



# String diagram

**String diagrams** are a formal graphical language for representing morphisms in monoidal categories, or more generally 2-cells in 2-categories. They are a prominent tool in applied category theory. When interpreted in the monoidal category of vector spaces and linear maps with the tensor product, string diagrams are called tensor networks or Penrose graphical notation. This has led to the development of categorical quantum mechanics where the axioms of quantum theory are expressed in the language of monoidal categories.

## History

Günter Hotz gave the first mathematical definition of string diagrams in order to formalise electronic circuits.<sup>[1]</sup> However, the invention of string diagrams is usually credited to Roger Penrose,<sup>[2]</sup> with Feynman diagrams also described as a precursor.<sup>[3]</sup> They were later characterised as the arrows of free monoidal categories in a seminal article by André Joyal and Ross Street.<sup>[4]</sup> While the diagrams in these first articles were hand-drawn, the advent of typesetting software such as LaTeX and PGF/TikZ made the publication of string diagrams more wide-spread.<sup>[5]</sup>

The existential graphs and diagrammatic reasoning of Charles Sanders Peirce are arguably the oldest form of string diagrams, they are interpreted in the monoidal category of finite sets and relations with the Cartesian product.<sup>[6]</sup> The **lines of identity** of Peirce's existential graphs can be axiomatised as a Frobenius algebra, the **cuts** are unary operators on homsets that axiomatise logical negation. This makes string diagrams a sound and complete two-dimensional deduction system for first-order logic,<sup>[7]</sup> invented independently from the one-dimensional syntax of Gottlob Frege's Begriffsschrift.

## Intuition

String diagrams are made of **boxes**  $f : x \rightarrow y$ , which represent processes, with a list of **wires**  $x$  coming in at the top and  $y$  at the bottom, which represent the input and output systems being processed by the box  $f$ . Starting from a collection of wires and boxes, called a **signature**, one may generate the set of all string diagrams by induction:

- each box  $f : x \rightarrow y$  is a string diagram,
- for each list of wires  $x$ , the identity  $\text{id}(x) : x \rightarrow x$  is a string diagram representing the process which does nothing to its input system, it is drawn as a bunch of parallel wires,
- for each pair of string diagrams  $f : x \rightarrow y$  and  $f' : x' \rightarrow y'$ , their tensor  $f \otimes f' : xx' \rightarrow yy'$  is a string diagram representing the parallel composition of processes, it is drawn as the horizontal concatenation of the two diagrams,

- for each pair of string diagrams  $f : x \rightarrow y$  and  $g : y \rightarrow z$ , their composition  $g \circ f : x \rightarrow z$  is a string diagram representing the sequential composition of processes, it is drawn as the vertical concatenation of the two diagrams.

## Definition

---



---

### Algebraic

Let the Kleene star  $X^*$  denote the free monoid, i.e. the set of lists with elements in a set  $X$ .

A **monoidal signature**  $\Sigma$  is given by:

- a set  $\Sigma_0$  of **generating objects**, the lists of generating objects in  $\Sigma_0^*$  are also called **types**,
- a set  $\Sigma_1$  of **generating arrows**, also called **boxes**,
- a pair of functions  $\text{dom}, \text{cod} : \Sigma_1 \rightarrow \Sigma_0^*$  which assign a **domain** and **codomain** to each box, i.e. the input and output types.

A morphism of monoidal signature  $F : \Sigma \rightarrow \Sigma'$  is a pair of functions  $F_0 : \Sigma_0 \rightarrow \Sigma'_0$  and  $F_1 : \Sigma_1 \rightarrow \Sigma'_1$  which is compatible with the domain and codomain, i.e. such that  $\text{dom} \circ F_1 = F_0 \circ \text{dom}$  and  $\text{cod} \circ F_1 = F_0 \circ \text{cod}$ . Thus we get the category **MonSig** of monoidal signatures and their morphisms.

There is a forgetful functor  $U : \mathbf{MonCat} \rightarrow \mathbf{MonSig}$  which sends a monoidal category to its underlying signature and a monoidal functor to its underlying morphism of signatures, i.e. it forgets the identity, composition and tensor. The free functor  $C_- : \mathbf{MonSig} \rightarrow \mathbf{MonCat}$ , i.e. the left adjoint to the forgetful functor, sends a monoidal signature  $\Sigma$  to the free monoidal category  $C_\Sigma$  it generates.

String diagrams (with generators from  $\Sigma$ ) are arrows in the free monoidal category  $C_\Sigma$ .<sup>[8]</sup> The interpretation in a monoidal category  $D$  is defined by a monoidal functor  $F : C_\Sigma \rightarrow D$ , which by freeness is uniquely determined by a morphism of monoidal signatures  $F : \Sigma \rightarrow U(D)$ . Intuitively, once the image of generating objects and arrows are given, the image of every diagram they generate is fixed.

### Geometric

A topological graph, also called a one-dimensional cell complex, is a tuple  $(\Gamma, \Gamma_0, \Gamma_1)$  of a Hausdorff space  $\Gamma$ , a closed discrete subset  $\Gamma_0 \subseteq \Gamma$  of **nodes** and a set of connected components  $\Gamma_1$  called **edges**, each homeomorphic to an open interval with boundary in  $\Gamma_0$  and such that  $\Gamma - \Gamma_0 = \coprod \Gamma_1$ .

A plane graph between two real numbers  $a, b \in \mathbb{R}$  with  $a < b$  is a finite topological graph embedded in  $\mathbb{R} \times [a, b]$  such that every point  $x \in \Gamma \cap \mathbb{R} \times \{a, b\}$  is also a node  $x \in \Gamma_0$  and belongs to the closure of exactly one edge in  $\Gamma_1$ . Such points are called **outer nodes**, they define the **domain** and **codomain**  $\text{dom}(\Gamma), \text{cod}(\Gamma) \in \Gamma_1^*$  of the string diagram, i.e. the list of edges that are connected to the top and bottom boundary. The other nodes  $f \in \Gamma_0 - \{a, b\} \times \mathbb{R}$  are called **inner nodes**.

A plane graph is **progressive**, also called **recumbent**, when the vertical projection  $e \rightarrow [a, b]$  is injective for every edge  $e \in \Gamma_1$ . Intuitively, the edges in a progressive plane graph go from top to bottom without bending backward. In that case, each edge can be given a top-to-bottom orientation with designated nodes

as source and target. One can then define the domain and codomain  $\mathbf{dom}(f), \mathbf{cod}(f) \in \Gamma_1^*$  of each inner node  $f$ , given by the list of edges that have source and target.

A plane graph is **generic** when the vertical projection  $\Gamma_0 - \{a, b\} \times \mathbb{R} \rightarrow [a, b]$  is injective, i.e. no two inner nodes are at the same height. In that case, one can define a list  $\mathbf{boxes}(\Gamma) \in \Gamma_0^*$  of the inner nodes ordered from top to bottom.

A progressive plane graph is **labeled** by a monoidal signature  $\Sigma$  if it comes equipped with a pair of functions  $v_0 : \Gamma_1 \rightarrow \Sigma_0$  from edges to generating objects and  $v_1 : \Gamma_0 - \{a, b\} \times \mathbb{R} \rightarrow \Sigma_1$  from inner nodes to generating arrows, in a way compatible with domain and codomain.

A **deformation** of plane graphs is a continuous map  $h : \Gamma \times [0, 1] \rightarrow [a, b] \times \mathbb{R}$  such that

- the image of  $h(-, t)$  defines a plane graph for all  $t \in [0, 1]$ ,
- for all  $x \in \Gamma_0$ , if  $h(x, t)$  is an inner node for some  $t$  it is inner for all  $t \in [0, 1]$ .

A deformation is progressive (generic, labeled) if  $h(-, t)$  is progressive (generic, labeled) for all  $t \in [0, 1]$ . Deformations induce an equivalence relation with  $\Gamma \sim \Gamma'$  if and only if there is some  $h$  with  $h(-, 0) = \Gamma$  and  $h(-, 1) = \Gamma'$ . String diagrams are **equivalence classes of labeled progressive plane graphs**. Indeed, one can define:

- the identity diagram  $\mathbf{id}(x)$  as a set of parallel edges labeled by some type  $x \in \Sigma_0^*$ ,
- the composition of two diagrams as their vertical concatenation with the codomain of the first identified with the domain of the second,
- the tensor of two diagrams as their horizontal concatenation.

## Combinatorial

While the geometric definition makes explicit the link between category theory and low-dimensional topology, a **combinatorial definition** is necessary to formalise string diagrams in computer algebra systems and use them to define computational problems. One such definition is to define string diagrams as equivalence classes of well-typed formulae generated by the signature, identity, composition and tensor. In practice, it is more convenient to encode string diagrams as formulae in **generic form**, which are in bijection with the labeled generic progressive plane graphs defined above.

Fix a monoidal signature  $\Sigma$ . A **layer** is defined as a triple  $(x, f, y) \in \Sigma_0^* \times \Sigma_1 \times \Sigma_0^* =: L(\Sigma)$  of a type  $x$  on the left, a box  $f$  in the middle and a type  $y$  on the right. Layers have a domain and codomain  $\mathbf{dom}, \mathbf{cod} : L(\Sigma) \rightarrow \Sigma_0^*$  defined in the obvious way. This forms a directed multigraph, also known as a quiver, with the types as vertices and the layers as edges. A **string diagram**  $d$  is encoded as a path in this multigraph, i.e. it is given by:

- a domain  $\mathbf{dom}(d) \in \Sigma_0^*$  as starting point
- a length  $\mathbf{len}(d) = n \geq 0$ ,
- a list of layers  $\mathbf{layers}(d) = d_1 \dots d_n \in L(\Sigma)$

such that  $\mathbf{dom}(d_1) = \mathbf{dom}(d)$  and  $\mathbf{cod}(d_i) = \mathbf{dom}(d_{i+1})$  for all  $i < n$ . In fact, the explicit list of layers is redundant, it is enough to specify the length of the type to the left of each layer, known as the **offset**. The **whiskering**  $d \otimes z$  of a diagram  $d = (x_1, f_1, y_1) \dots (x_n, f_n, y_n)$  by a type  $z$  is defined as

the concatenation to the right of each layer  $d \otimes z = (x_1, f_1, y_1 z) \dots (x_n, f_n, y_n z)$  and symmetrically for the whiskering  $z \otimes d$  on the left. One can then define:

- the identity diagram  $\text{id}(x)$  with  $\text{len}(\text{id}(x)) = 0$  and  $\text{dom}(\text{id}(x)) = x$ ,
- the composition of two diagrams as the concatenation of their list of layers,
- the tensor of two diagrams as the composition of whiskerings  
 $d \otimes d' = d \otimes \text{dom}(d') \circ \text{cod}(d) \otimes d'$ .

Note that because the diagram is in generic form (i.e. each layer contains exactly one box) the definition of tensor is necessarily biased: the diagram on the left hand-side comes above the one on the right-hand side. One could have chosen the opposite definition  $d \otimes d' = \text{dom}(d) \otimes d' \circ d \otimes \text{cod}(d')$ .

Two diagrams are equal (up to the axioms of monoidal categories) whenever they are in the same equivalence class of the congruence relation generated by the **interchanger**:

$$d \otimes \text{dom}(d') \circ \text{cod}(d) \otimes d' = \text{dom}(d) \otimes d' \circ d \otimes \text{cod}(d')$$

That is, if the boxes in two consecutive layers are not connected then their order can be swapped. Intuitively, if there is no communication between two parallel processes then the order in which they happen is irrelevant.

The word problem for free monoidal categories, i.e. deciding whether two given diagrams are equal, can be solved in polynomial time. The interchanger is a confluent rewriting system on the subset of **boundary connected** diagrams, i.e. whenever the plane graphs have no more than one connected component which is not connected to the domain or codomain and the Eckmann–Hilton argument does not apply.<sup>[9]</sup>

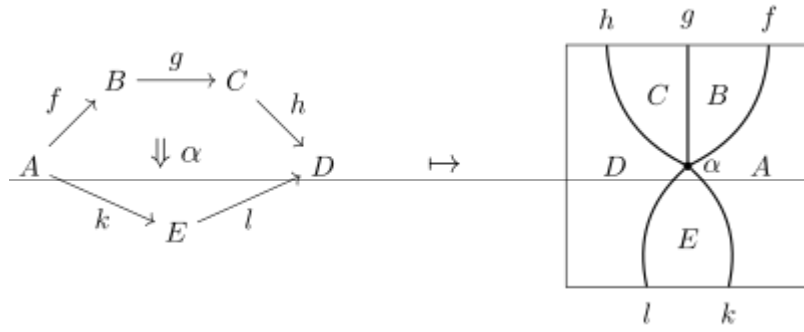
## Extension to 2-categories

---

The idea is to represent structures of dimension  $d$  by structures of dimension  $2-d$ , using Poincaré duality. Thus,

- an object is represented by a portion of plane,
- a 1-cell  $f : A \rightarrow B$  is represented by a vertical segment—called a *string*—separating the plane in two (the right part corresponding to  $A$  and the left one to  $B$ ),
- a 2-cell  $\alpha : f \Rightarrow g : A \rightarrow B$  is represented by an intersection of strings (the strings corresponding to  $f$  above the link, the strings corresponding to  $g$  below the link).

The parallel composition of 2-cells corresponds to the horizontal juxtaposition of diagrams and the sequential composition to the vertical juxtaposition of diagrams.



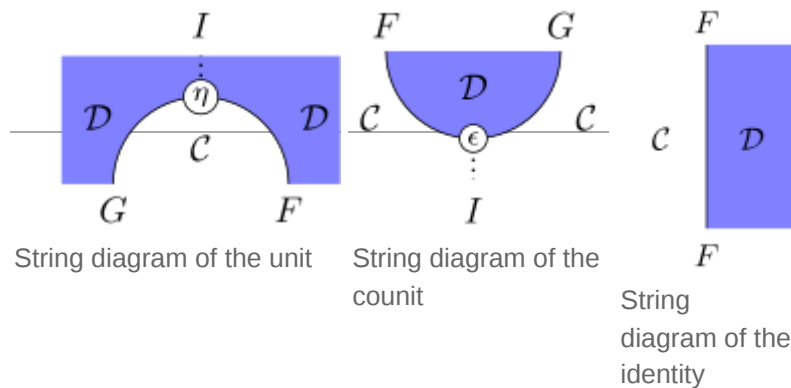
Duality between commutative diagrams (on the left hand side) and string diagrams (on the right hand side)

A monoidal category is equivalent to a 2-category with a single 0-cell. Intuitively, going from monoidal categories to 2-categories amounts to adding colours to the background of string diagrams.

## Examples

### The snake equation

Consider an adjunction  $(F, G, \eta, \epsilon)$  between two categories  $\mathcal{C}$  and  $\mathcal{D}$  where  $F : \mathcal{C} \leftarrow \mathcal{D}$  is left adjoint of  $G : \mathcal{C} \rightarrow \mathcal{D}$  and the natural transformations  $\eta : I \rightarrow GF$  and  $\epsilon : FG \rightarrow I$  are respectively the unit and the counit. The string diagrams corresponding to these natural transformations are:

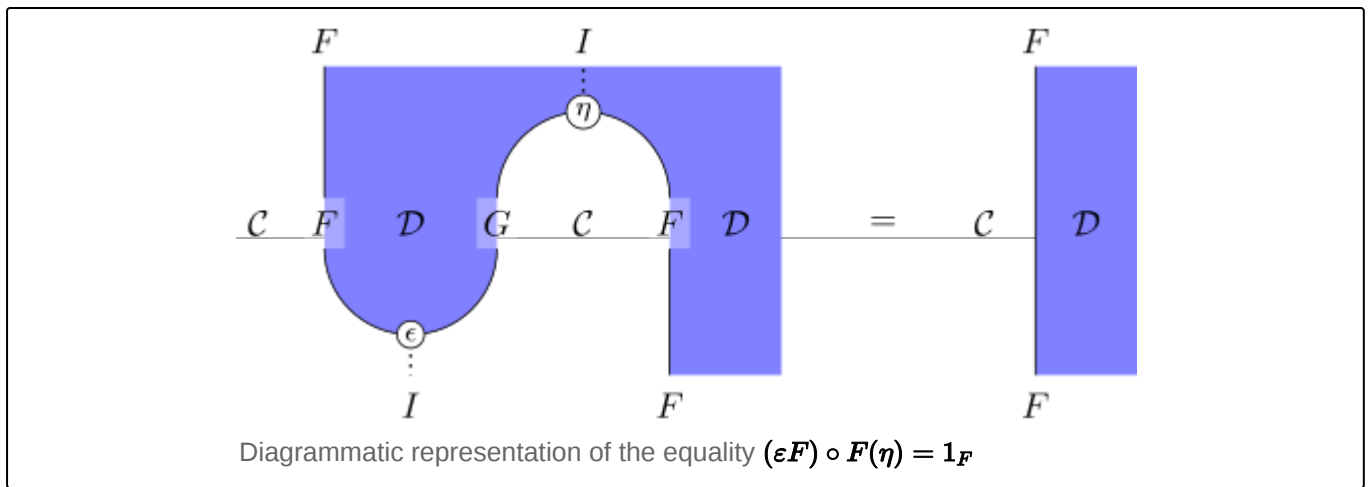


The string corresponding to the identity functor is drawn as a dotted line and can be omitted. The definition of an adjunction requires the following equalities:

$$(\epsilon F) \circ F(\eta) = 1_F$$

$$G(\epsilon) \circ (\eta G) = 1_G$$

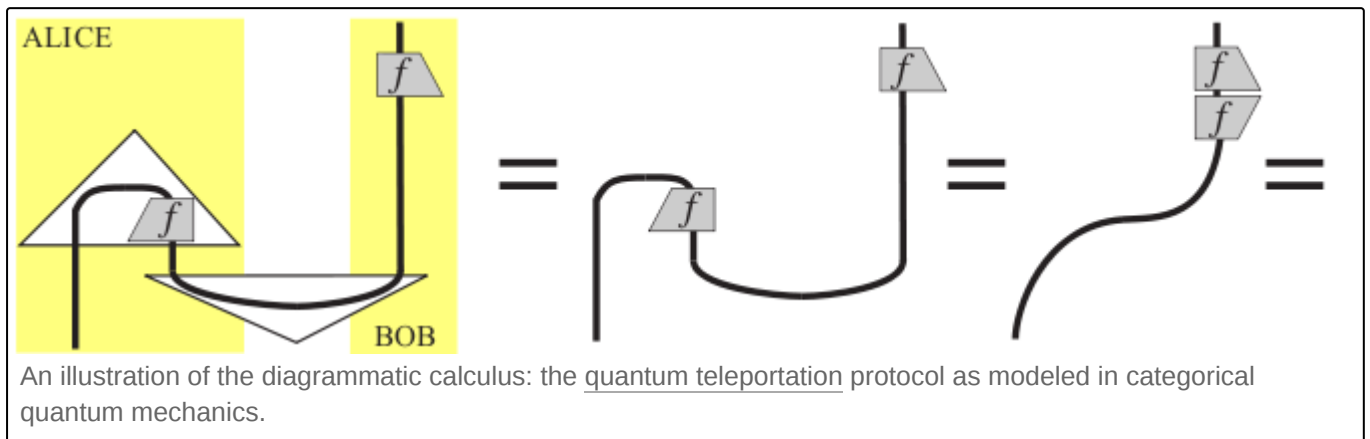
The first one is depicted as



A monoidal category where every object has a left and right adjoint is called a rigid category. String diagrams for rigid categories can be defined as **non-progressive** plane graphs, i.e. the edges can bend backward.

In the context of categorical quantum mechanics, this is known as the **snake equation**.

The category of Hilbert spaces is rigid, this fact underlies the proof of correctness for the quantum teleportation protocol. The unit and counit of the adjunction are an abstraction of the Bell state and the Bell measurement respectively. If Alice and Bob share two qubits Y and Z in an entangled state and Alice performs a (post-selected) entangled measurement between Y and another qubit X, then this qubit X will be teleported from Alice to Bob: quantum teleportation is an identity morphism.



The same equation appears in the definition of pregroup grammars where it captures the notion of information flow in natural language semantics. This observation has led to the development of the DisCoCat framework and quantum natural language processing.

## Hierarchy of graphical languages

Many extensions of string diagrams have been introduced to represent arrows in monoidal categories with extra structure, forming a hierarchy of graphical languages which is classified in Selinger's *Survey of graphical languages for monoidal categories*.<sup>[10]</sup>

- Braided monoidal categories with 3-dimensional diagrams, a generalisation of braid groups.

- Symmetric monoidal categories with 4-dimensional diagrams where edges can cross, a generalisation of the symmetric group.
- Ribbon categories with 3-dimensional diagrams where the edges are undirected, a generalisation of knot diagrams.
- Compact closed categories with 4-dimensional diagrams where the edges are undirected, a generalisation of Penrose graphical notation.
- Dagger categories where every diagram has a horizontal reflection.

## List of applications

---

String diagrams have been used to formalise the following objects of study.

- Concurrency theory<sup>[11]</sup>
- Artificial neural networks<sup>[12]</sup>
- Game theory<sup>[13]</sup>
- Bayesian probability<sup>[14]</sup>
- Consciousness<sup>[15]</sup>
- Markov kernels<sup>[16]</sup>
- Signal-flow graphs<sup>[17]</sup>
- Conjunctive queries<sup>[18]</sup>
- Bidirectional transformations<sup>[19]</sup>
- Categorical quantum mechanics
- Quantum circuits, measurement-based quantum computing and quantum error correction, see ZX-calculus
- Natural language processing, see DisCoCat
- Quantum natural language processing

## See also

---

- Proof nets, a generalisation of string diagrams used to denote proofs in linear logic
- Existential graphs, a precursor of string diagrams used to denote formulae in first-order logic
- Penrose graphical notation and Feynman diagrams, two precursors of string diagrams in physics
- Tensor networks, the interpretation of string diagrams in vector spaces, linear maps and tensor product

## References

---

1. Hotz, Günter (1965). "Eine Algebraisierung des Syntheseproblems von Schaltkreisen I.". *Elektronische Informationsverarbeitung und Kybernetik*. **1** (3): 185–205.
2. Penrose, Roger (1971). "Applications of negative dimensional tensors" (<https://scholar.googleusercontent.com/scholar.bib?q=info:DYCIJiWzIJQJ:scholar.google.com/&output=citation&scisdr=CgXswJALEJT6ufwbECU:AAGBfm0AAAAAY2odCCW7hYhwtWjn35RhUaHM9aZA6tiD&scisig=AAGBfm0AAAAAY2odCANHUoew7LZP05UGFPOFP2IA4Syr&scisf=4&ct=citation&cd=-1&hl=en>). *Combinatorial Mathematics and Its Applications*. **1**: 221–244.

3. Baez, J.; Stay, M. (2011), Coecke, Bob (ed.), "Physics, Topology, Logic and Computation: A Rosetta Stone" ([https://doi.org/10.1007/978-3-642-12821-9\\_2](https://doi.org/10.1007/978-3-642-12821-9_2)), *New Structures for Physics*, Lecture Notes in Physics, vol. 813, Berlin, Heidelberg: Springer, pp. 95–172, arXiv:0903.0340 (<https://arxiv.org/abs/0903.0340>), Bibcode:2011LNP...813...95B (<https://ui.adsabs.harvard.edu/abs/2011LNP...813...95B>), doi:10.1007/978-3-642-12821-9\_2 ([https://doi.org/10.1007%2F978-3-642-12821-9\\_2](https://doi.org/10.1007%2F978-3-642-12821-9_2)), ISBN 978-3-642-12821-9, S2CID 115169297 (<http://api.semanticscholar.org/CorpusID:115169297>), retrieved 2022-11-08
4. Joyal, André; Street, Ross (1991). "The geometry of tensor calculus, I". *Advances in Mathematics*. **88** (1): 55–112. doi:10.1016/0001-8708(91)90003-P (<https://doi.org/10.1016%2F0001-8708%2891%2990003-P>).
5. "Categories: History of string diagrams (thread, 2017may02-...)" (<http://angg.twu.net/categories-2017may02.html#1>). *angg.twu.net*. Retrieved 2022-11-11.
6. Brady, Geraldine; Trimble, Todd H (2000). "A categorical interpretation of CS Peirce's propositional logic Alpha" (<https://scholar.googleusercontent.com/scholar.bib?q=info:z17ig3wfQZ0J:scholar.google.com/&output=citation&scisdr=CgXswJALEJT6ufwi7EM:AAGBfm0AAAY2ok9ENrthZYwCuLljH9jsSEukPjkBaE&scisig=AAGBfm0AAAAAY2ok9MQgwqWjQgHWkelw5gGb82nKMBSR&scisf=4&ct=citation&cd=-1&hl=en>). *Journal of Pure and Applied Algebra*. **149** (3): 213–239. doi:10.1016/S0022-4049(98)00179-0 (<https://doi.org/10.1016%2FS0022-4049%2898%2900179-0>).
7. Haydon, Nathan; Sobociński, Paweł (2020). "Compositional diagrammatic first-order logic" ([https://scholar.googleusercontent.com/scholar.bib?q=info:pd5MrLoLlLQJ:scholar.google.com/&output=citation&scisdr=CgXswJALEJT6ufwgXcs:AAGBfm0AAAAAY2omRcs9kaERdk6UxEg\\_v331hutAjGXP&scisig=AAGBfm0AAAAAY2omRSLM2O\\_mGNH9sYDVO1ThNJZEohlv&scisf=4&ct=citation&cd=-1&hl=en](https://scholar.googleusercontent.com/scholar.bib?q=info:pd5MrLoLlLQJ:scholar.google.com/&output=citation&scisdr=CgXswJALEJT6ufwgXcs:AAGBfm0AAAAAY2omRcs9kaERdk6UxEg_v331hutAjGXP&scisig=AAGBfm0AAAAAY2omRSLM2O_mGNH9sYDVO1ThNJZEohlv&scisf=4&ct=citation&cd=-1&hl=en)). *International Conference on Theory and Application of Diagrams*. Springer: 402–418.
8. Joyal, André; Street, Ross (1988). "Planar diagrams and tensor algebra" (<https://scholar.googleusercontent.com/scholar.bib?q=info:TfFDpTlfWZUJ:scholar.google.com/&output=citation&scisdr=CgXswJALEJT6ufw-rGM:AAGBfm0AAAAAY2o4tGPckoYZUFk0JBWakD3cW6bJqMhy&scisig=AAGBfm0AAAAAY2o4tLofCCFsCbKd5Kt7puVkFPptvNYP&scisf=4&ct=citation&cd=-1&hl=en>). *Unpublished Manuscript, Available from Ross Street's Website*.
9. Vicary, Jamie; Delpeuch, Antonin (2022). "Normalization for planar string diagrams and a quadratic equivalence algorithm" (<https://scholar.googleusercontent.com/scholar.bib?q=info:0hLKQDlelEoJ:scholar.google.com/&output=citation&scisdr=CgUx7uJgEJT6uf2E8kc:AAGBfm0AAAAAY2uC6kfyvrxazKIUNsmStrSSEtivNgu3&scisig=AAGBfm0AAAAAY2uC6qabf-0CYsEHJcuQFVW5JfUgn5rm&scisf=4&ct=citation&cd=-1&hl=en>). *Logical Methods in Computer Science*. **18**.
10. Selinger, Peter (2010), "A survey of graphical languages for monoidal categories" ([https://scholar.googleusercontent.com/scholar.bib?q=info:QQAiC8FDpEkJ:scholar.google.com/&output=citation&scisdr=CgXswJALEJT6ufwkDI8:AAGBfm0AAAAAY2oiFI-xbrK6Yvmdu1pm4UTxzjcsBmsl&scisig=AAGBfm0AAAAAY2oiFif9Jw73lkbjf\\_-NmQG8dYgwrYj&scisf=4&ct=citation&cd=-1&hl=en](https://scholar.googleusercontent.com/scholar.bib?q=info:QQAiC8FDpEkJ:scholar.google.com/&output=citation&scisdr=CgXswJALEJT6ufwkDI8:AAGBfm0AAAAAY2oiFI-xbrK6Yvmdu1pm4UTxzjcsBmsl&scisig=AAGBfm0AAAAAY2oiFif9Jw73lkbjf_-NmQG8dYgwrYj&scisf=4&ct=citation&cd=-1&hl=en)), *New structures for physics*, Springer, pp. 289–355, retrieved 2022-11-08
11. Abramsky, Samson (1996). "Retracing some paths in process algebra" ([https://scholar.googleusercontent.com/scholar.bib?q=info:2ZQv\\_XcOfWIJ:scholar.google.com/&output=citation&scisdr=CgUx7uJgEJT6uf2JddA:AAGBfm0AAAAAY2uPbdDaGQFHjZkHLuzjXJ1-TWNxoAi&scisig=AAGBfm0AAAAAY2uPbXS9abvbrYityk1sYnsz4q1NFzDI&scisf=4&ct=citation&cd=-1&hl=en](https://scholar.googleusercontent.com/scholar.bib?q=info:2ZQv_XcOfWIJ:scholar.google.com/&output=citation&scisdr=CgUx7uJgEJT6uf2JddA:AAGBfm0AAAAAY2uPbdDaGQFHjZkHLuzjXJ1-TWNxoAi&scisig=AAGBfm0AAAAAY2uPbXS9abvbrYityk1sYnsz4q1NFzDI&scisf=4&ct=citation&cd=-1&hl=en)). *International Conference on Concurrency Theory*. Springer: 1–17.
12. Fong, Brendan; Spivak, David I.; Tuyéras, Rémy (2019-05-01). "Backprop as Functor: A compositional perspective on supervised learning". arXiv:1711.10455 (<https://arxiv.org/abs/1711.10455>) [math.CT (<https://arxiv.org/archive/math>.CT)].



13. Ghani, Neil; Hedges, Jules; Winschel, Viktor; Zahn, Philipp (2018). "Compositional Game Theory". *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science* ([https://scholar.googleusercontent.com/scholar.bib?q=info:LGUG3U6Bp-kJ:scholar.google.com/&output=citation&scisdr=CgUx7uJgEJT6uf13NIw:AAGBfm0AAAAAY2txLizGMVUpPg6\\_ihwN2dU8PlulpMG-4&scisig=AAGBfm0AAAAAY2txLO2TG0gZXx2xxpt2AfWUrr5KIziD&scisf=4&ct=citation&cd=-1&hl=en](https://scholar.googleusercontent.com/scholar.bib?q=info:LGUG3U6Bp-kJ:scholar.google.com/&output=citation&scisdr=CgUx7uJgEJT6uf13NIw:AAGBfm0AAAAAY2txLizGMVUpPg6_ihwN2dU8PlulpMG-4&scisig=AAGBfm0AAAAAY2txLO2TG0gZXx2xxpt2AfWUrr5KIziD&scisf=4&ct=citation&cd=-1&hl=en)). pp. 472–481. doi:10.1145/3209108.3209165 (<https://doi.org/10.1145%2F3209108.3209165>). ISBN 9781450355834. S2CID 17887510 (<https://api.semanticscholar.org/CorpusID:17887510>).
14. Coecke, Bob; Spekkens, Robert W (2012). "Picturing classical and quantum Bayesian inference" (<https://scholar.googleusercontent.com/scholar.bib?q=info:eNWS7luaOWYJ:scholar.google.com/&output=citation&scisdr=CgUx7uJgEJT6uf2luqs:AAGBfm0AAAAAY2uOoqt0qaPzhZqEbOoYRpydAyZvbF0g&scisig=AAGBfm0AAAAAY2uOomzjHno-ePkhrJB3WfamgsIs3Jwg&scisf=4&ct=citation&cd=-1&hl=en>). *Synthese*. **186** (3): 651–696. arXiv:1102.2368 (<https://arxiv.org/abs/1102.2368>). doi:10.1007/s11229-011-9917-5 (<https://doi.org/10.1007%2Fs11229-011-9917-5>). S2CID 3736082 (<https://api.semanticscholar.org/CorpusID:3736082>).
15. Signorelli, Camilo Miguel; Wang, Quanlong; Coecke, Bob (2021-10-01). "Reasoning about conscious experience with axiomatic and graphical mathematics" (<https://doi.org/10.1016%2Fj.concog.2021.103168>). *Consciousness and Cognition*. **95**: 103168. doi:10.1016/j.concog.2021.103168 (<https://doi.org/10.1016%2Fj.concog.2021.103168>). hdl:10230/53097 (<https://hdl.handle.net/10230%2F53097>). ISSN 1053-8100 (<https://search.worldcat.org/issn/1053-8100>). PMID 34627099 (<https://pubmed.ncbi.nlm.nih.gov/34627099>). S2CID 235683270 (<https://api.semanticscholar.org/CorpusID:235683270>).
16. Fritz, Tobias (August 2020). "A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics". *Advances in Mathematics*. **370**: 107239. arXiv:1908.07021 (<https://arxiv.org/abs/1908.07021>). doi:10.1016/j.aim.2020.107239 (<https://doi.org/10.1016%2Fj.aim.2020.107239>). S2CID 201103837 (<https://api.semanticscholar.org/CorpusID:201103837>).
17. Bonchi, Filippo; Sobociński, Paweł; Zanasi, Fabio (September 2014). "A Categorical Semantics of Signal Flow Graphs" (<https://hal.archives-ouvertes.fr/hal-02134182>). *CONCUR 2014 – Concurrency Theory*. Lecture Notes in Computer Science. Vol. CONCUR 2014 - Concurrency Theory - 25th International Conference. Rome, Italy. pp. 435–450. doi:10.1007/978-3-662-44584-6\_30 ([https://doi.org/10.1007%2F978-3-662-44584-6\\_30](https://doi.org/10.1007%2F978-3-662-44584-6_30)). ISBN 978-3-662-44583-9. S2CID 18492893 (<https://api.semanticscholar.org/CorpusID:18492893>).
18. Bonchi, Filippo; Seeber, Jens; Sobocinski, Paweł (2018-04-20). "Graphical Conjunctive Queries". arXiv:1804.07626 (<https://arxiv.org/abs/1804.07626>) [cs.LO (<https://arxiv.org/archive/cs/LO>)].
19. Riley, Mitchell (2018). "Categories of optics". arXiv:1809.00738 (<https://arxiv.org/abs/1809.00738>) [math.CT (<https://arxiv.org/archive/math/CT>)].

## External links

---

- TheCatsters (2007). *String diagrams 1* (<https://www.youtube.com/watch?v=USYRDDZ9yEc>) (streamed video). Youtube. Archived (<https://ghostarchive.org/varchive/youtube/20211219/USYRDDZ9yEc>) from the original on 2021-12-19.
- String diagrams (<https://ncatlab.org/nlab/show/string+diagram>) at the *nLab*

- [DisCoPy \(https://discopy.readthedocs.org/\)](https://discopy.readthedocs.org/), a Python toolkit for computing with string diagrams

## External links

---

-  Media related to [String diagram](#) at Wikimedia Commons
- 

Retrieved from "[https://en.wikipedia.org/w/index.php?title=String\\_diagram&oldid=1222282077](https://en.wikipedia.org/w/index.php?title=String_diagram&oldid=1222282077)"