A diagram editor to mechanise categorical proofs

Ambroise Lafont

JFLA, 30 January 2024
Packaging

A standalone desktop program

A web app that runs locally in your browser (without mechanisation features)

https://amblafont.github.io/graph-editor/index.html
Naming convention

Yet Another Diagram Editor

(not to be confused with Yet Another Dynamical Engine!)

I will refer to the editor as **YADE**, or **Coreact-YADE**

ANR Project¹ (2023 - 2027): Coq-based Rewriting: Towards Executable Applied Category Theory

¹ [https://coreact.wiki/](https://coreact.wiki/)
Related software: Quiver

“a modern, graphical editor for commutative and pasting diagrams, capable of rendering high-quality diagrams for screen viewing, and exporting to LaTeX via tikz-cd.”
Comparison with quiver

About the same size (around 10k of LoC)

<table>
<thead>
<tr>
<th></th>
<th>Quiver</th>
<th>YADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Languages</td>
<td><img src="image" alt="Languages" /></td>
<td><img src="image" alt="Languages" /></td>
</tr>
<tr>
<td></td>
<td><strong>Languages</strong></td>
<td><strong>Languages</strong></td>
</tr>
<tr>
<td></td>
<td>JavaScript 90.7%</td>
<td>Elm 72.6%</td>
</tr>
<tr>
<td></td>
<td>CSS 5.3%</td>
<td>HTML 12.7%</td>
</tr>
<tr>
<td></td>
<td>TeX 2.4%</td>
<td>TeX 8.5%</td>
</tr>
<tr>
<td></td>
<td>Other 1.6%</td>
<td>TypeScript 4.7%</td>
</tr>
<tr>
<td></td>
<td>JavaScript 1.5%</td>
<td>JavaScript 1.5%</td>
</tr>
<tr>
<td>Styling options</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>User-friendly</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Editing features</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Tabs, copy &amp; paste, find &amp; replace, expand selection to connected components, ...</td>
<td></td>
</tr>
<tr>
<td>LaTeX export</td>
<td>yes</td>
<td>yes¹</td>
</tr>
</tbody>
</table>

¹ Implemented by Tom Hirschowitz
Comparison with quiver

About the same size (around 10k of LoC)

<table>
<thead>
<tr>
<th>Programming Languages</th>
<th>Quiver</th>
<th>YADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Styling options</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>User-friendly</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Editing features</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>LaTeX export</td>
<td>yes</td>
<td>yes(^1)</td>
</tr>
<tr>
<td>Mechanisation features</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

\(^1\) Implemented by Tom Hirschowitz
Architecture

YADE (standalone version)

- Show diagram under cursor
- Generate proofs

Visual Studio Code

- Custom vscode extension (building upon coq-lsp)

(+ Coq library for custom notations)
Natural transformations compose:

\[ 
\begin{align*}
     &F\alpha_c \circ Ff 
     &= \beta_c \circ \alpha_c 
     &= \beta_c \circ \beta_c \circ \alpha_c 
     = \beta_c \circ \alpha_c \circ Ff \\
     &G\beta_c \circ Gf 
     &= \alpha_c 
     &= \beta_c \circ \alpha_c \circ Gf 
     = \beta_c \circ \alpha_c \circ Ff 
\end{align*}
\]

Diagrammatic proof

Automatic generation?

Computer-friendly proof
Diagram editor (standalone version)

Architecture

Visual Studio Code

Custom vscode extension (building upon coq-lsp)

(+ Coq library for custom notations)
Building the diagrammatic proof interactively

1) Select a subdiagram

2) Create a proof node, labelled with the Coq tactic naturality.

⇒ Coq (in vscode) checks that this tactic solves the goal:

\[ Hf \circ n_a = n_b \circ _- \]

⇒ The diagram gets completed in YADE:

• The **unnamed arrow** is refined by Coq’s inferred instantiation
• The proof node is marked as validated (indicated by a green ✓)
Demo\(^1\) of YADE

(Based on the category theory library of Hierarchy Builder + custom tactics & notations)

A distributive law \(\delta: TS \Rightarrow ST\) between two monads \(S\) and \(T\) induces a monad structure on \(ST\).

Let us show that the induced multiplication \(STST \xrightarrow{\delta^{ST}} SSTT \xrightarrow{\mu^S \mu^T} ST\) is associative.

\(^1\) https://github.com/amblafont/vscode-yade-example