

Automatic Generation of Commutative Diagrams from Category-theoretic Propositions in Agda

MWPLS '24

Kelton OBrien

November 22, 2024

Why Categories?

- ▶ What is Category theory?

Why Categories?

- ▶ What is Category theory?
- ▶ (elided)

Why Categories?

- ▶ What is Category theory?
- ▶ (elided)
- ▶ (Formal) mathematics, denotational semantics, homotopy type theory

The Problem

- ▶ Setup: We are doing category theory in a proof assistant (agda)

The Problem

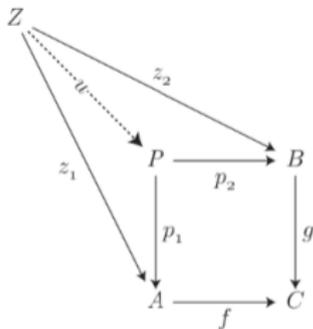
- ▶ Setup: We are doing category theory in a proof assistant (agda)
- ▶ Want to do reasoning closer to pen-and-paper while still being verified

The Problem

- ▶ Setup: We are doing category theory in a proof assistant (agda)
- ▶ Want to do reasoning closer to pen-and-paper while still being verified
- ▶ What does pen-and-paper category theory look like?

Math

$z_1 = p_1 u$ and $z_2 = p_2 u$. The situation is indicated in the following diagram:



Agda

```
record is-pullback {P} (p1 : Hom P X) (f : Hom X Z) (p2 : Hom P Y) (g : Hom Y Z)
  : Type (o ⊔ ℓ) where

no-eta-equality
field
  square      : f ∘ p1 ≡ g ∘ p2
  universal   : ∀ {P'} {p1' : Hom P' X} {p2' : Hom P' Y}
    → f ∘ p1' ≡ g ∘ p2' → Hom P' P
  p1∘universal : {p : f ∘ p1' ≡ g ∘ p2'} → p1 ∘ universal p ≡ p1'
  p2∘universal : {p : f ∘ p1' ≡ g ∘ p2'} → p2 ∘ universal p ≡ p2'

  unique      : {p : f ∘ p1' ≡ g ∘ p2'} {lim' : Hom P' P}
    → p1 ∘ lim' ≡ p1'
    → p2 ∘ lim' ≡ p2'
    → lim' ≡ universal p
```

What is to be done?

- ▶ Go totally graphical!

What is to be done?

- ▶ Go totally graphical!
- ▶ Nothing at all!

What is to be done?

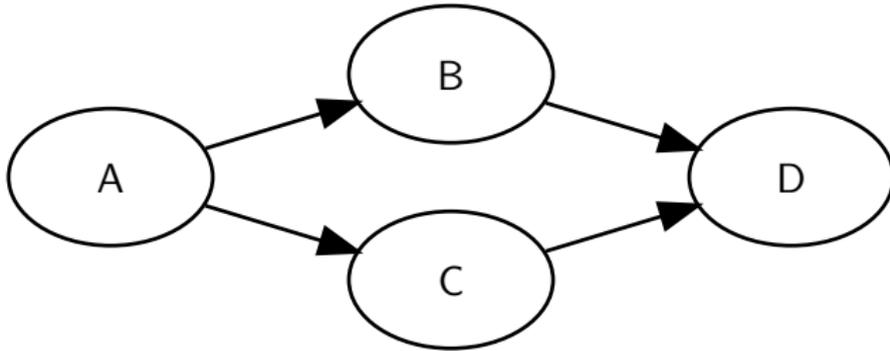
- ▶ Go totally graphical!
- ▶ Nothing at all!
- ▶ middle ground: generate diagrams (based on some proposition)

What do we want?

- ▶ Why not just pop everything into dot?

What do we want?

- ▶ Why not just pop everything into dot?
- ▶ Does not have the *shape* we want



What do we want?

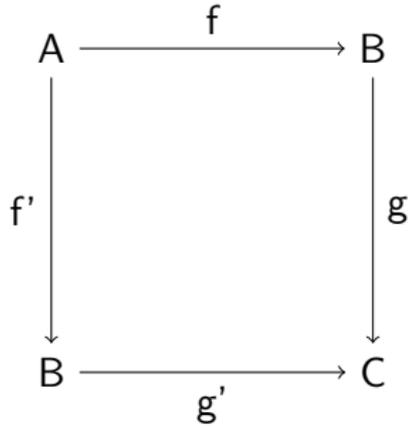
- ▶ Why not just pop everything into dot?
- ▶ Does not have the *shape* we want
- ▶ Automatic graphing: well studied for different constraints

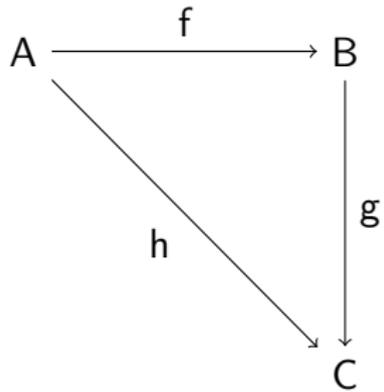
Shaping up

- ▶ 3 fundamental shapes: Square, Triangle, parallel lines

Shaping up

- ▶ 3 fundamental shapes: Square, Triangle, parallel lines
- ▶ Correspond to particular equalities of morphism





Approach

- ▶ Choose a category theory library to extend (1Lab)

Approach

- ▶ Choose a category theory library to extend (1Lab)
- ▶ Reflect

Approach

- ▶ Choose a category theory library to extend (1Lab)
- ▶ Reflect
- ▶ Shape

Approach

- ▶ Choose a category theory library to extend (1Lab)
- ▶ Reflect
- ▶ Shape
- ▶ Graph

Approach

- ▶ Choose a category theory library to extend (1Lab)
- ▶ Reflect
- ▶ Shape
- ▶ Graph
- ▶ Bonus: easy switching between library

Reflection

- ▶ want to graph, e.g, $g \circ f \equiv g' \circ f'$

Reflection

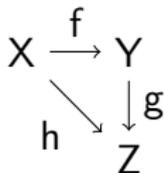
- ▶ want to graph, e.g, $g \circ f \equiv g' \circ f'$
- ▶ Note the lack of information!

Reflection

- ▶ want to graph, e.g, $g \circ f \equiv g' \circ f'$
- ▶ Note the lack of information!
- ▶ Agda has all of this in the types!

(Begin Micro-Demo)

Triangle



Square

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ f' \downarrow & & \downarrow f' \\ Y & \xrightarrow{g'} & Z \end{array}$$

Functors

$$\begin{array}{ccc} & Gf & \\ GX & \rightarrow & GY \\ Gf' \downarrow & & \downarrow Gf' \\ GY & \xrightarrow{Gg'} & GZ \end{array}$$

The Hard Part: pasting

- ▶ Nice to be able graph more than proposition...

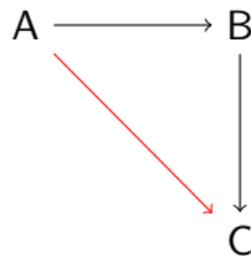
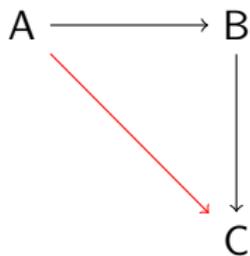
The Hard Part: pasting

- ▶ Nice to be able graph more than proposition...
- ▶ this means merging graphs!

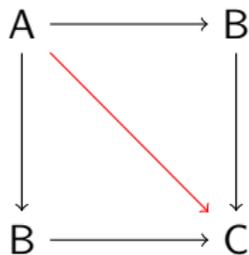
The Hard Part: pasting

- ▶ Nice to be able graph more than proposition...
- ▶ this means merging graphs!
- ▶ Determining how this should look is very very hard!

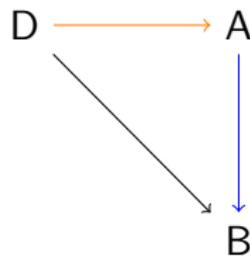
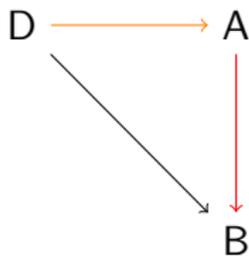
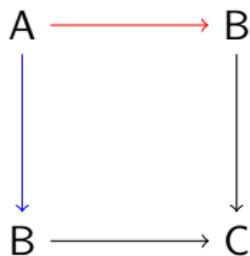
A problem case



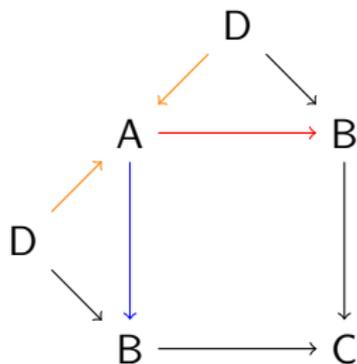
A problem solved

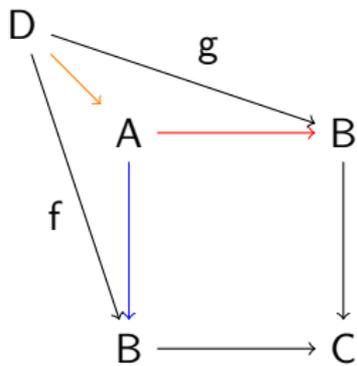


A problem case: 2



The method





Punchline

This is very hard to do in full generality, and becomes a very particular geometric problem!

Thanks!