A Category of Plane Graphs

Malin Altenmüller, Ross Duncan

malin.altenmuller@strath.ac.uk

SYCO 9, September 2022
Why Graphs?

- string diagrams as syntax for monoidal categories
  – we can draw pictures!
- want a combinatorial representation to reason about them
- use (some form of) graphs and their morphisms

Remark

*here: vertices represent generators, edges represent wires*
Why Plane Graphs?

• graphs are \((V, E)\), it’s all sets
• drawings contain information about the surface as well
• in SMC the surface doesn’t matter:

\[
\begin{array}{c}
\text{\includegraphics[width=0.2\textwidth]{example.png}} \\
= \\
\text{\includegraphics[width=0.2\textwidth]{example2.png}} \\
\end{array}
\]

• other monoidal theories may require non-trivial topology
• example: string diagrams for quantum processes
What are Surface Embeddings?

• Plane graph embeddings
  = drawing in the plane (or on the sphere)

• A graph is planar if it has a plane embedding
• (similarly for higher genus surfaces)
• we work at the level of embedding of a graph
Representing Graph Embeddings

*Rotation Systems* fix order of edges around vertices

Theorem

*Rotation systems uniquely determine a graph embedding.*

Our Plan

*construct a category of graphs, then add rotation information*
An Example of DPO Rewriting

given: rewrite rule

![Diagram](image-url)
An Example of DPO Rewriting

given: rewrite rule as span with common boundary in the middle
An Example of DPO Rewriting

given: matching of the LHS within a graph
An Example of DPO Rewriting

construct: context graph by pushout complement
An Example of DPO Rewriting

construct: final graph by pushout
DPO Rewriting

\[
\begin{align*}
L & \leftarrow B \rightarrow R \\
\downarrow & \downarrow \downarrow \\
G & \leftarrow G \setminus L \rightarrow G[R/L]
\end{align*}
\]

- rewrite rules are \( L \Rightarrow R \), with common boundary \( B \)
- double-pushout diagram, all maps are embeddings
- need: pushouts, pushout complements, notion of embedding
DPO Rewriting

\[
\begin{array}{ccc}
L & \leftarrow & B \\
\downarrow & & \downarrow \\
G & \leftarrow & G \setminus L \\
& & \rightarrow \\
& & G[R/L]
\end{array}
\]

• rewrite rules are \( L \Rightarrow R \), with common boundary \( B \)
• double-pushout diagram, all maps are embeddings
• need: pushouts, pushout complements, notion of embedding
• \( C = G \setminus L \): context with a hole
DPO Rewriting

- rewrite rules are $L \Rightarrow R$, with common boundary $B$
- double-pushout diagram, all maps are embeddings
- need: pushouts, pushout complements, notion of embedding
- $C = G \setminus L$: context with a hole
- $L = G \setminus C$: LHS with a “hole”
Category of Graphs

We start from the standard category of graphs:

- graphs are $E \xrightarrow{s} V \xleftarrow{t}

- morphisms are pairs of edge map $f_E$ and vertex map $f_V$ s.t.

\[
\begin{align*}
E & \xrightarrow{f_E} E' & E & \xrightarrow{f_E} E' \\
V & \xrightarrow{f_V} V' & V & \xrightarrow{f_V} V'
\end{align*}
\]

Disclaimer

(Almost) all graphs are drawn undirected in this presentation.
Open Graphs

- have to encode inputs and outputs of the diagrams
- different approaches: open graphs, representative vertices, cospans
- morphisms for open graphs don’t preserve the surface:
Boundary Vertices

- identify the “outside” of a graph
- attach input and output edges to this region
  item represent the outside with a boundary vertex

This provides:
- total graphs
- strategy to deal with the outside, and any holes in a graph
Requirements for Graph Morphisms

- vertex map needs to be partial

- cannot be injective on edges

How to define graph embeddings?
Flags

- 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: flag injectivity
Flag Surjectivity

Starting with the condition for standard graph morphisms \((V, E) \to (V', E')\):

\[
\begin{align*}
E & \xrightarrow{f_E} E' \\
V & \xrightarrow{f_V} V'
\end{align*}
\]

What about vertices with no edges attached?
Flag Surjectivity

Condition on vertices, by considering the preimage:

\[ V \xrightarrow{f_V} V' \]
\[ P(E) \xrightarrow{P(f_E)} P(E') \]

What about vertices where \( f_V \) is undefined?
Flag Surjectivity

Flag surjectivity = lax commutation of the square:

\[ V \xrightarrow{f_V} V' \]
\[ s^{-1} \quad \geq \quad \quad s'^{-1} \]
\[ P(E) \xrightarrow{P(f_E)} P(E') \]
Graphs with Circles

Objects are total graphs, as defined above

Morphisms are \((f_E, f_V)\) where

- \(f_E\) is total
- the flag map is surjective
  (no increase of flags at a vertex)

+ other conditions

Graph \textit{embeddings} are

- flag injective (no decrease of flags at a vertex)

+ other conditions

It’s a category!
Define rewriting for a specific case

**Boundary Graph**

*boundary vertex and dual boundary vertex, connected by edges:*
Partitioning Spans

partition a graph into two (connected) parts: context and subgraph
Partitioning Spans

partition a graph into two (connected) parts: context and subgraph

Theorem

*Pushouts of partitioning spans exist, and all morphisms in the pushout square are embeddings.*
Boundary Embeddings

for constructing pushout complements which give rise to partitioning spans
Boundary Embeddings

for constructing pushout complements which give rise to partitioning spans

Theorem

*Pushout complements of boundary embeddings exist and are unique (up to degeneracies).*
Remember this example?

The Same Example of DPO Rewriting
The Same Example of DPO Rewriting

Let's add some boundary regions ...
The Same Example of DPO Rewriting

...and use their representative vertices
Category of Rotation Systems

obj: graphs + cyclic ordering of flags for all vertices

arr: same as graphs + order preservation condition

Example

\[
\begin{array}{ccc}
V & \xrightarrow{f_{V}} & V' \\
\downarrow t^{-1} & \geq & \downarrow t'^{-1} \\
\text{P}(E) & \xrightarrow{P(f_E)} & \text{P}(E')
\end{array}
\]

\[
\begin{array}{ccc}
V & \xrightarrow{f_{V}} & V' \\
\downarrow t^{-1} & \geq & \downarrow t'^{-1} \\
\text{CList}(E) & \xrightarrow{\text{CList}(f_E)} & \text{CList}(E')
\end{array}
\]

Theorem

*Pushouts and pushout complements are the same as in the underlying category of graphs.*
Let’s talk about Loops!

problem: construct a pushout complement of a loop
Let’s talk about Loops!

problem: construct a pushout complement of a loop

has a plane solution
Let’s talk about Loops!

problem: construct a pushout complement of a loop

has a plane solution
and a non-plane solution
Summary

• fix inputs and outputs to control topology – boundary vertices!
• restrict your rewrite rules to meaningful cases
• category of graphs with circles extendable to rotation systems

Future Thoughts

• How about surface-embedded loops?
• How about multiple boundary vertices?

Thank You for Your Attention!
Appendix: Examples

Valid morphisms:

Embeddings:
Appendix: Non-Examples

These aren’t morphisms in the category:
Appendix: Definition Graphs with Circles

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

$$
f_E : E \rightarrow E' \quad f_{EO} : E \rightarrow O' \\
f_{OE} : O \rightarrow E' \quad f_O : O \rightarrow O'
$$

The following conditions must be satisfied:

- $f_A : A \rightarrow A'$ is total;
- the component $f_{OE} : O \rightarrow E'$ is the empty function;
- the pair $(f_V, f_E)$ forms a flag surjection between the underlying graphs.

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

- $f_V : V \rightarrow V'$ is injective;
- the component $f_O$ is injective;
- the pair $(f_V, f_E)$ forms a flag bijection between the underlying graphs.
Appendix: Two regions on a sphere