Toward a Formalization of 'Virtual Sets'

School of Computer Science, University of Nottingham

Christina O'Donnell
Under supervision of Thorsten Altenkirch

September 2025

Overview

The aim of this project was to formally construct a structure called a 'Traced Monoidal Category' around the notion of injective functions on finite sets. And implementing the Int construction to explore a notion of 'negative sets'.

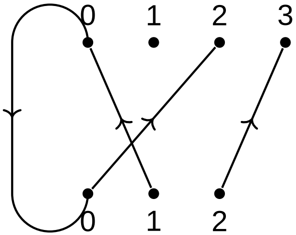


Figure: $(2\ 0\ 3)$

Objectives

- Construct the category of finite sets and injective in Agda
- $(\oplus)$ Define a tensor produt and prove coherence
- $(\ominus)$ Define a trace operation and prove coherence
- Formalize the **Int**-construction to produce virutal sets

Methodology

- Written in Cubical Agda, a proof assistant and implementation of homotopy type theory.
- Two different representations of injective functions were used: depdendent sum representation, and inductive representation.
- Work was performed between June and September, and involved the production of 5.6k SLoC accross 74 files with 340 definitions.

Category Construction

- A category is an abstract structure with many examples.
- Many examples come from algebraic objects and structure preserving maps called *morphisms*.
- We use the following mapping:
  - **objects** are natural numbers ($\mathbb{N}$) representing each finite set size.
  - **morphisms** are injective maps between finite sets, defined inductively.
  - **composition** comes from joining two function graphs in a way that preserves application.
  - **identity** is just the graph that maps all elements to themselves.
  - **associativity** and **left/right-unit** laws also hold.

Tensor Product ($\oplus$)

Components of the tensor product:

- object level: arithmetic +
- morphism level: split $\rightarrow$ apply $\rightarrow$ join (see next slide)
- unit object: $\mathbb{0} = 0$
- preserves identity: *id* $\oplus$ *id* $=$ *id*
- preserves composition: $(f \circ f') \oplus (g \circ g') = (f \oplus g) \circ (f' \oplus g')$
- associator: $\alpha_{A,B,C} : A \oplus (B \oplus C) \cong (A \oplus B) \oplus C$.
- left-unitor: $\eta_A : \mathbb{0} \oplus A = A$
- right-unitor: $\rho_A : A \oplus \mathbb{0} = A$
- coherence laws: Show result of composition is indepdendent of order

Tensor Product ($\oplus$): Example

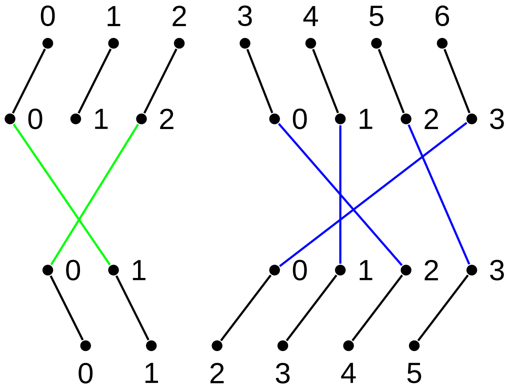
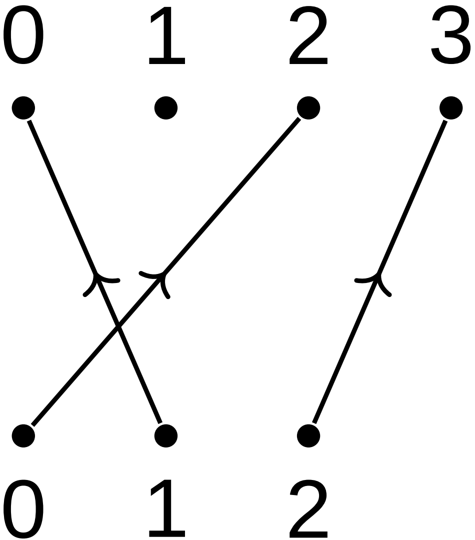


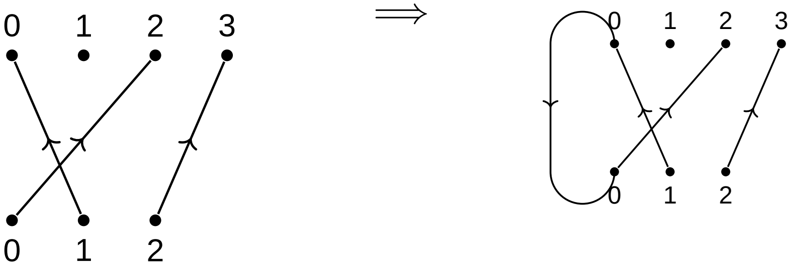
Figure: $(2\ 0) \oplus (3\ 1\ 0\ 2) = (2\ 0\ 6\ 4\ 3\ 5)$

Trace ($\ominus$): By example

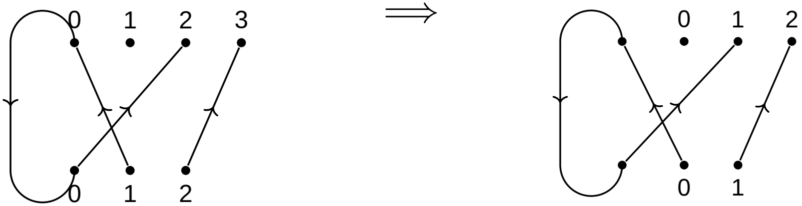
We will construct $\mathrm{tr}_1((2\ 0\ 3))$:



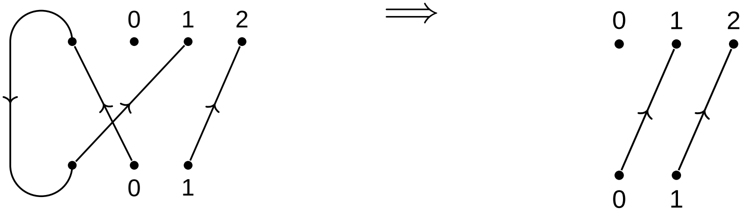
Trace (⊖): Example: Step 1 Add Loop



Trace ($\ominus$): Example: Step 2 Shift indices



Trace ($\ominus$): Example: Step 3 Join directly



Trace ($\ominus$): Definition (part 2)

Let $f : [X + A \rightarrowtail X + B]$
Then $f \ominus X : [A \rightarrowtail B]$, Such that the following properties are satisfied:

- Vanishing

- Superimposing

$$(\forall A, B, X : \mathbb{N})$$

Trace ($\ominus$): Definition (part 3)

Let $f : [X + A \rightarrowtail X + B]$
Then $f \ominus X : [A \rightarrowtail B]$, Such that the following properties are satisfied:

- Tightening

- Sliding

$$(\forall A, B, X : \mathbb{N})$$

Results

- Constructed the category of injective functions Inj.
- Defined a tensor product, showing that it preserves identity and composition
- Defined a trace operation
- Proven ’vanishing’ on the trace.
- Partial construciton of monoidal category
- Partial proof of superimposing axiom

Future Work

- Finish completion of monoidal construction
- Express the remaining trace axioms formally
- Complete the Int-construction
- ’Upstream’ results to Agda Cubical library.

Thank you

Thank you for listening