A Category of Plane Graphs with Substitution and Pattern Matching

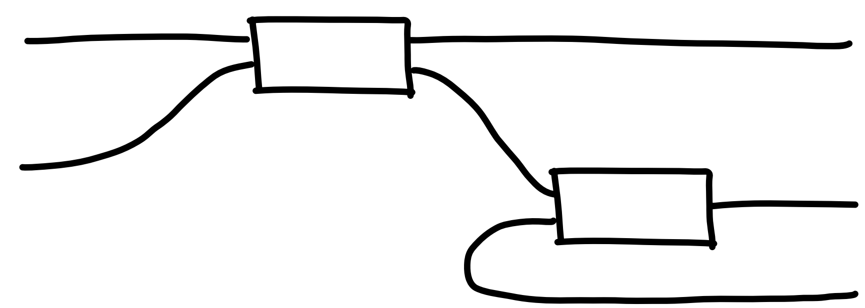
Malin Altenmüller

CATNIP 19/11/24

malin.altenmuller@ed.ac.uk maltenmuller.github.io

String Diagrams

- graphical syntax for monoidal categories
- composition & tensor product straight forward



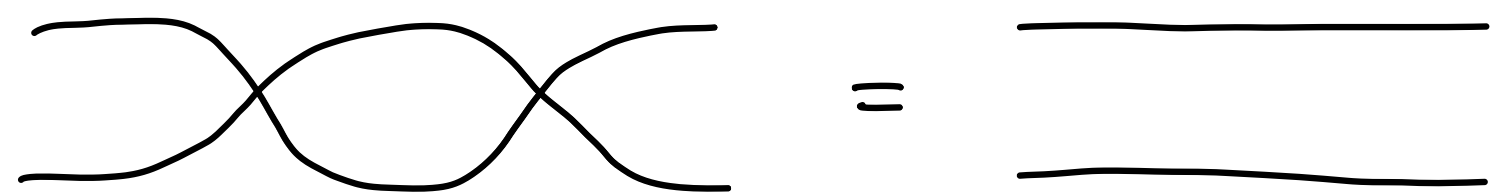
- represent computational processes
- reasoning by rewriting

String Diagrams

- specific properties in the MC translate to their diagrams

- symmetric monoidal categories (SMC)

"only connectivity matters"

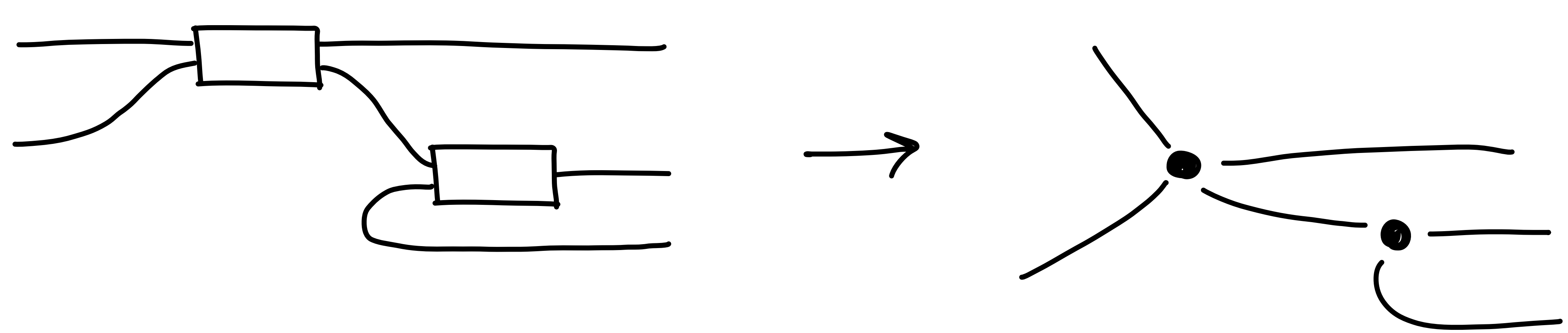


- interested in the non-symmetric case:

    - quantum circuits: swap is non-trivial
    - printing circuits: swap is not possible
    - generalises symmetric & braided case

Graphs

- graphs as combinatorial representation
- translation : wires → edges , boxes → vertices



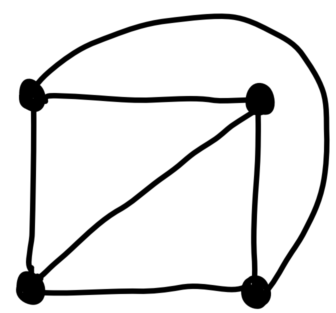
- preserve vertex arity!
- reasoning by graph rewriting

$$\boxed{\text{Graphs of non-symmetric diagrams?}}$$

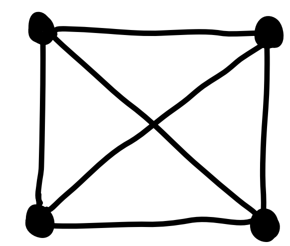
Graph Embeddings

"connectivity
+ topology"

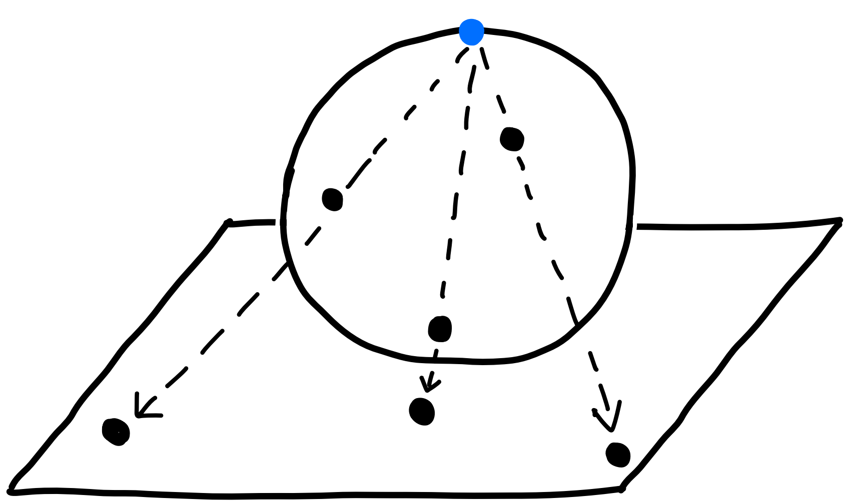
- drawing of a graph onto a surface

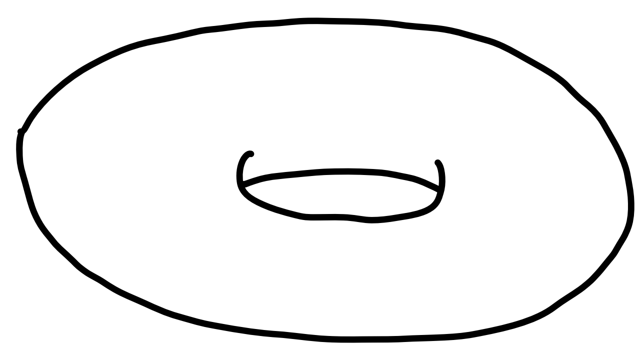


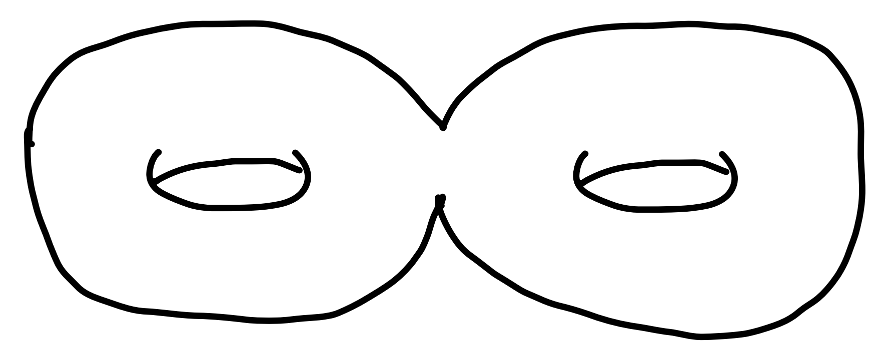
plane



non-plane

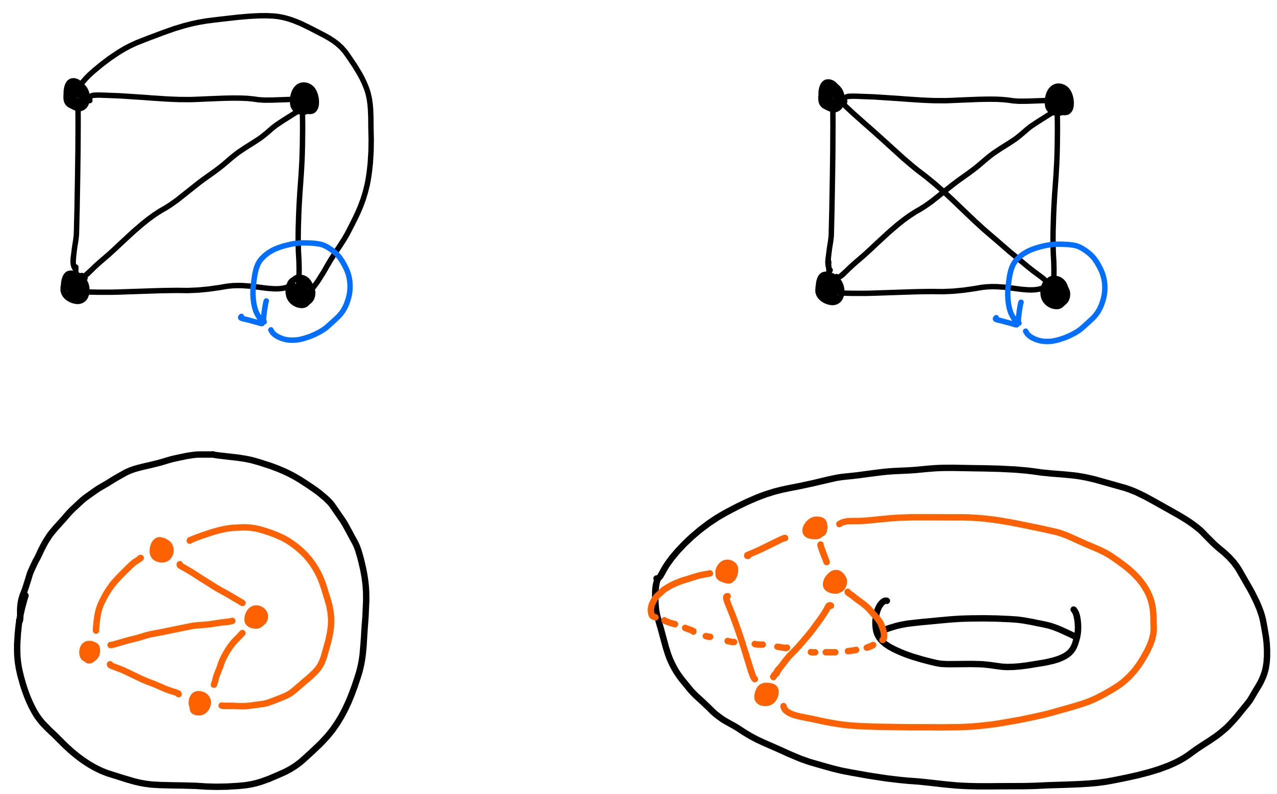






Rotation Systems

- store the order of edges around each vertex

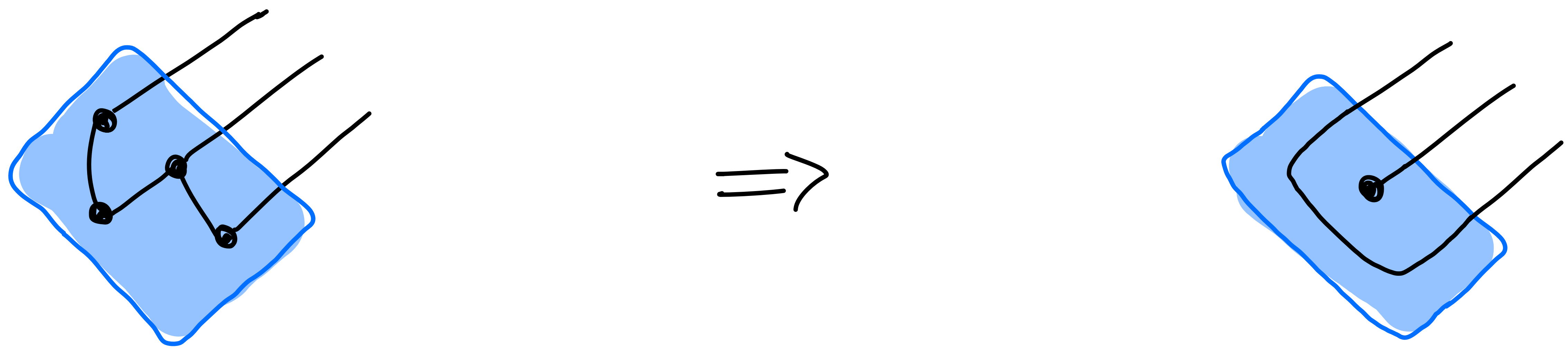


Theorem:

A rotation system uniquely determines a graph embedding. [1]

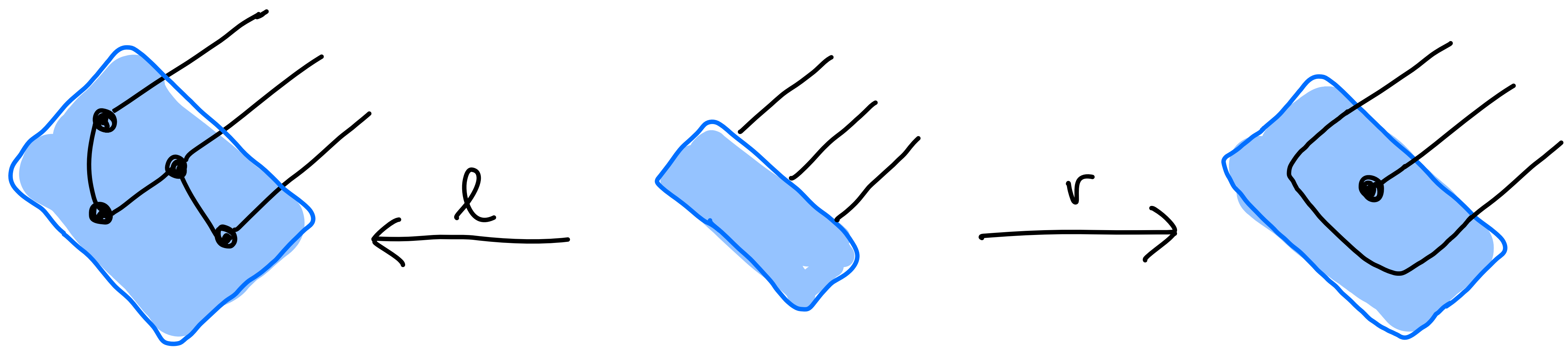
An Example of DPO-Rewriting

given a rewrite rule



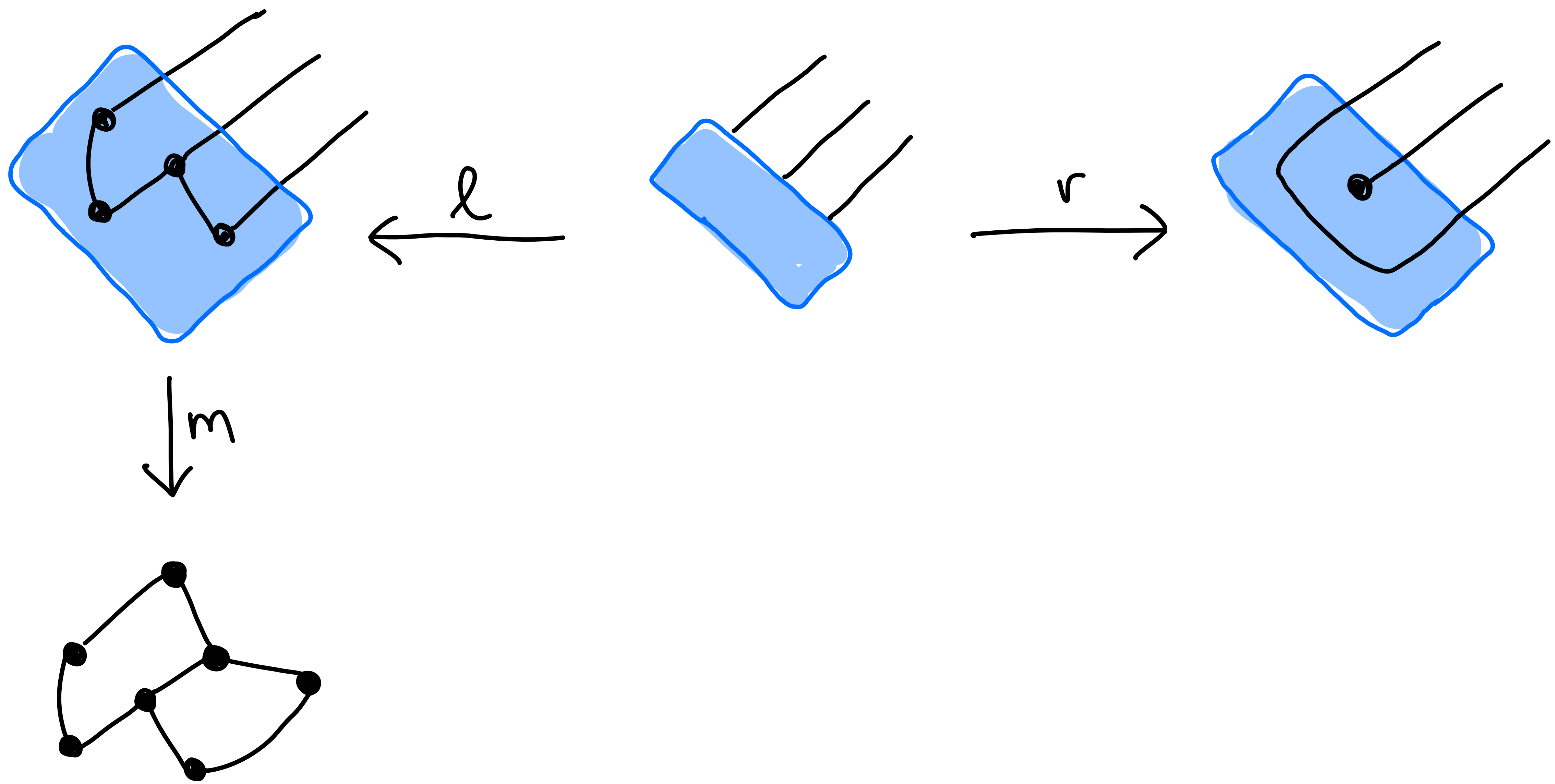
An Example of DPO - Rewriting

rewrite rule as span with common boundary



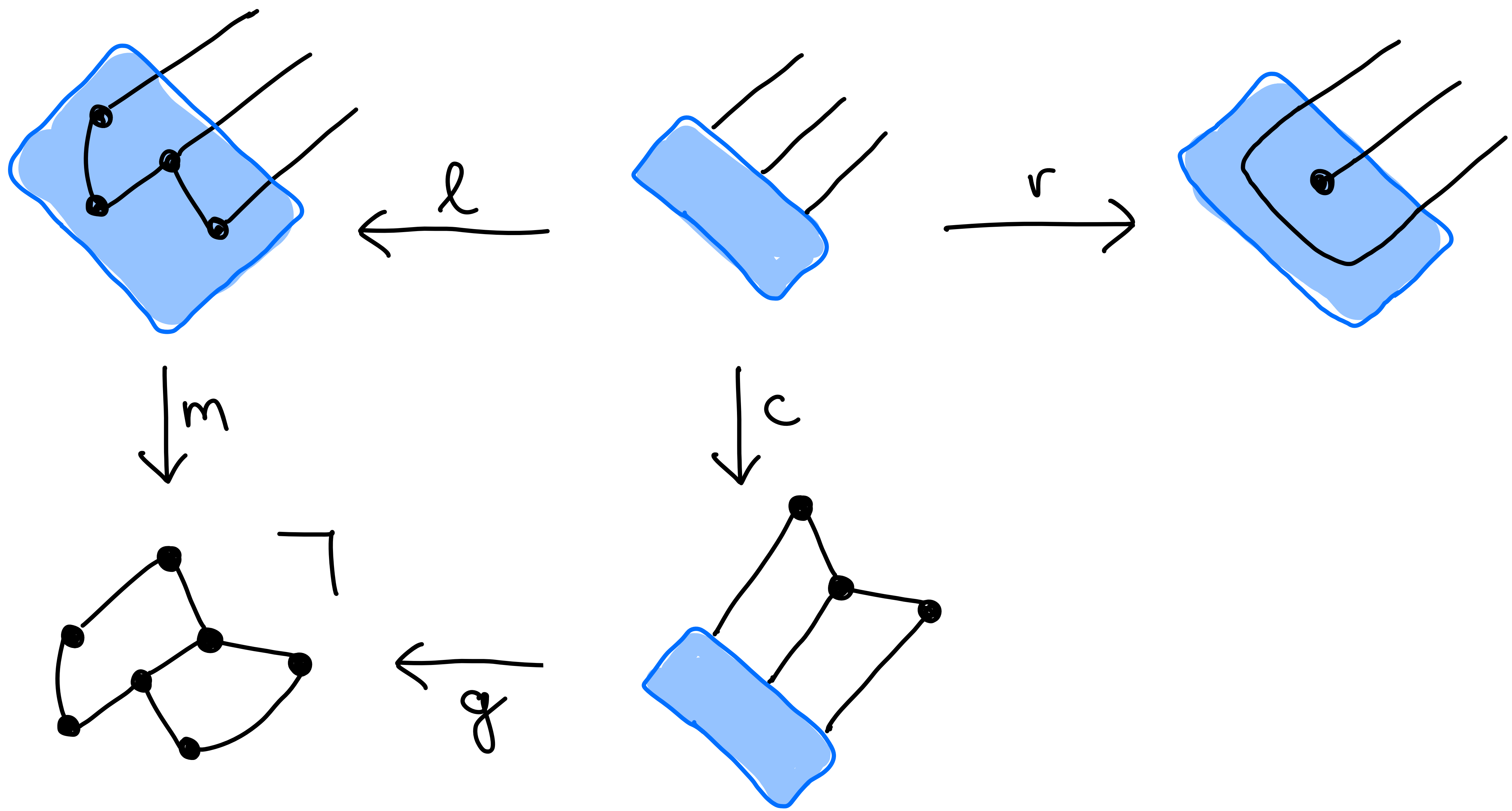
An Example of DPO-Rewriting

matching of the LHS onto a graph



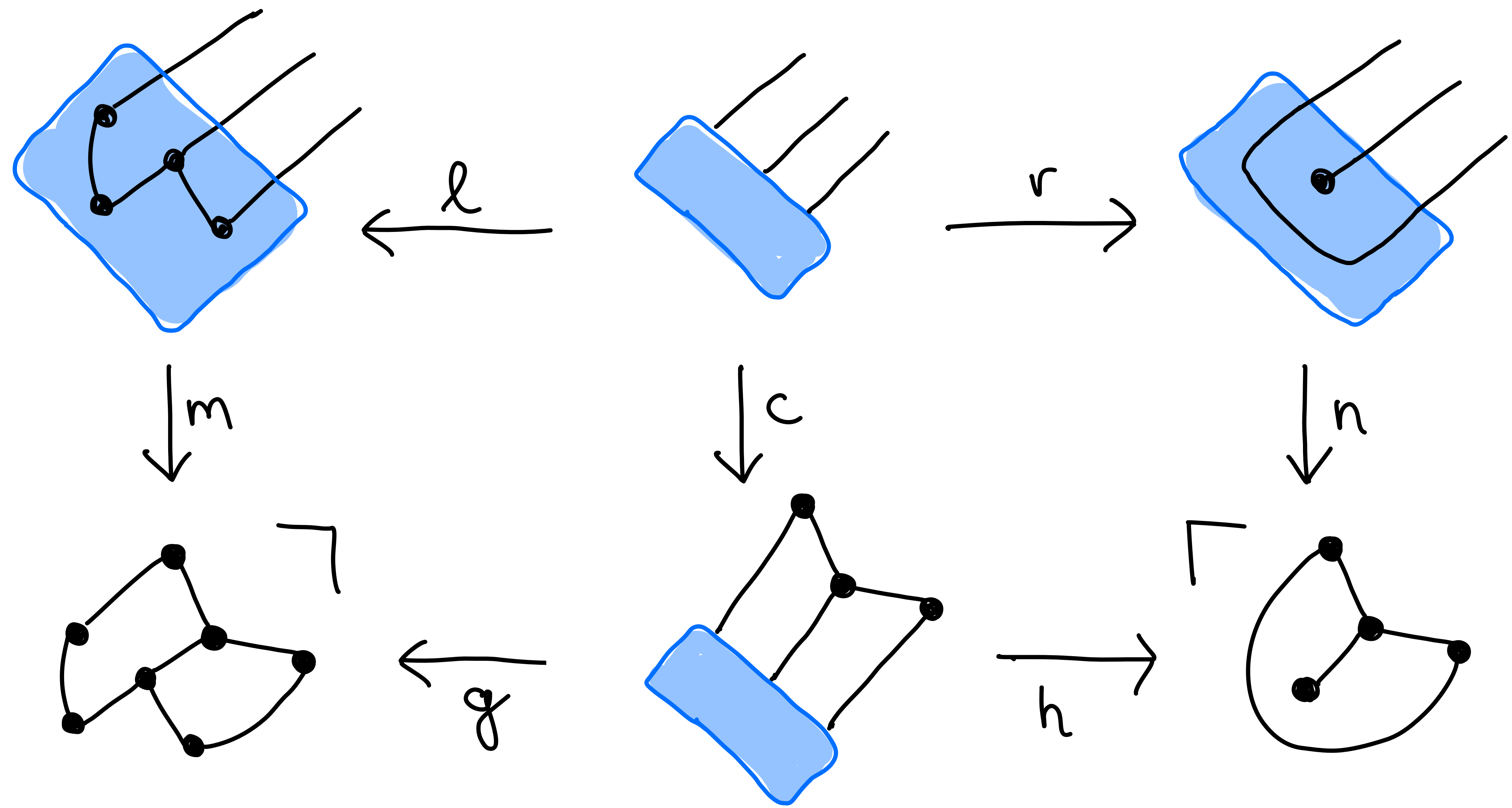
An Example of DPO-Rewriting

construct context graph by pushout complement



An Example of DPO-Rewriting

construct final graph by pushout



need: pushouts, (unique) pushout complements, notion of embedding

"adhesive" categories [2]

Standard Category of Graphs

- graphs $\quad G : E \underset{t}{\overset{s}{\rightrightarrows}} V$

- morphisms $G \to G'$ are pairs $\quad f_E : E \to E'$
  $\qquad f_V : V \to V'$

  such that

$$\begin{tikzcd} E \arrow[r, "f_E"] \arrow[d, "s"'] & E' \arrow[d, "s'"] \\ V \arrow[r, "f_V"'] & V' \end{tikzcd}$$

(and similar for t)

⌜Remark: all graphs are drawn undirected here⌟

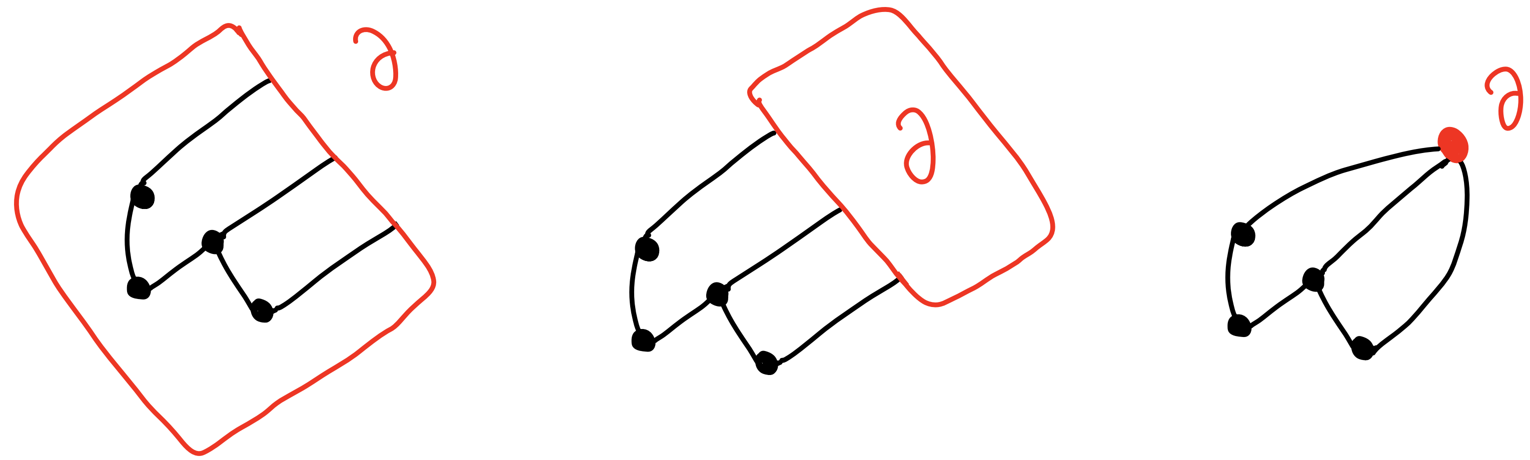
Open Graphs

- processes have inputs and outputs
- diagrams can have holes

but

- morphisms of open graphs don't preserve the surface
- cannot assign rotation information to loose edges

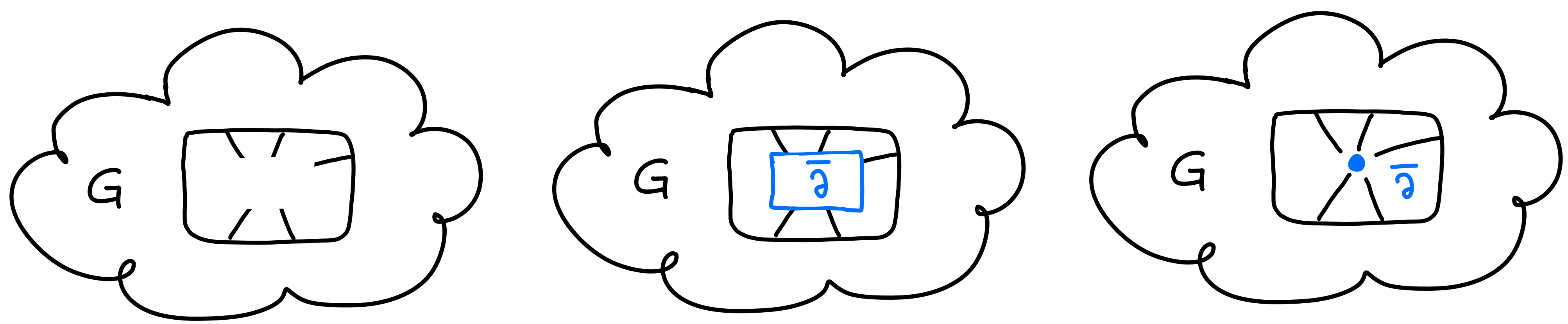
Boundary Vertex



- identify the outside of a graph
- attach input and output edges to it
- outside as a region of the graph
- contract to a single boundary vertex

Dual Boundary Vertex

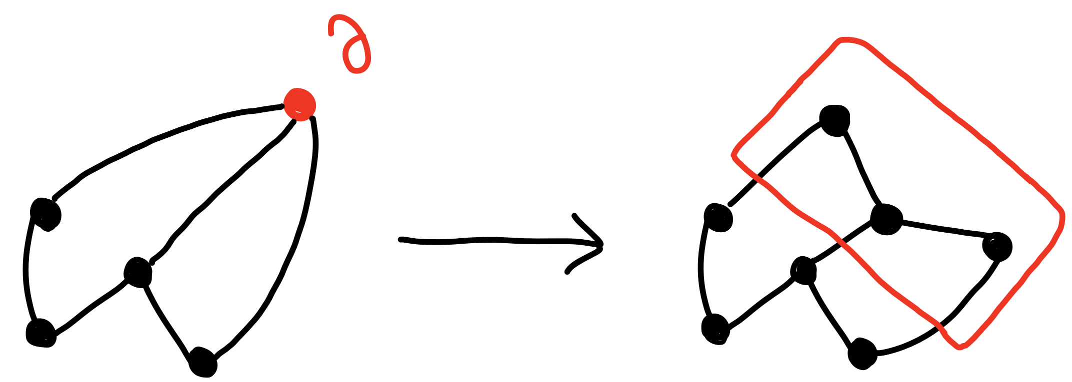
- same idea, for any holes in a graph



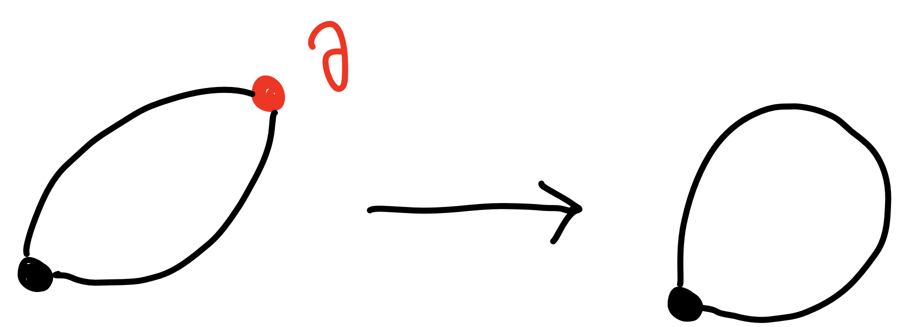
- all graphs are total
  $\rightarrow$ I can add rotation information to all edges

Requirements for Graph Morphisms

- vertex map needs to be partial



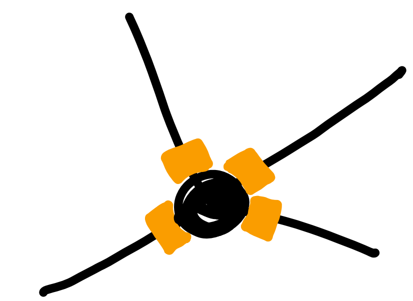
- edge map cannot be injective



What is the right notion of embedding?

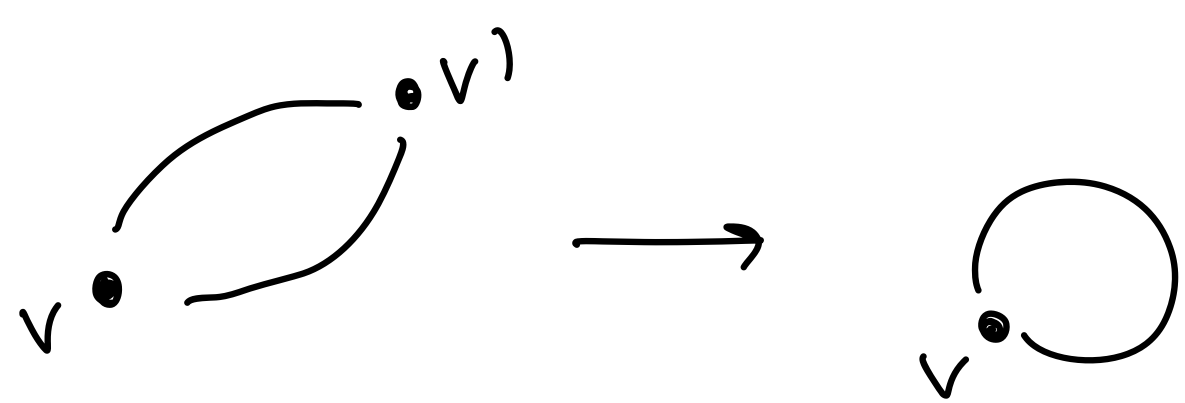
Flags and Flag Maps

- connection points between vertices and their incident edges
  → pairs (v, e)



- flag map (f_E, f_V) partial map induced by graph map
- characterise morphisms / embeddings on the flag map

Example:



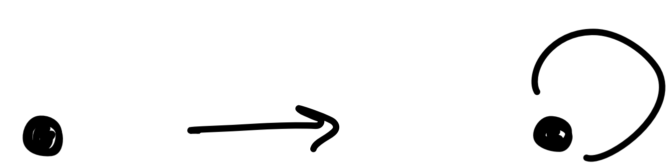
is flag-injective

Flag Surjectivity

start with the condition for standard graph morphisms

$$\begin{tikzcd} E \arrow[r, "f_E"] \arrow[d, "s"'] & E' \arrow[d, "s'"] \\ V \arrow[r, "f_V"] & V' \end{tikzcd}$$

What about vertices with no edges attached ?

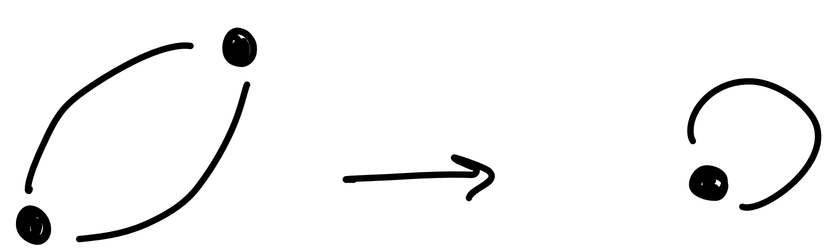


Flag Surjectivity

condition on vertices, by considering the preimage

$$\begin{tikzcd} V \arrow[r, "f_V"] \arrow[d, "S^{-1}"'] & V' \arrow[d, "S'^{-1}"] \\ P(E) \arrow[r, "P(f_E)"] & P(E') \end{tikzcd}$$

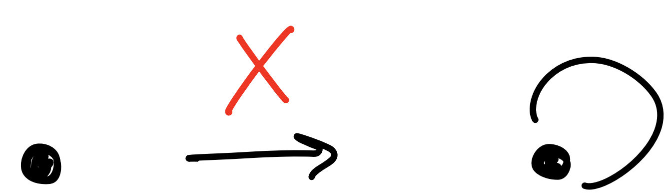
What about vertices where f_V is undefined ?

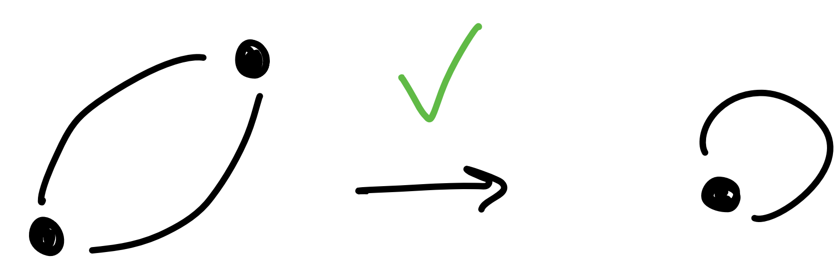


Flag Surjectivity

Flag surjectivity = lax commutation of the square:

$$\begin{tikzcd} V \arrow[r, "f_V"] \arrow[d, "s^{-1}"'] \arrow[dr, phantom, "\geq"] & V' \arrow[d, "s'^{-1}"] \\ P(E) \arrow[r, "P(f_E)"] & P(E') \end{tikzcd}$$





Graphs with Circles $\underline{\mathcal{G}}$

- objects are total graphs (as before)
- morphisms are (f_E, f_V) where
  - f_E is total
  - the induced flag map is surjective

  \+ other conditions
- embeddings additionally are
  - flag injective

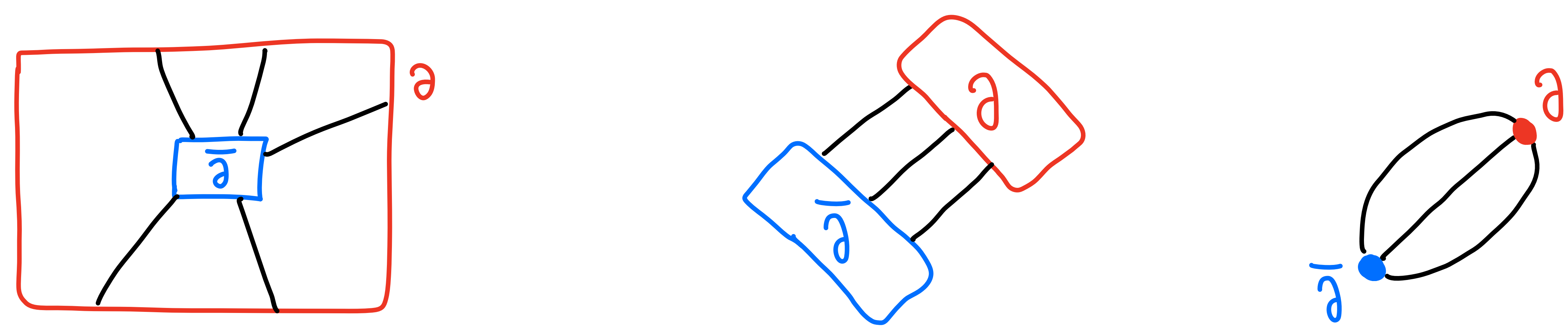
  \+ other conditions

Rewriting

- this category of graphs is not adhesive!

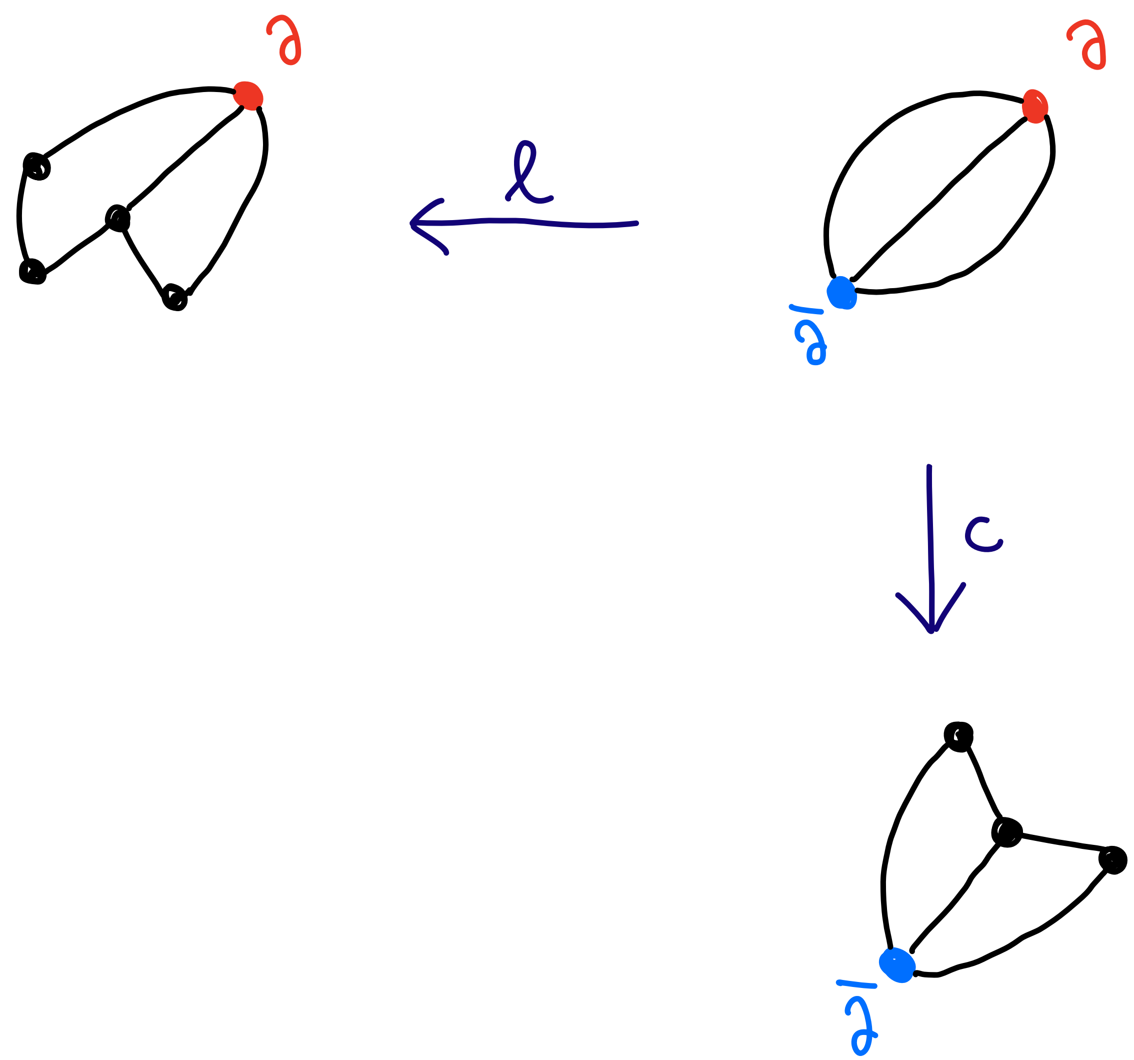
but it has enough adhesive properties in the case we're interested in

Boundary Graphs have an outside and a hole:



Partitioning Spans

split the graph into context and subgraph

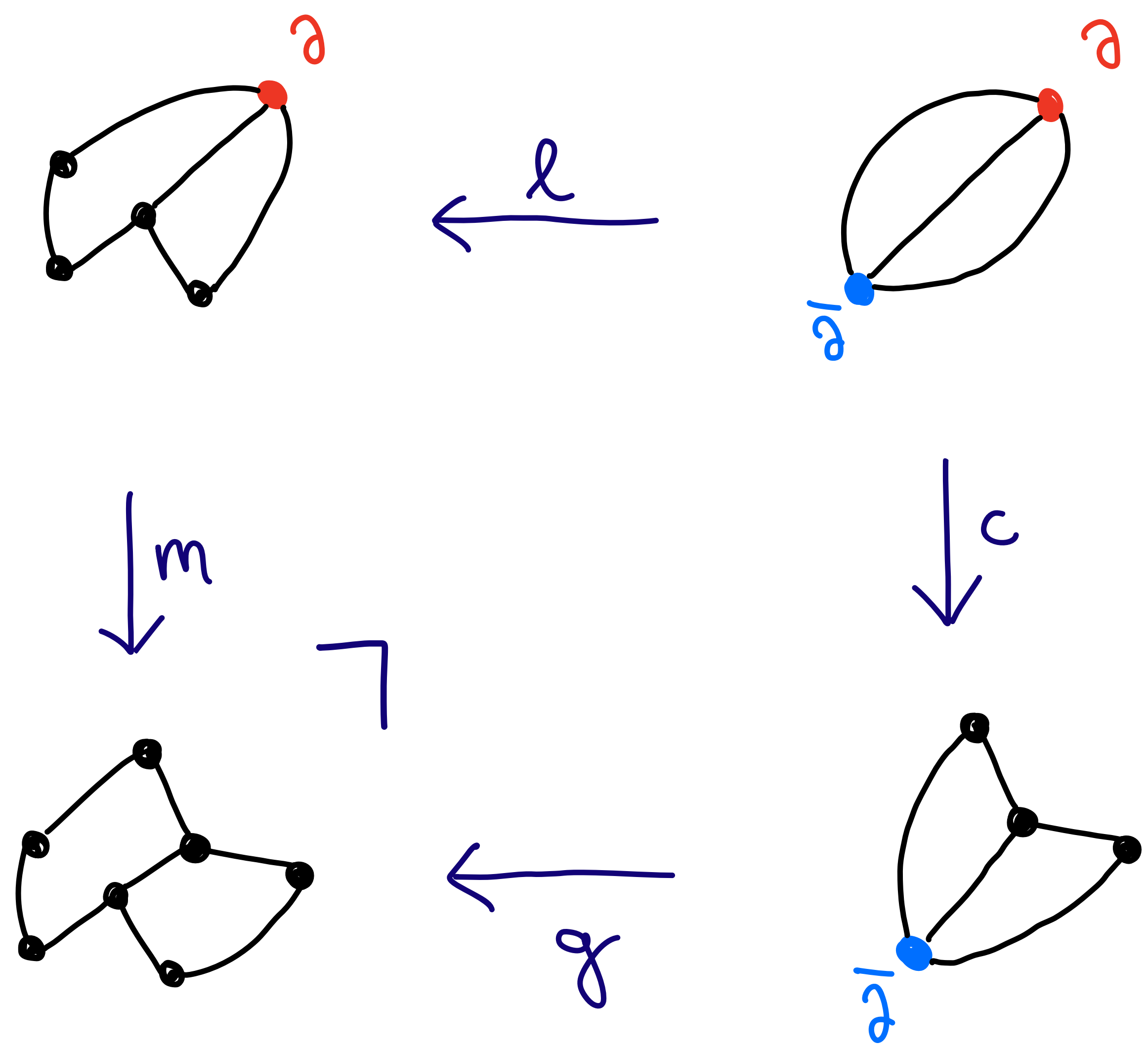


ℓ_V undefined on $\bar{\partial}$
defined on ∂

c_V defined on $\bar{\partial}$
undefined on ∂

Partitioning Spans

split the graph into context and subgraph



ℓ_V undefined on $\bar{\partial}$

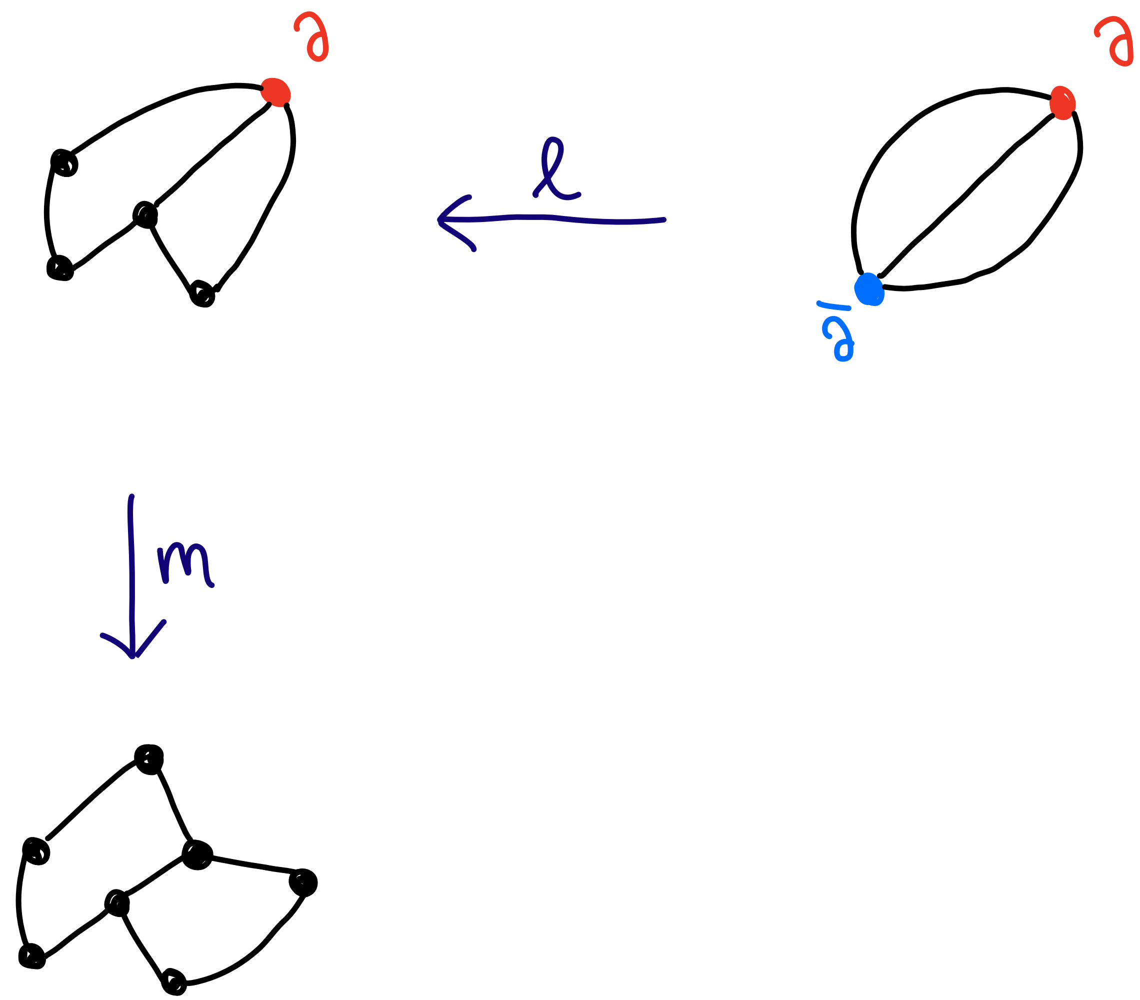
defined on ∂

c_V defined on $\bar{\partial}$

undefined on ∂

Theorem: In $\underline{G}$ pushouts of partitioning spans exist.

Boundary Embeddings

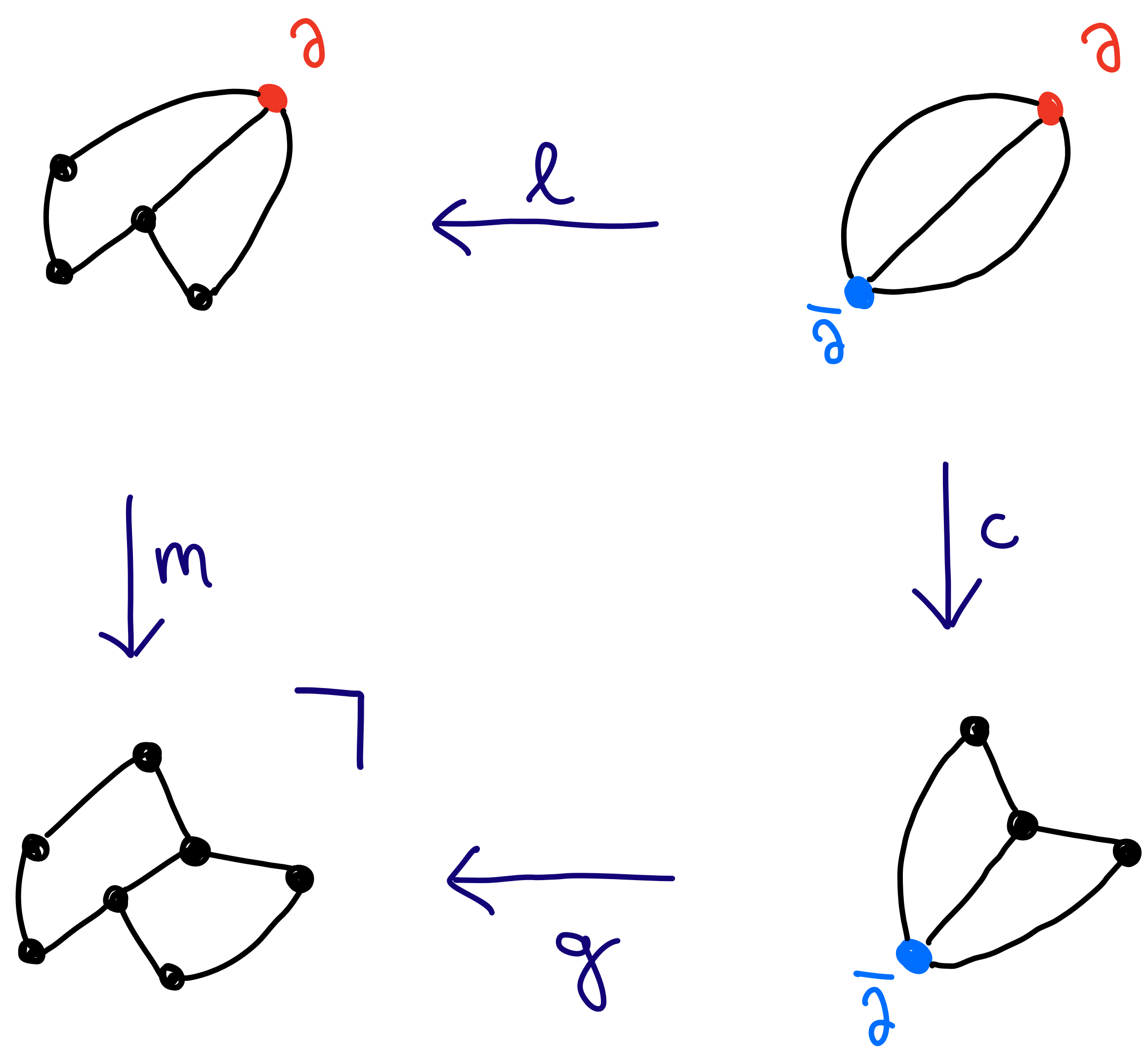


ℓ_v undefined on $\overline{\partial}$

defined on ∂

m undefined on ∂

Boundary Embeddings



ℓ_V undefined on $\bar{\partial}$

defined on ∂

m undefined on ∂

Theorem: In $\underline{G}$ pushout complements of boundary embeddings exist and are unique*.

Category of Rotation Systems

- objects : graphs + cyclic ordering of flags for all vertices
- morphisms: same as $\underline{G}$ + order preservation condition

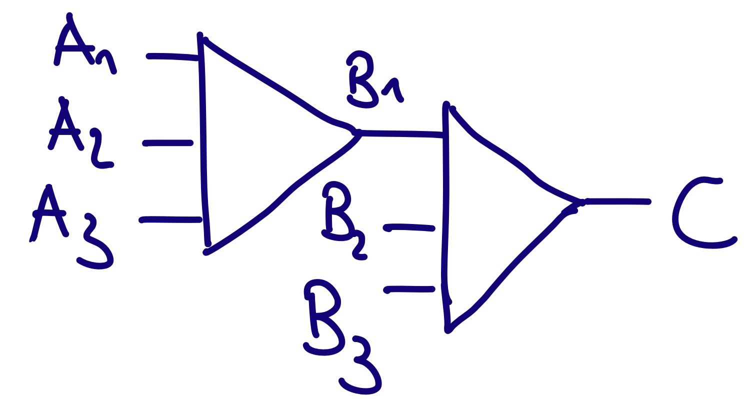
e.g.

$$\begin{tikzcd}
V \arrow[r, "f_V"] \arrow[d, "s^{-1}"'] \arrow[dr, phantom, "\geq"] & V' \arrow[d, "s'^{-1}"] \\
\mathrm{CList}(E) \arrow[r, "\mathrm{CList}(f_E)"'] & \mathrm{CList}(E')
\end{tikzcd}$$

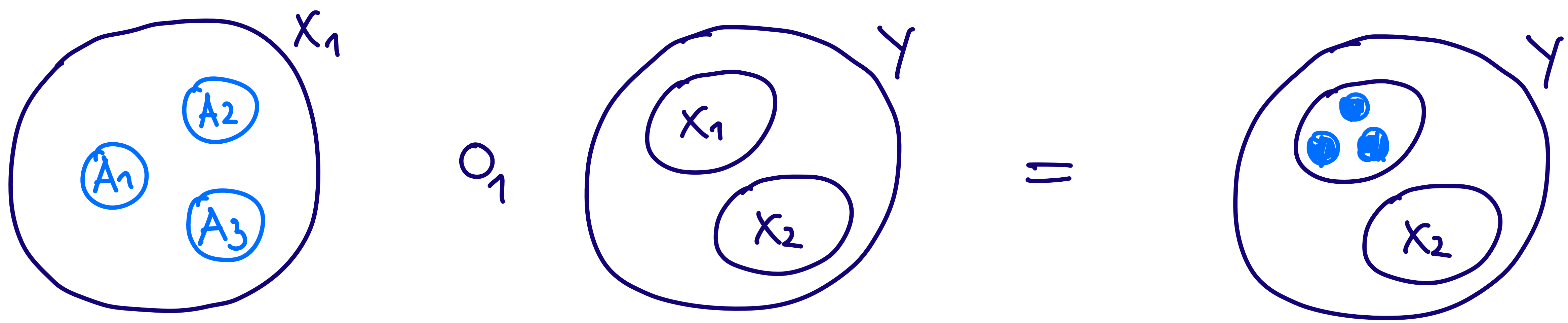
Proposition: Pushouts and pushout complements are the same as in the underlying category of graphs.

Operads

- arrows can take multiple arguments



- little discs operad [3]



$$A_1, A_2, A_3 \longrightarrow X_1 \quad \circ_1 \quad X_1, X_2 \longrightarrow Y \qquad\qquad A_1, A_2, A_3, X_2 \longrightarrow Y$$

- wiring diagrams operad [4]: substitute diagram for special vertex

Operad of Plane Graphs

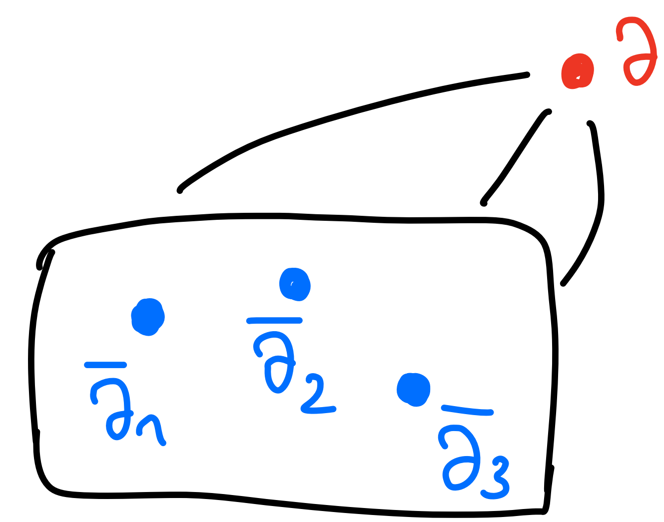
- objects are rotations
- morphisms are plane graphs

  inputs are dual boundary vertices

  output is its boundary vertex

$$G : \overline{\partial}_1, \dots, \overline{\partial}_n \leftarrow \partial$$

- composition is given by substitution

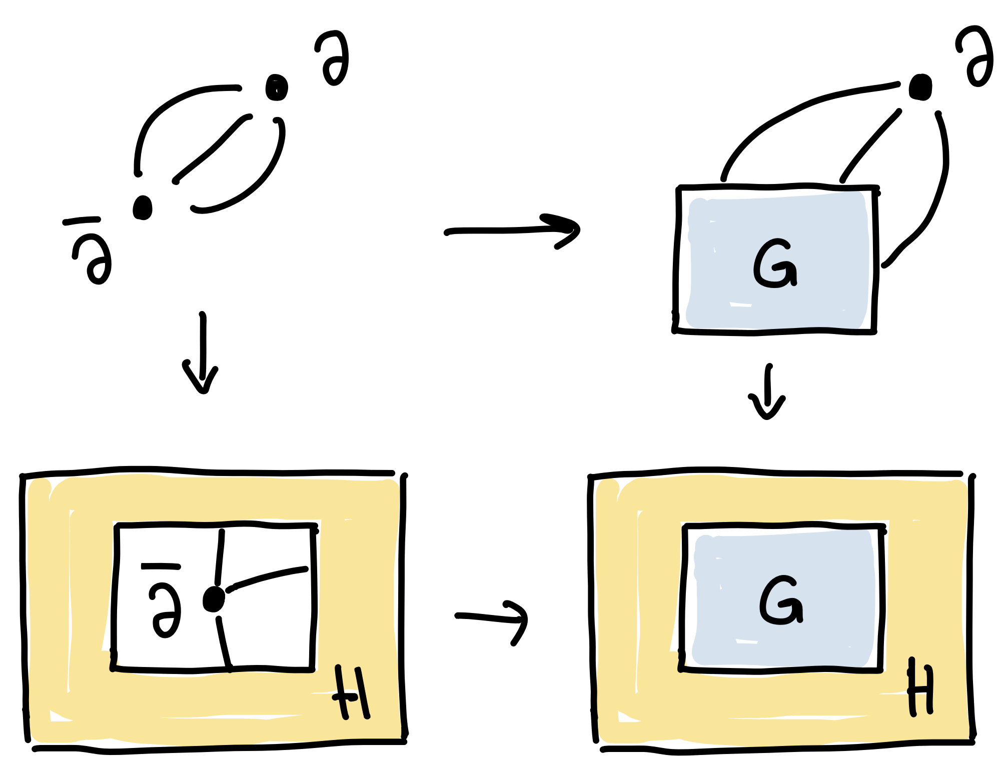


Composition is Substitution

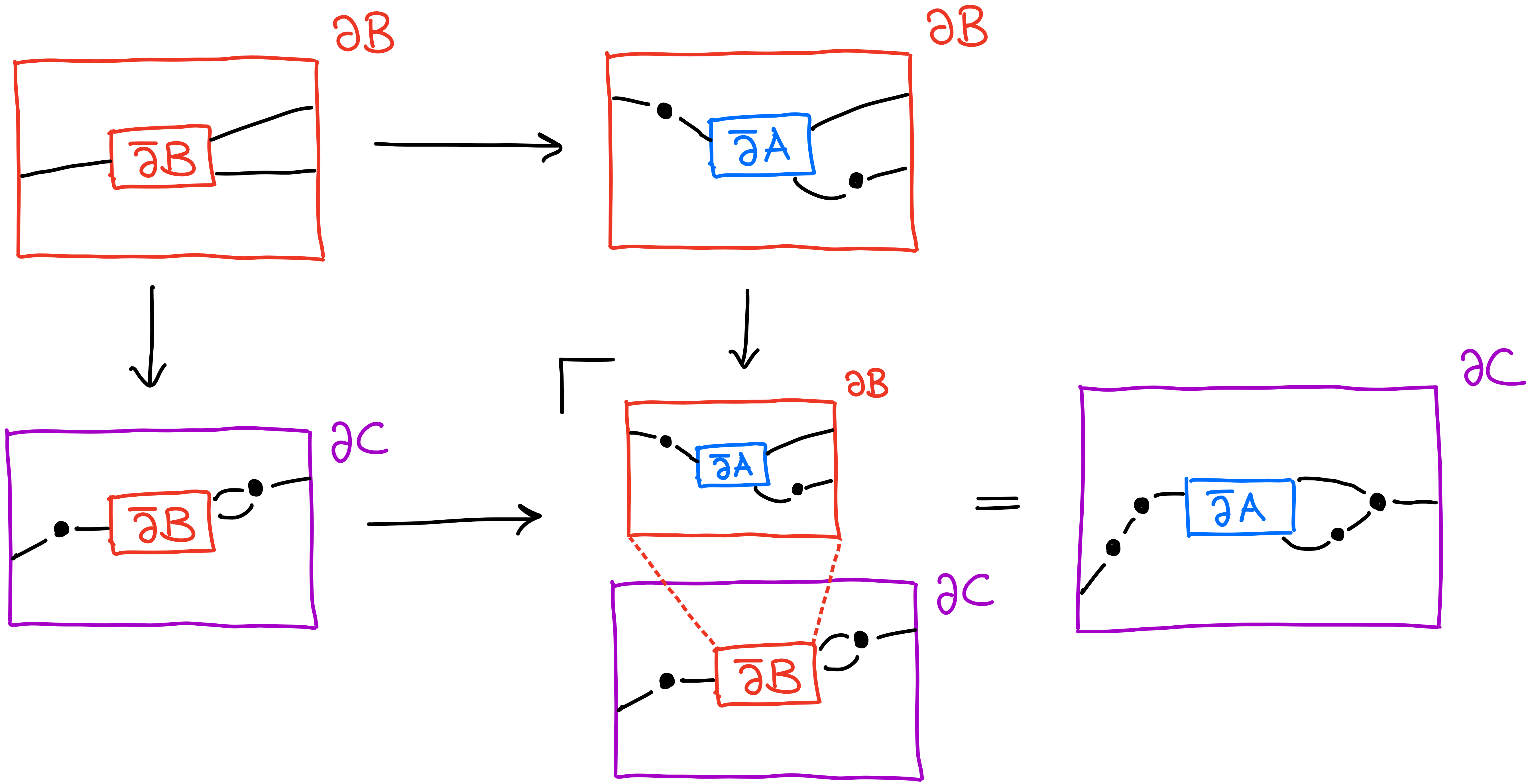
$$G : \bar{\partial}_1, \ldots, \bar{\partial}_i, \ldots, \bar{\partial}_n \leftarrow \partial \qquad H : \bar{\partial}'_1, \ldots, \bar{\partial}'_m \leftarrow \partial_i$$

- composition $H \circ_i G$ is the pushout of the partitioning span

$$G \leftarrow \bar{\partial}_i \partial_i \longrightarrow H$$



Example of Operad Composition



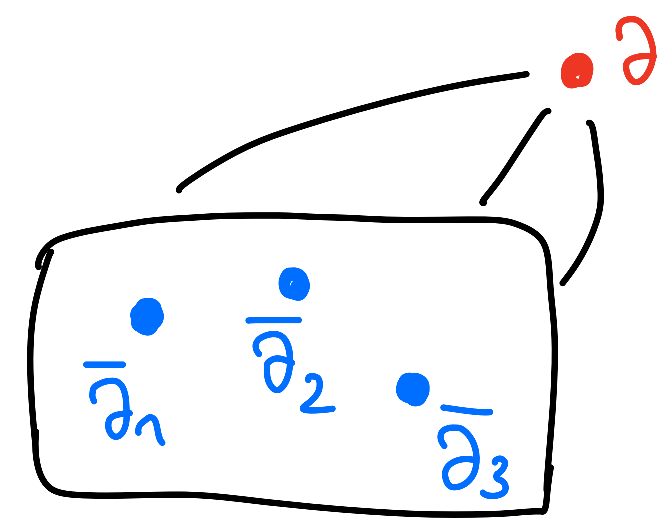
Co-Operad

- objects are rotations
- morphisms are plane graphs

  input is its boundary vertex

  outputs are dual boundary vertices

$$P : \partial \to \bar{\partial}_1, \dots, \bar{\partial}_n$$



- composition is given by substitution

- co-operads are patterns

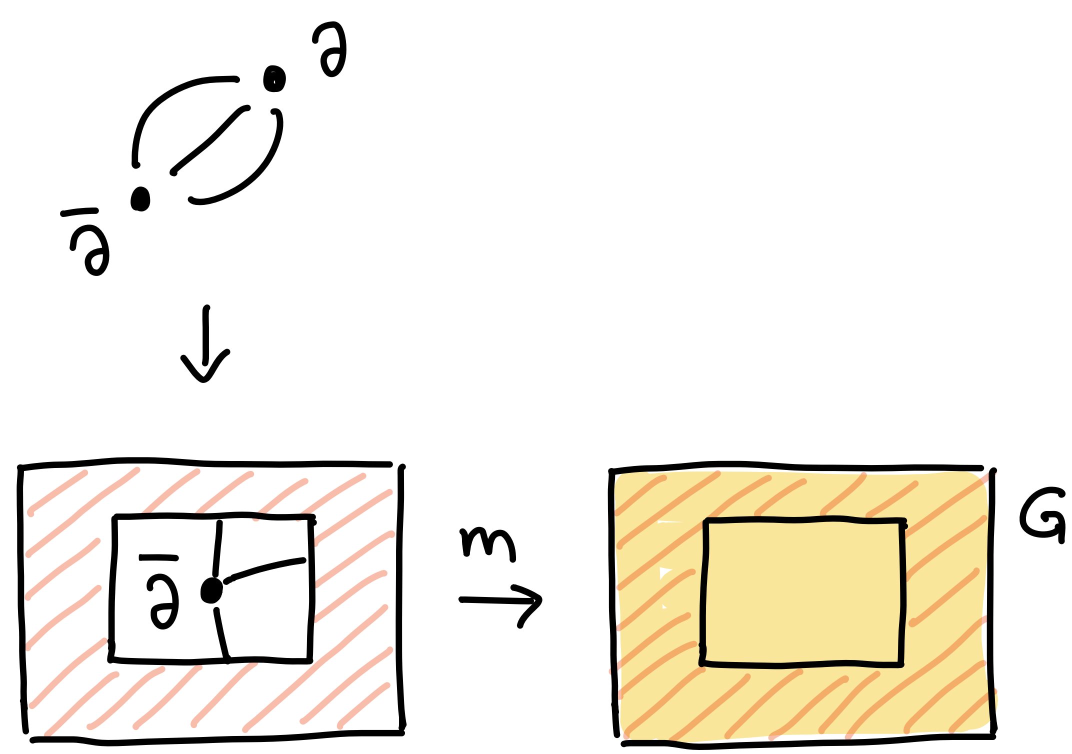
  the $\bar{\partial}_i$ are the pattern variables

Operad-Cooperad Interaction

Given a pattern $P: \partial \to \bar{\partial}_1, \ldots, \bar{\partial}_n$ and a graph $G: \bar{\partial}'_1, \ldots, \bar{\partial}'_m \leftarrow \partial$

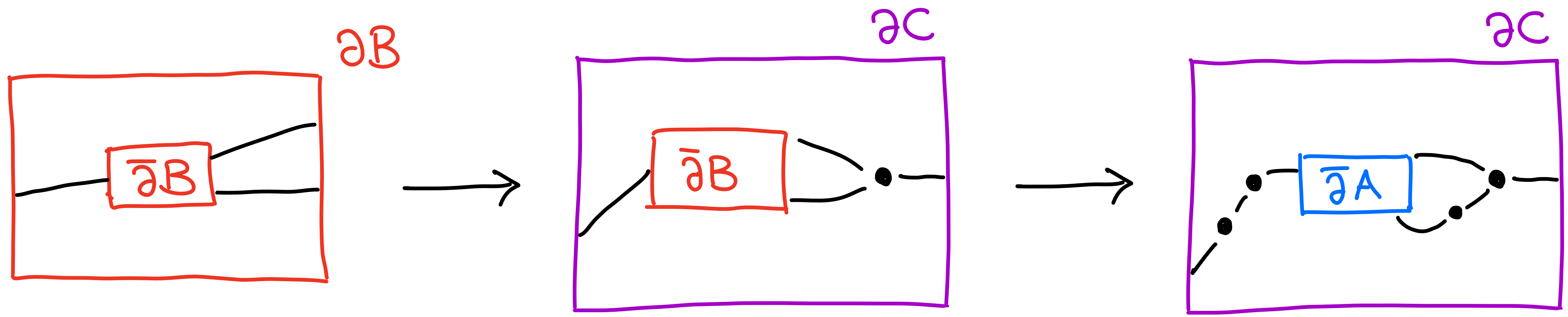
A match is a map $m: P \to G$ such that $\partial\bar{\partial} \to P \to G$

is a boundary embedding.



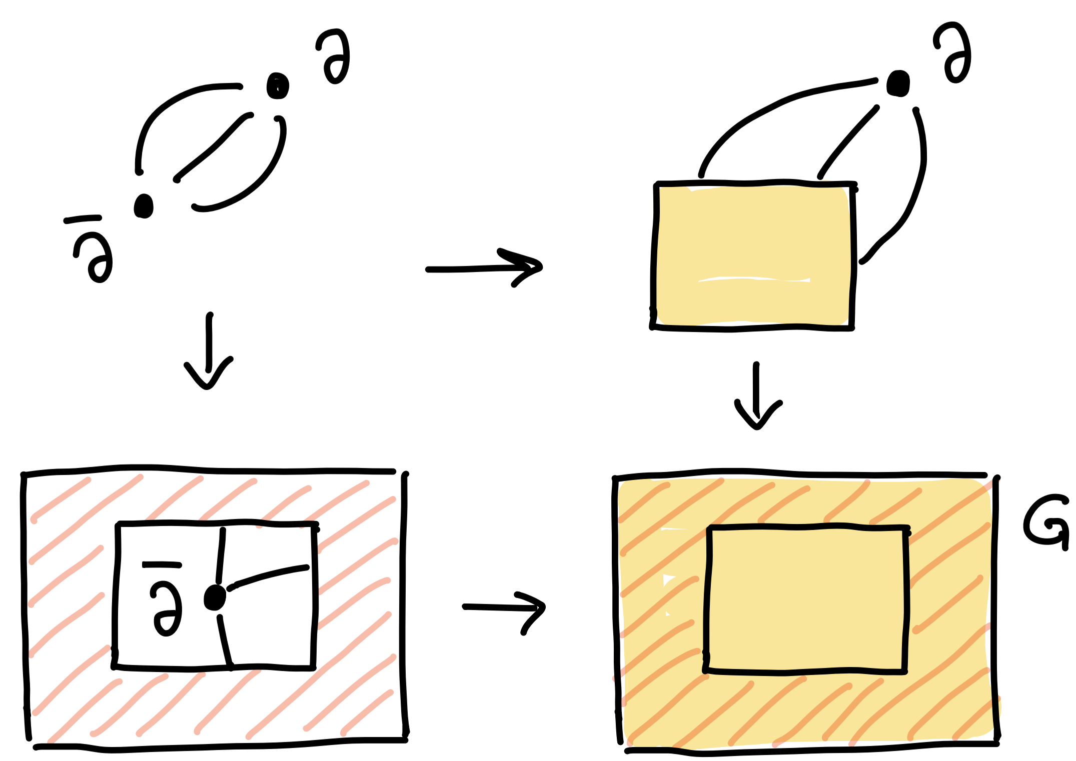
(A match can fail if there is no such m)

Example of a Match

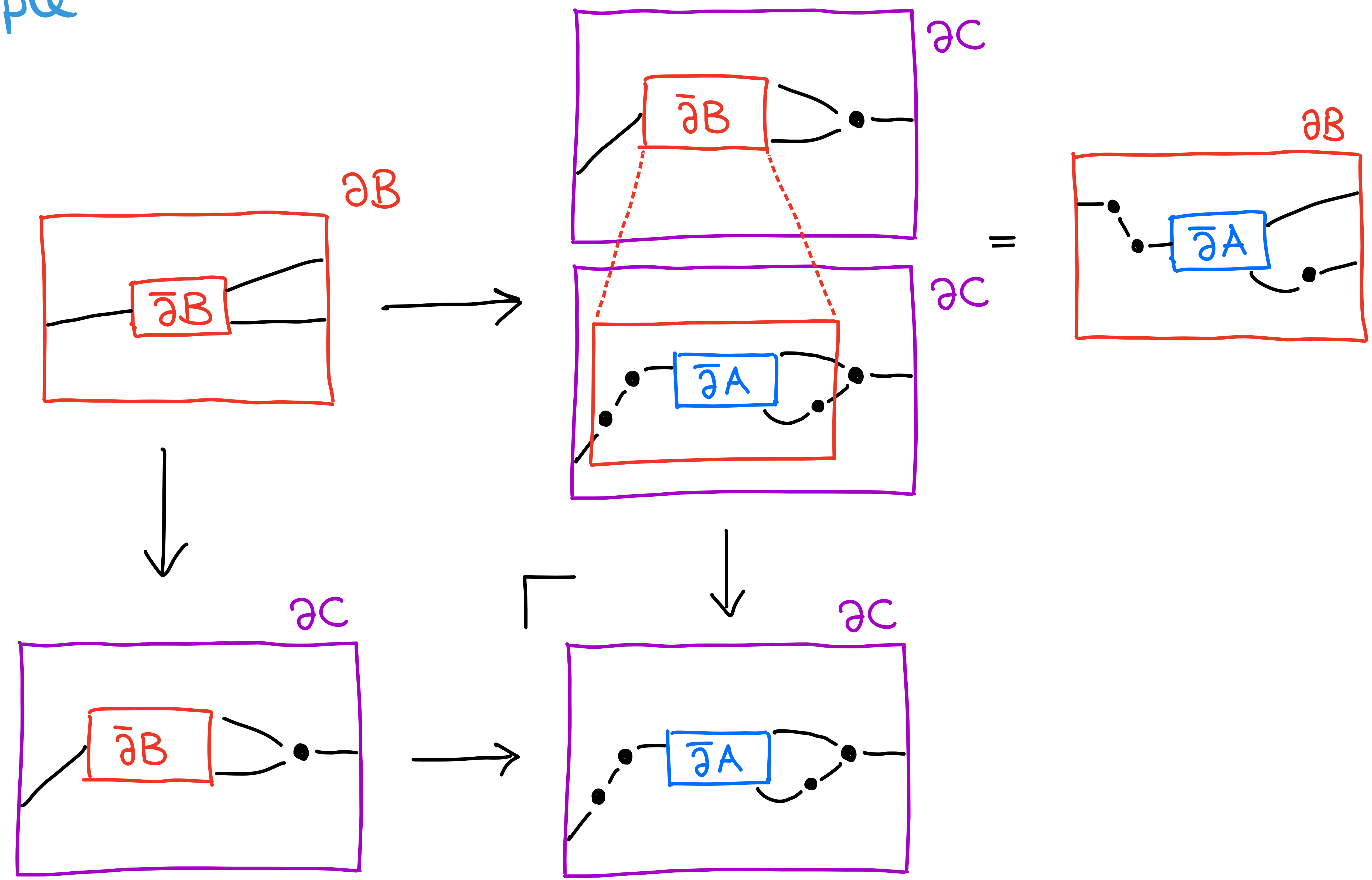


Calculating the match

... amounts to calculating the pushout complement of the boundary embedding $\partial\bar{\partial} \to P \to G$ (which exists!)



Example

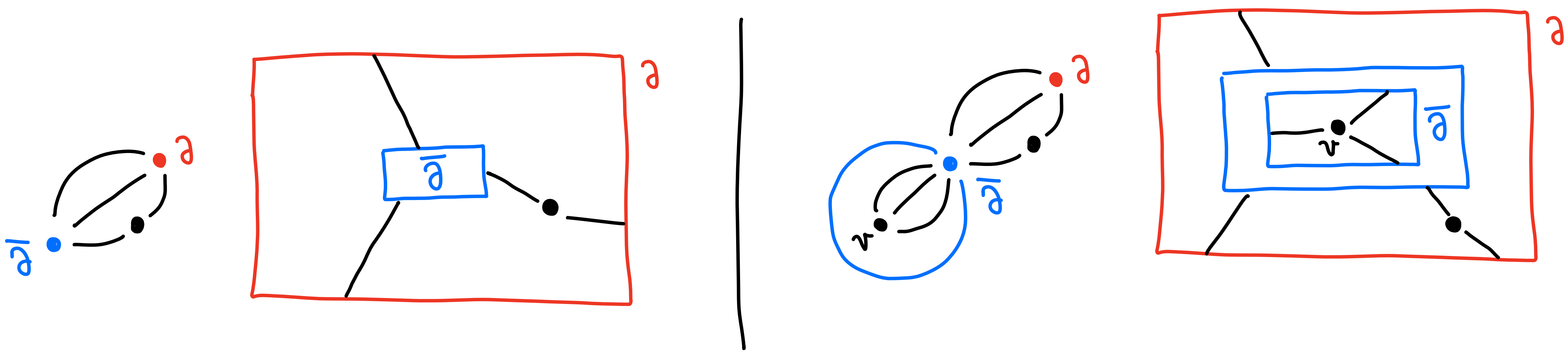


Summary

- non-symmetric monoidal categories are interesting
- represented by plane graphs
- introducing boundary vertices to avoid open graphs
- operad of graphs with substitution
- co-operad of patterns with substitution
- interaction yields notion of pattern matching

Future Work

- more than one hole for the operad / cooperad part
- higher genus surfaces, non-orientable surfaces?
- more complex boundary graphs, e.g.:



- co-operads framework for patterns (& matching) in other contexts. Metaprogramming?

References

[1] Jonathan Gross, Thomas Tucker: Topological Graph Theory. 2001.

[2] Steven Lack, Paweł Sobociński: Adhesive Categories. 2004.

[3] Tom Leinster: Higher Operads, Higher Categories. 2004.

[4] David I. Spivak: The operad of wiring diagrams [...]. 2013.

Appendix A – Graphs with Circles

Definition 1.18. A *graph with circles* is a 5-tuple $G = (V, E, O, s, t)$ where (V, E, s, t) is a total graph and O is a set of *circles*. For notational convenience we define the set of *arcs* as the disjoint union $A = E + O$.

A morphism $f : G \to G'$ between two graphs with circles consists of two (partial) functions $f_V : V \rightharpoonup V'$ as above, and $f_A : A \to A'$, satisfying the conditions listed below. Note that any such f_A factors as four maps,

$$f_E : E \to E' \qquad f_{EO} : E \to O'$$

$$f_{OE} : O \to E' \qquad f_O : O \to O'$$

The following conditions must be satisfied:

1. $f_A : A \to A'$ is total
2. the component $f_{OE} : O \to E'$ is the empty function
3. the pair (f_V, f_E) forms a flag surjection between the underlying graphs in $\mathbf{B}$.

If, additionally, the following three conditions are satisfied, we call the morphism an *embedding*:

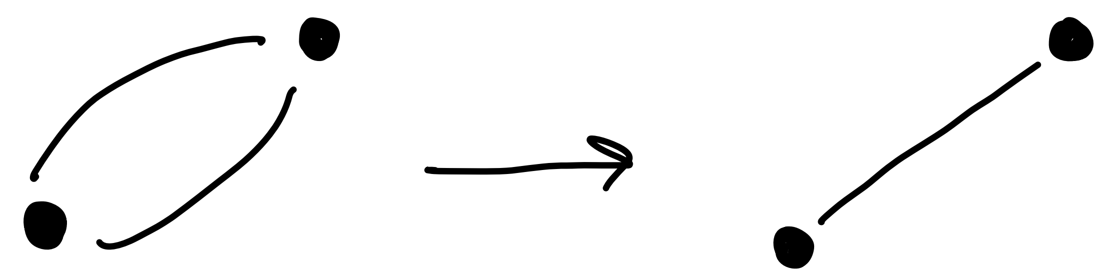
4. $f_V : V \rightharpoonup V'$ is injective,

5. the component f_O is injective,

6. the pair (f_V, f_E) forms a flag bijection between the underlying graphs in $\mathbf{B}$.

Appendix B

a morphism in $\underline{G}$ that is flag-surjective but not flag-injective:



identity graph: