

1153-18-378

Dmitry Vagner* (dmitryvagner@gmail.com). *Defining and Programming Generic Compositions in Symmetric Monoidal Categories.*

We define a class of wiring diagrams called *flows*, which correspond to generic compositions in strict symmetric monoidal categories. Said otherwise, any string diagram in a symmetric monoidal category $(\mathcal{V}, \otimes, I)$ can be realized by plugging-in \mathcal{V} -arrows into some flow. In fact, if we define objects to be *boxes*, i.e. input slots for arrows, then flows are the morphisms of a typed operad whose algebras are symmetric monoidal categories.

We then turn our attention to exploring this formalism in the programming language Idris, a Haskell-like language with native dependent types. In particular, there is—roughly speaking—a category whose objects are Idris types and morphisms are functional programs between them. We present an Idris implementation of this category as a flow-algebra. (Received September 02, 2019)