# Coinduction and topology:
# an unexpected connection

Wojciech Kołowski

17 June 2020

# Review of approaches to coinduction

- So far we have seen two approaches to coinduction and bisimulation:

- The LTS approach, in which a coinductive process is (re)presented using a particular kind of machine. This machine can be in any of a number of states and can transition between them by performing an appropriate action.

- The categorical approach, in which we are interested in coalgebras of endofunctors. In the final coalgebra ($\nu F$, **unfold**) the "coinductively" defined object is the carrier $\nu F$, and **unfold** takes the object apart, splitting into its constituent parts (it can also be seen as performing some observations on $\nu F$). Corecursion is a consequence of finality.

## The LTS approach: a critique 1/2

- I think that the LTS approach to coinduction and bisimulation is quite bad from the explanatory point of view, for a few reasons:

- First, it obscures the very important duality between induction and coinduction, which everybody wants to learn about instantly upon seeing the name "coinduction".

- Interlude: the right notion of equality for LTSes is of course graph isomorphism, as they are nothing more than labeled graphs – static, immobile objects that prescribe actions and transitions, but don't act and don't transition.

## The LTS approach: a critique 2/2

- Second, the idea of bisimulation is a bit ad hoc and circular.
- Bisimulation was advised in the book as the right notion of behavioural equality of LTSes, but it is in fact the right notion of equality for behaviours of LTSes. The behaviour of an LTS is its dynamic aspect – where the actions and transitions take place.
- However, to define it formally, we need coinduction (or else we will miss "infinite" phenomena). Thus, there is some kind of circularity in explaining coinduction using LTSes, even if only conceptual.

# The categorical approach: a critique

- The categorical approach is much better, as it makes the duality between induction and coinduction more explicit and also doesn't give the false impression that coinduction is about automata.

- However, it is not without faults:

- By using the machinery of category theory it makes coinduction seem more magical and arcane than it really is. It is unlikely to be enlightening to ordinary programmers and people with category theory disability.

- It makes the operational and computational aspects of corecursion less explicit.

- It does not provide a nice syntax/notation for corecursive definitions (even though it does provide $\nu X.F(X)$ for objects) – and that's very important! "Notation is the tool of thought", they say.

# The duality

| Feature | induction | coinduction |
|---|---|---|
| shape | sum of products | product of sums |
| polarity | positive | negative |
| basic activity | construction | deconstruction (observation) |
| derived activity | deconstruction (observation) | construction |
| tree width | any | any |
| tree height | necessarily finite | possibly infinite |
| evaluation | possibly eager | necessarily lazy |
| easy to define functions with | inductive domain | coinductive codomain |
| every step | shrinks the principal argument | grows the result |