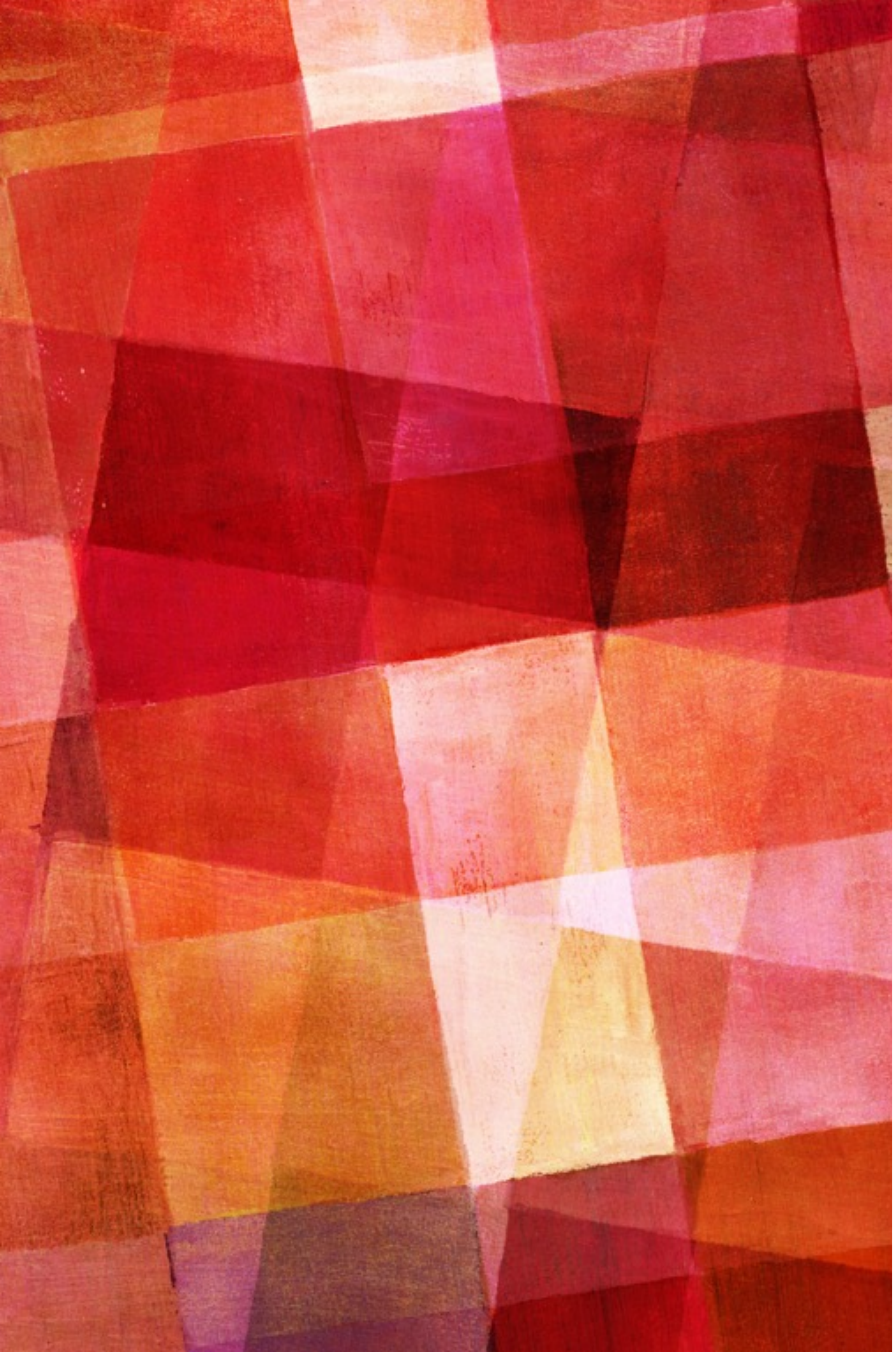
THE TOPOS OF AUTOMORPHISMS

An attempt to explain the deep mystery of permutation parity

BEWL ORIGIN STORY

- The original motivation for Bewl was to investigate the unexplained phenomenon of *permutation parity*
- This is a concept from elementary math
- So I had to learn about category theory, topos theory, and abstract types in Scala
- The project took on a life of its own, as a language for talking about all kinds of objects as if they were sets
- But now it turns out that permutations form a topos. So you can use Bewl to talk about them as if they were sets
- No one seems to have done this before!



PERMUTATIONS

- A *permutation* is just a rearrangement of some finite set of objects.
- Example: on the Honeybee team, swap Cris with Abdul and rotate James -> Simon -> Felix -> James.
- You can multiply and divide permutations, and there's a 1 (the do-nothing permutation which leaves everything fixed). So they form a little arithmetic.

- Every permutation has a *parity*, +1 or -1.
- So, there's a concept of *even* and *odd* permutations.
- even + even = even, odd + odd =even, even + even = odd
- It's as if every finite set had an *orientation* or *handedness*: the permutation either flips this over, or else it doesn't.
- But, there is no decent explanation of this. It's some sort of deep unexplained property of sets.

CALCULATING PARITY

- Swapping Cris with Abdul and rotate James -> Simon -> Felix -> James:
- $$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$
  $$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$
  $$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$
  $$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$
  $$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$
- Compute the determinant: -1
- So now we understand parity, right?

NO, WE DON'T

To develop the theory of determinants in the first place, we have to take parity as an unexplained ingredient

- Idea: what would it mean to have a mapping from one permutation to another?
- ```
  X   —- f —-> Y
  |            |
  p            q
  |            |
  V            V
  X   —- f —-> Y
  ```

- So now we have a category, **Perm** which turns out to be a topos. So you can use all the Bewl DSL machinery.
  ```

THE TOPOS OF AUTOMORPHISMS

- Permutations are just invertible arrows from some object to itself in the category of finite sets
- i.e. *automorphisms* of finite sets
- **Perm = Aut(FinSet)**
- The category of automorphisms, **Aut C** can be defined for any category **C**
- **Aut** is a monad on the category **Cat** of categories
- If **C** is a topos, so is **Aut C**
- Obviously, **Aut** should be a construction in Bewl! And now it is.