

Box of Strings: A String Diagram Rewriting Tool

(Software Demonstration)

Niels Voorneveld*

Cybernetica AS
Tallinn, Estonia

niels.vorneveld@cyber.ee

The *Box of Strings* tool prototype is a program in which you can visualize string diagrams and apply rewrites to them [1]. The tool can load in any equational theory on a monoidal categories generated by a single object and some finite signature. It displays morphisms as string diagrams, with the strings denoting objects in the category and boxes denoting atomic operations from the signature [2]. The tool is meant to enable you to look at definitions, manipulate string diagrams and prove lemmas. As such, it is intended to work alongside scientific papers, helping to understand them as well, as well as to develop and verify new results.

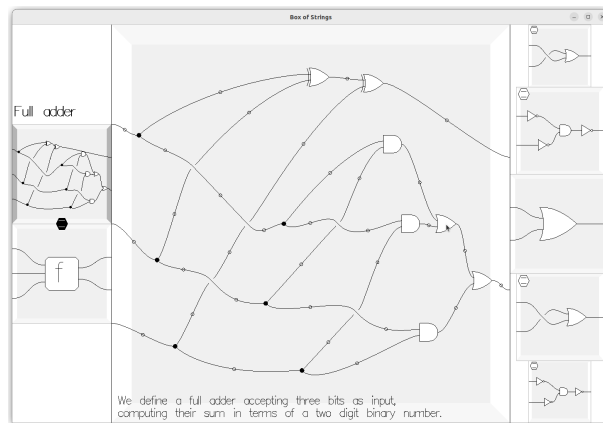


Figure 1: Screenshot of *Box of Strings*, displaying the definition of a full adder in Boolean circuits, using standard circuit gate representation. Currently highlights two rewrites for the or-operation.

Rewriting in the tool: Often many rewrites apply to a particular morphism. The tool aims to cut down on this potential *combinatorial explosion* of possibilities by only showing exact matches of rewrites to the currently selected node. The user groups together nodes to find matching rewrites, and then chooses one to apply. The tool uses mouse controls to attempt to give an intuitive way of doing this. Rewriting one string diagram into another is done by performing the following three actions in sequence:

1. Holding the *left mouse button* allows you to drag around and collect operations into box representing an operation consisting of some composite morphism.
2. Hovering over an operation, composite or not, displays possible rewrites on the right hand side of the screen. Use the *mouse wheel* (or arrow keys) to cycle through the possible rewrites.
3. Clicking with the *right mouse button* on a box unfolds its contents, effectively applying the rewrite.

*This research was supported by ARIA UK under the Safeguarded AI programme.

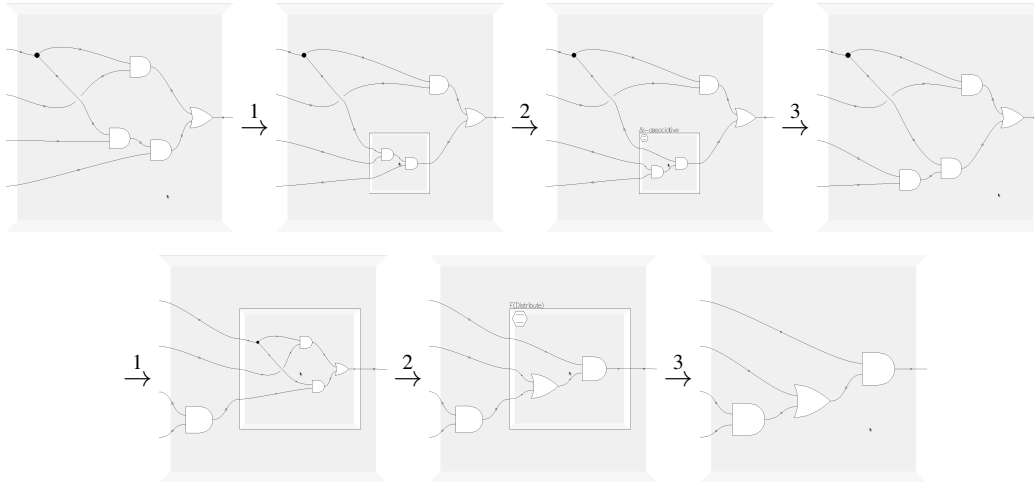


Figure 2: A sequence of two rewrites in the tool, following the three steps.

Data Structure of the tool: Box of strings takes a combinatorial approach to string diagrams, representing morphisms as explicit sequences of operations. This has several benefits. First of all, the representation allows for easy drawing of string diagrams, with the elements more naturally spaced out. Secondly, it makes it theoretically straightforward to group elements together and unfold them. Thirdly, the representation cuts down on a lot of categorical equations, only needing to work with the interchange law explicitly. Equality between morphisms is checked by reducing to some canonical form, which prioritizes alternating between parallel processes. This equality check is then used to check for matches in the rewrite theory.

$$\frac{a \in \mathbb{N}}{id_a : a \rightarrow b} \quad \frac{m : a \rightarrow b \quad (\text{op} : c \rightarrow d) \in \Sigma \quad (i + c + j) = b}{m \triangleright (id_i \otimes \text{op} \otimes id_j) : a \rightarrow (i + d + j)}$$

The tool has the following additional features.

- A selection of axiom schema, which automatically adds rewrites when new operations and rewrites are added, including:
 - Symmetric monoidal schema for dragging an operation across a wire.
 - Naturality of operations, with respect to all or some subselection of operations.
 - Symmetries on rewrites, such as flipping and mirroring.
- Defining new composite operations, and zooming into composite boxes.
- Partial order enriched categories, using directed rewrites and antisymmetry.

In the talk, we shall go over some of the features of the tool, and explore different categorical theories. The tool has been written in Haskell, with vector drawing done using OpenGL. A version of the tool is found at <https://github.com/Voorn/Box-of-Strings>.

References

- [1] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski & Fabio Zanasi (2020): *String Diagram Rewrite Theory I: Rewriting with Frobenius Structure*. CoRR abs/2012.01847. arXiv:2012.01847.
- [2] P. Selinger (2010): *A Survey of Graphical Languages for Monoidal Categories*, p. 289–355. Springer Berlin Heidelberg, doi:10.1007/978-3-642-12821-9_4.