

BEWL UPDATE



BEGINNING CALCULATIONS WITH MUSICAL OBJECTS

WARNING: ABSTRACT MATH AHEAD



BUT IT WON'T HURT ALL THAT MUCH



STORY SO FAR

- ❖ Paper “The Topos of Triads” (Thomas Noll, 2005)
- ❖ Shows how to analyse musical compositions using topos theory
- ❖ People have used it to compose music, too
- ❖ This looks like a job for... Bewl.
- ❖ But: too slow



Things seemed
hopeless for a
while, but then
inspiration struck.




```

import com.fdilke.bewl.helper.⊕
import com.fdilke.bewl.topos._

import scala.language.{higherKinds, postfixOps,
reflectiveCalls}

trait ConstructDefaultMonoidAssistant extends
  BaseTopos with
  ToposEnrichments {

   $\mathcal{E}$ : ToposPrerequisites =>

  trait MonoidAssistant {
    def actionAnalyzer[
      M <: ~
    ] (
      monoid: Monoid[M]
    ): monoid.ActionAnalyzer
  }

  object DefaultMonoidAssistant extends
    MonoidAssistant {
      // ...
    }
  }

```



- ❖ Skipping lightly over the details, it's faster now.

- ❖ Bewl can now model the octave, chord of C major, triadic monoid, and other ‘musical objects’ described in Noll’s paper.



- ❖ In fact, there seems no reason why it can’t do all the calculations he does, although this is a bold claim that would be all the better for proof.
- ❖ This ““““real-life application”””” will drive more optimisations and enhancements to Bewl.
- ❖ So, let’s do some calculations with the chord.

IT'S ONLY 3 NOTES :
SMALL BUT
PERFECTLY FORMED

SO WE EXPECT:



-
- ❖ It's *minimal* (no subobjects other than itself and 0)
 - ❖ It's *simple* (no homomorphic images but itself and 1)
 - ❖ More ambitiously, it is probably not *injective* (algebraically closed)

PUTTING THE CHORD OF C MAJOR UNDER THE (TOPOS-THEORETIC) MICROSCOPE

chord.isMinimal

true

chord.isSimple

...Err, the relational
algebra module needs
work.

chord.isInjective

(an hour later) No. But
I do have good news.

AN OBJECT IS INJECTIVE IF YOU DON'T REALLY NEED TO EXTEND IT

- ❖ Example: the long history of extending the natural numbers $\mathbf{N} = \{ 1, 2, 3, \dots \}$ to build bigger and badder number systems.
- ❖ Natural numbers $\mathbf{N} \leq$ integers $\mathbf{Z} \leq$ rationals $\mathbf{Q} \leq$ reals $\mathbf{R} \leq$ complex numbers \mathbf{C}
- ❖ For many purposes, \mathbf{C} is as complete a workspace for calculation as you'll ever need
- ❖ Codified in the “Fundamental Theorem of Algebra” (1806)

The chord does NOT have this property.

ANOTHER EXAMPLE OF AN INJECTIVE: THE UNIT INTERVAL

- ❖ The *unit interval* $\mathbf{I} = [0, 1]$ is injective in a (suitably chosen) category of topological spaces.
- ❖ It's so injective that any space* can be embedded in a power of it, by multiplying up arrows into \mathbf{I} that collectively separate the space. Powers of \mathbf{I} look like the cube, but in more dimensions.
- ❖ This is like embedding a spider in plastic.
- ❖ Maybe a similar structure theory applies to musical objects! That would be fun wouldn't it.

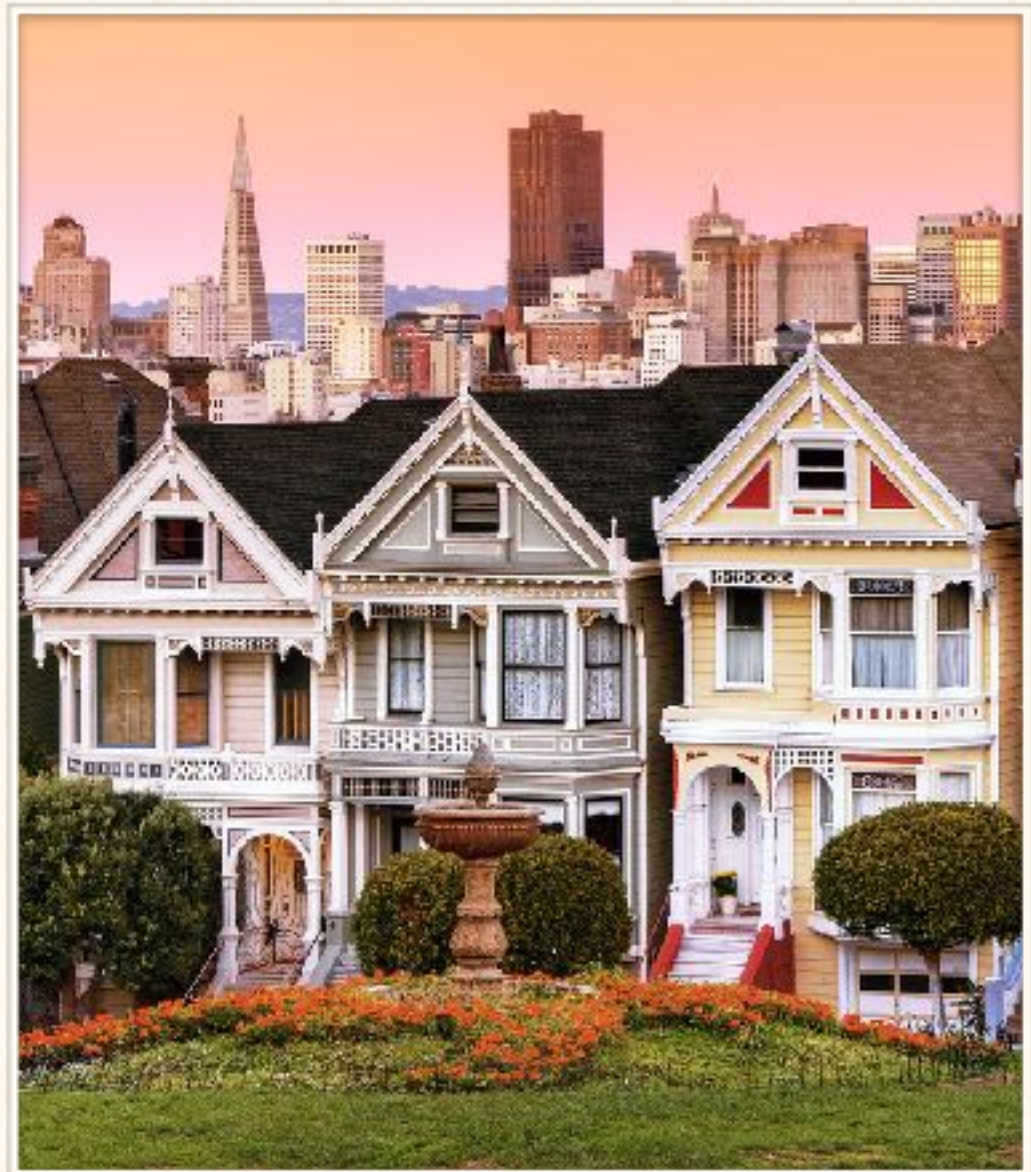
*Terms and conditions apply



- ❖ In a category (= an abstract collection of dots and arrows), a dot (object) is *injective* if there are lots of arrows into it. More precisely, arrows into it can be extended along any monic.
- ❖ In some categories, there is an embedding for any object into a unique smallest injective one, called its *injective hull*. This is like a very abstract version of extending the reals, \mathbf{R} , to the complexes, \mathbf{C} .
- ❖ The musical objects have this property, so there is an injective hull of the chord! It has size < 1300 and I look forward to modelling it.

NEW VISTAS OPEN UP

- ❖ In a *locally finite topos*, i.e. anything Bewl could reasonably model, there is a relatively simple construction generalizing this. So there can be a generic, built-in method in Bewl to calculate the injective hull.
- ❖ This little discovery seems to be new, so I'm writing it up as a maths paper, using Overleaf. (The hard bit is the typesetting.) Maybe a Springer journal will publish it!



THANK YOU

<http://github.com/fdilke/bewl>