

Widip: An Open-source Computing System for String Diagrams

Subjects: Programming Languages (cs.PL); Logic in Computer Science (cs.LO); Mathematical Software (cs.MS)

Author: Martin Coll, Buenos Aires

Abstract

We present the Widip Open-source Computing System for String Diagrams. We motivate our work in their equivalence as syntax with the Cospans of Hypergraphs computing structures and its relation to the LISP Programming System.

Introduction

In String Diagram Rewrite Theory [BGK+] we see *symbolic computation* from a new perspective. The authors show that the syntax of **String Diagrams** can be regarded as computation under the **Rewrite Theory of Cospans of Hypergraphs**.

The LISP Programming System [McC] modelled a similar framework for the AI group at MIT. An important decision was to define programs in the same class of symbolic expressions, as that “has advantages both as a programming language and as vehicles for developing a theory of computation”.

Implementation

We implement the `widip` program with the `discopy` toolkit presented in DisCoPy: Monoidal Categories in Python [FTC]. In [BGK+] the authors present diagrams together with TeX equations and in [FTC] there is an imperative Python API. It is not clear how a system built around TeX or Python files can support a LISP-based approach. Two-dimensional diagrams present a challenge for the syntax design.

We chose the YAML data language [YAML] to model such structures for several practical reasons but we will focus on the aspects that make it applicable to modelling diagrams. YAML is a popular data language that makes it possible to write friendly yet complex data structures made of recursive sequences, mappings and scalars. Documents are modelled as a Presentation Stream and an equivalent combinatorial structure called Representation Graph.

There is a straightforward transformation from the representation graph of a YAML document to Cospans of Hypergraphs. in Python using `discopy` for computing and drawing image files. In the following diagram the box named “widip” is our implementation:

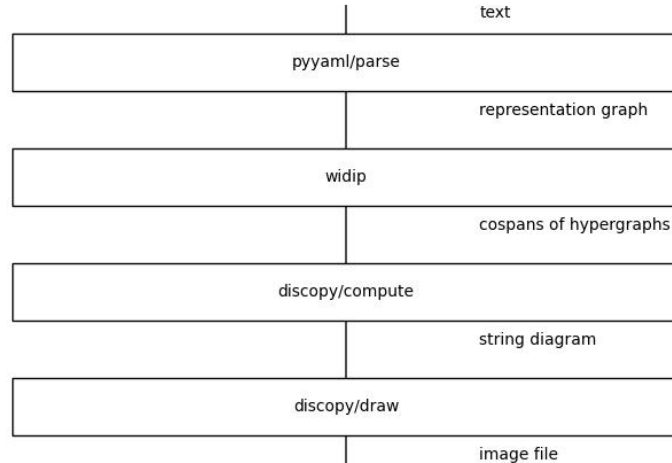


Figure 1: The `widip` Python implementation diagram

We find it remarkable that the equivalent Presentation Stream, namely the YAML document below, closely resembles the string diagram two-dimensional image. We believe this is evidence of the deep roots of the Equivalence along with the long-lasting contributions of LISP.

```

- !pyyaml/parse text: representation graph
- !widip representation graph: cospans of hypergraphs
- !discopy/compute cospans of hypergraphs: string diagram
- !discopy/draw string diagram: image file

```

Future work

We would like to implement the Run language from Programs as Diagrams [Pav] as part of the Widip computing system. We are also interested in the Catlab package discussed in [PSV] for compiling to native code.

References

- [FTC] Giovanni de Felice, Alexis Toumi, Bob Coecke. DisCoPy: Monoidal Categories in Python. EPTCS 333, 2021, pp. 183-197.
- [BGK+] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. String Diagram rewrite theory I: Rewriting with Frobenius structure. Journal of the ACM, 69(2):14:1–14:58, 2022.
- [YAML] YAML Language Development Team. YAML Ain’t Markup Language. Revision 1.2.2 (2021-10-01).

- [McC] John McCarthy. Recursive functions of symbolic expressions and their computation by machine. *Communications of the ACM*, Volume 3, Issue 4, pp 184–195.
- [Ing] Daniel Ingalls. Design Principles Behind Smalltalk. *BYTE Magazine*, August 1981.
- [Pav] Dusko Pavlovic. Programs as Diagrams: From Categorical Computability to Computable Categories.
- [RT] Dennis Ritchie, Ken Thompson. The UNIX Time-Sharing System. *Fourth ACM Symposium on Operating Systems Principles*. October 15–17, 1973.
- [PSV] Evan Patterson, David Spivak, Dmitry Vagner. Wiring diagrams as normal forms for computing in symmetric monoidal categories. *Applied Category Theory 2020 (ACT2020)*. *EPTCS 333*, 2021, pp. 49–64, doi:10.4204/EPTCS.333.4