing), applying a free functor to compute all possible paths, and then hide the "partial paths" that do no connect two extremities. Doing this, one can start from a category of *open graphs*, very similar to the one in the formalism of "open system", that is a double categories of structured cospans defined by Baez and Courser [1].

By applying an Int-construction, this defines a category of graphs, whose paths are exactly the alternated paths inside the execution. Now, by applying a "Free" functor, one gets something close to a category whose objects are sets, and morphisms are not just graphs between said sets, but the data of all po paths. That is, this data defines a semi-category.

All previous constructions on graphs can then be recovered by different "Hiding" functors that would erase more or less data. The situation is summed up in the following drawing:

$$\begin{array}{ccccc} openGraph & \xrightarrow{Int} & gG & \xrightarrow{Free} & isemCat \\ \downarrow{\scriptstyle Ex} & & \downarrow & \swarrow{\scriptstyle H} & \downarrow{\scriptstyle H+wager} \\ 2Graph & \xrightarrow[Int]{} & iG & \xhookrightarrow[wager=0]{} & \mathbf{Project} \end{array}$$

Note that the upper layer no longer defines a denotational model of logic, but *contains many*. One can then take two things out of this:

1. It seems the wager is a construction that allows to extend the original model of computation, to still get an interesting "sub-model" of the here-defined category. Are there others?
2. This construction "lost" the dynamics aspect, since the paths are still constructed all at once. But it isn't truly lost: we can try to adapt it to a 2-categorical context where paths are just partially computed. The dynamics would lie in computing these further.

[1] Some subtleties are hidden here.

References

[1] John C. Baez, Kenny Courser, and Christina Vasilakopoulou. “Structured versus Decorated Cospans”. In: *Compositionality* 4 (Sept. 2022), p. 3. (Visited on 03/17/2025).

[2] Thomas Seiller. “Interaction Graphs: Additives”. In: *Annals of Pure and Applied Logic* 167.2 (Feb. 2016), pp. 95–154. (Visited on 03/17/2025).

[3] Thomas Seiller. “Interaction Graphs: Multiplicatives”. In: *Annals of Pure and Applied Logic* 163.12 (Dec. 2012), pp. 1808–1837. (Visited on 03/17/2025).

[4] Thomas Seiller. “Mathematical Informatics”. Thesis. Université Sorbonne Paris Nord, June 2024. (Visited on 03/17/2025).