Monoidal Width

joint work with Elena Di Lavore

Pawel Sobocinski, Tallinn University of Technology, Estonia
Structure meets Power 2022, Paris, 4 July 2022

Euroopa Liit
Euroopa Sotsiaalfond

Eesti
tuleviku heaks

What is a graph width?

- a function Graphs → **N**, i.e. every graph is assigned a unique natural number
- this number can say interesting things about a graph
  - **tree width**
    - how hard do you have to squint for the graph to look like a tree?
  - **rank width**
    - how much information do you need to describe the connectivity of the graph?

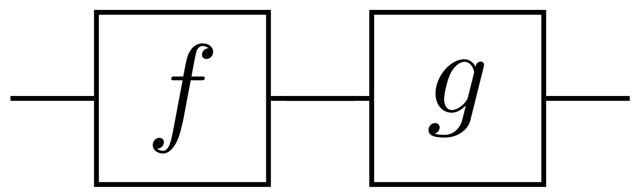
The next 700 graph widths

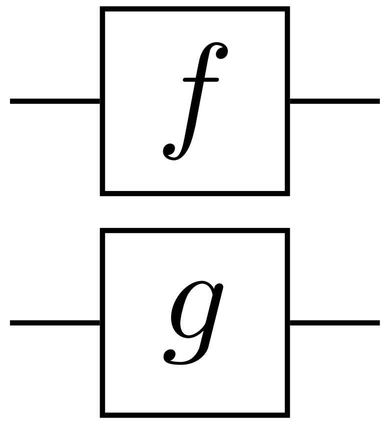
- tree width, branch width, rank width, twin width, clique width, …
  - **broccoli width**
    - come up with bespoke technique of broccoli **decomposing** a graph
    - impress your colleagues by specifying it in an impressively complicated way
    - assign a cost to the each part of a broccoli decomposition
    - the cost of a broccoli decomposition = max of the cost of its parts
    - broccoli width of a graph G = min of costs of broccoli decompositions of G

Why should structure people care?

- Widths can actually be super **powerful** things
  - families of graphs of bounded **tree width** allow for the development of efficient algorithms, which is very useful e.g. in verification (Courcelle)
  - other widths, like **rank width,** seem like good candidates for something like “Kolmogorov complexity” of graphs
    - e.g. discrete graphs and cliques have rank width of 0 and 1 respectively
- The seemingly ad hoc definitions of **decompositions** must therefore correspond to canonical **algebras** of “open” graphs
- So what are these algebras?

Monoidal categories as algebras of graphs

- monoidal category = **algebra** where one can
  - compose = glue things 

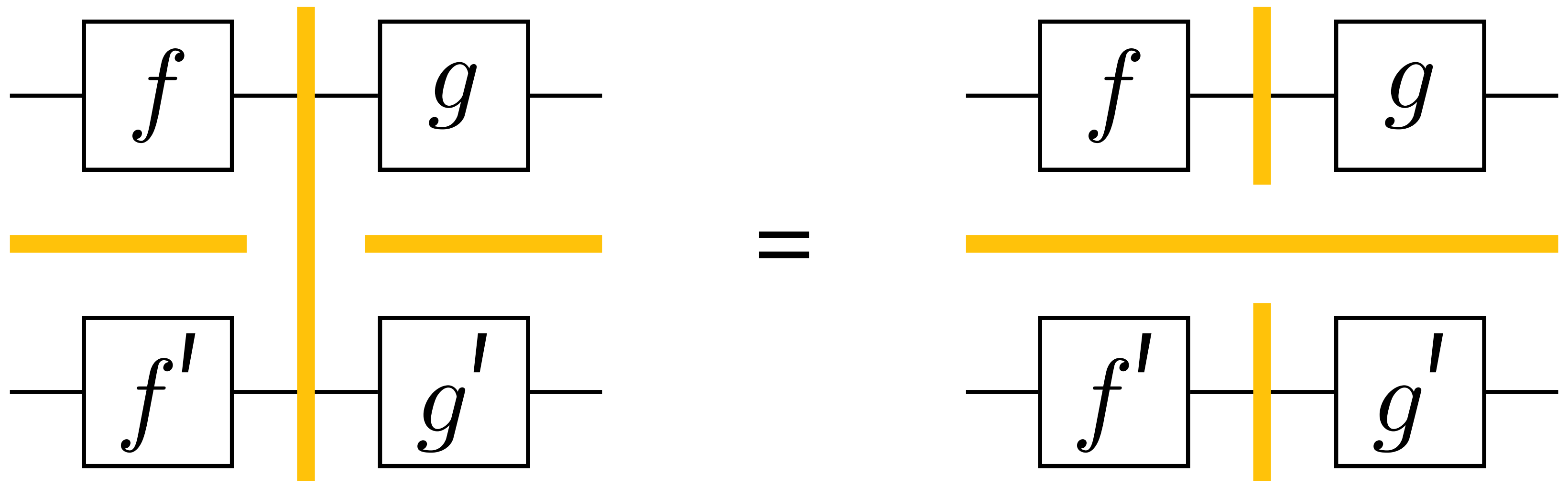
  - tensor = stack things 

- fix a prop **OGrph** of open graphs, where the scalar morphisms 0->0 are graphs
- there are several different possibilities for **OGrph**, specifying one amounts to choosing an algebra

Claim

(all?) reasonable notions of width arise from an underlying monoidal category of open graphs

- start with an **OGrph**
- a **decomposition** of G is a syntactic expression in ; and ⊗ that evaluates to G
- define **monoidal width**

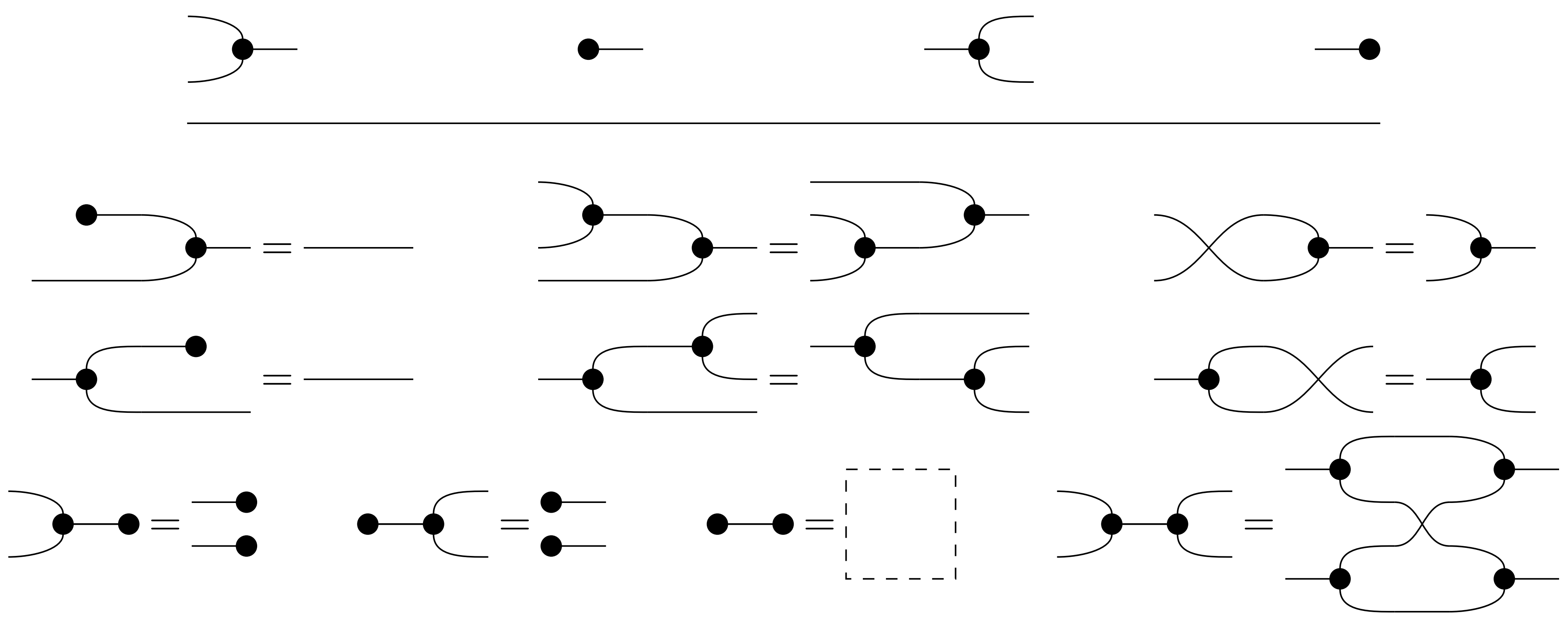


  - compositions along m cost m
  - tensor products cost 0
  - “atoms” cost something reasonable, like number of vertices
  - price a decomposition according to its most expensive operation
  - monoidal width = the price of cheapest decomposition

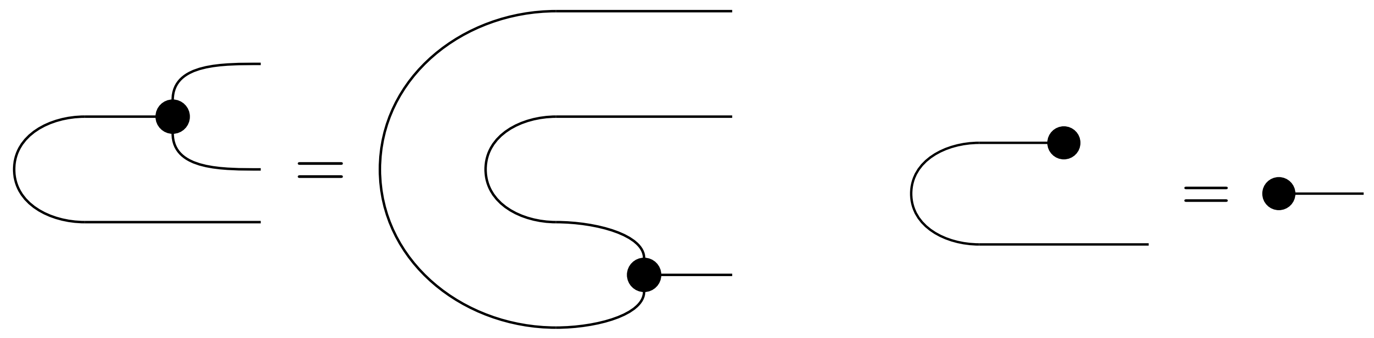
First **OGrph = Csp(UGraph)**

- An undirected graph G=(V, E, ends) where ends : E -> $\boldsymbol{P}_2$(v)
- **UGraph** = category with undirected graphs as objects, their homomorphisms as arrows
  - not difficult to verify that **UGraph** has colimits
- Csp(**UGraph**) cospans m -> G <- n where m, n are finite discrete graphs with m, n vertices, respectively
- composition is by pushout
  - open graphs are glued along common vertices
- tensor product is coproduct

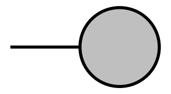
Second **OGrph** = “Bialgebra + cups + vertices”



= Mat(**N**)



+ algebra of adjacency matrices



+ vertices

- We call this **Gph**
  - open graphs are glued along edges

Main results (so far)

- tree width ~ monoidal width in Csp(**UGraph**)
  - we actually show that monoidal width ~ branch width, but it is known that branch width ~ tree width
- rank width ~ monoidal width in **Grph**
  - ~ = within a constant factor

- All these are simple to state, but pretty hard work to prove = peer review kryptonite

So what?

- For power people
  - a general theory of decomposition and a unified approach can help
    - what are the natural notions of width for other kinds of graphs (e.g. tree width for directed graphs?)
- For structure people
  - cool new algebras to discover
- For everyone
  - notion of monoidal with makes sense in other settings (e.g. Petri nets, matrices, affine relations, …) and often seems relevant
  - decompositions as elements of a data structure

Bibliography

- Elena Di Lavore and PS. *Monoidal Width: Unifying Tree Width, Path Width and Branch Width*, arXiv:2202.07582
- Elena Di Lavore and PS. *Monoidal Width: Capturing Rank Width*, arXiv:2205.08916, to appear in proceedings of ACT 2022