Investigations into Graph-theoretical Constructions in HoTT

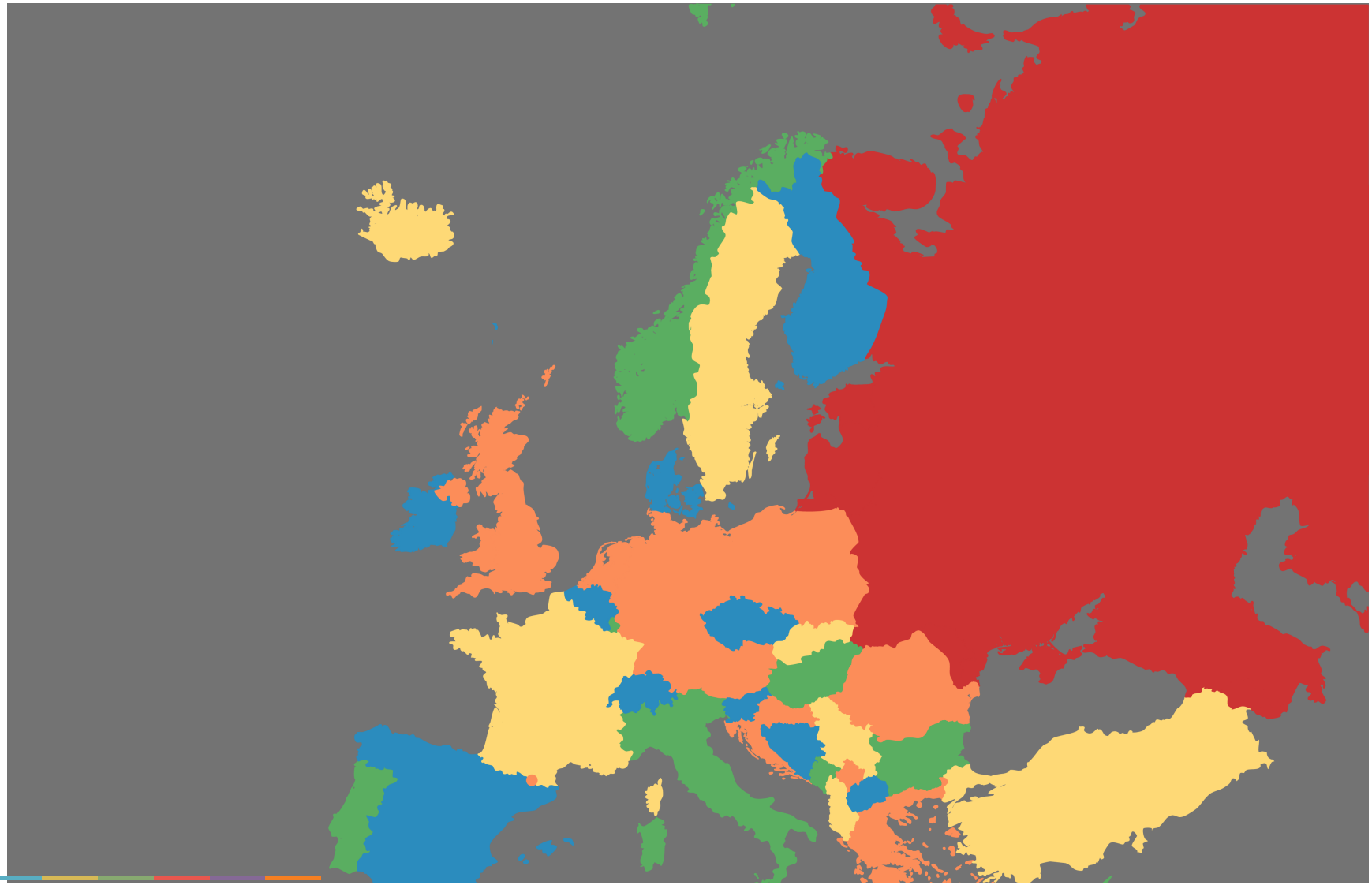
Jonathan Prieto-Cubides

December 13, 2024

UNIVERSITETET I BERGEN

Related publications I

- J. Prieto-Cubides.
  On homotopy of walks and spherical maps in homotopy type theory.
  In *Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2022, page 338–351, New York, NY, USA, 2022. Association for Computing Machinery.
- J. Prieto-Cubides.
  Mechanised proofs in Agda for the manuscript, Investigations into Graph-theoretical Constructions in Homotopy Type Theory.
  Zenodo, May 2024.
- J. Prieto-Cubides and H. R. Gylterud.
  On planarity of graphs in homotopy type theory.
  Mathematical Structures in Computer Science, 34(4):281–321, 2024.




UNIVERSITAS
BERGENSIS

What this thesis is about

- Define different concepts including directed multigraphs, combinatorial maps, faces, walk homotopy; with focus on computing their identity type.
- Provide a characterisation of planarity using a special kind of graph maps
- Show that for a (finite) graph, the following are (finite) homotopy sets:
  - Stars at nodes, combinatorial maps, spherical maps,
  - faces of any map, quasi-simple walks (walks with possibly a loop attached at the end), and
  - the collection of planar maps.
- Explain a mechanism to construct planar graphs through planar extensions.
- Study of the 1- and 2-cell geometric realization of graphs as higher inductive types.
- Provides a formal verification of some results in the proof assistant Agda.

Background

Foundations of mathematics

- Foundational crisis in the 19th century.
- Foundations of mathematics: Set theories, type theories, ETCS, among others.
- Set theories
  - *Set* as a basic unit in set theories.
  - Governed by external logic, such as first-order logic, which can be:
    - Classical such as ZFC, and that accept AC, LEM, RAA (reductio ad absurdum), or
    - Constructive such as CZF.
- Type theories
  - Constructive formal systems that move away from classical logic principles.
  - Classical reasoning can be recovered by adding AC, or any equivalent axiom.
  - Logic is via propositions-as-types correspondence, or extended versions of it.

Type theories

- *Type* represents collections of terms, such as `Bool`.
- A type is defined by formation, introduction, elimination, and computation rules. Usually presented in a natural deduction style.
- Notation: $x : A$ denotes that x is of type A.
- Extend Lambda Calculus with richer type systems, e.g. STLC, CC, MLTT, HoTT.
- *propositions-as-types* correspondence establishes a deep connection between logic and computation:
  - Logical propositions are types and
  - proofs of these propositions are terms of these types.
    - $A \wedge B$ corresponds to $A \times B$.
    - $A \vee B$ corresponds to $A + B$.
    - $A \implies B$ corresponds to $A \to B$.
    - $\bot$ corresponds to $\mathbb{0}$.
    - $\top$ corresponds to $\mathbb{1}$.
  - Example: a proof of $A \wedge B$ is a pair (a, b) where a is a proof of A and b is a proof of B.

Martin-Löf Type Theories (MLTTs)

- MLTTs form the foundation for modern constructive frameworks like HoTT, and others found in proof-assistants like Agda, Coq, and Lean.
- These theories emphasise dependent types, enabling types to be parameterized by terms of other types, and families of types indexed by other types.
- A few concepts/constructs found in MLTTs:
  - Type families: collections of types indexed by another type.
    - Example: $\mathrm{Vec}(A, n)$ for vectors of length n of elements of type A.
  - Dependent product (Π-types): functions where output type may depend on input.
    - Notation: $\Pi_{x:A} B(x)$ for functions f where $f(x) : B(x)$. $A \to B := \Pi_{(x:A)} B$.
  - Dependent sum (Σ-types): Pairs (a, b) where type of b may depend on a.
    - Notation: $\Sigma_{x:A} B(x)$ includes all such pairs. $A \times B := \Sigma_{(x:A)} B$.
- Π- and Σ-types can be seen as corresponding to universal and existential quantification, respectively.

Example: Vectors, pairs, and lists

- Notation: $\mathcal{U}$, $\mathbb{N}$ represent the universe of types and the type of natural numbers.
- Vectors as type family indexed by the type of elements and a natural number if the length of the vector.

$$\begin{aligned}
&\mathrm{Vec} : \mathcal{U} \to \mathbb{N} \to \mathcal{U}.\\
&\mathrm{Vec}(A, \mathrm{zero}) :\equiv \top.\\
&\mathrm{Vec}(A, \mathrm{suc}(n)) :\equiv \mathrm{Vec}(A, n) \times A.
\end{aligned}$$

- Pairs on A are vectors of length 2 and then, $\mathrm{Pair}(A) :\equiv \mathrm{Vec}(A, 2)$.
- List of elements of A can be defined as a pair of a natural number and a vector of elements of A.

$$\begin{aligned}
&\mathrm{List} : \mathcal{U} \to \mathcal{U}.\\
&\mathrm{List}(A) :\equiv \sum_{(n:\mathbb{N})} \mathrm{Vec}(A, n).
\end{aligned}$$

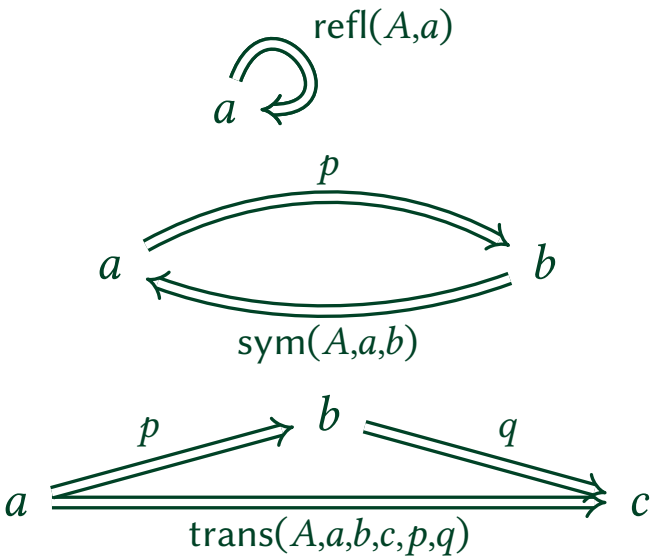
Identity types

- $\mathrm{Id}_A(x, y)$ captures the proposition that x and y are equal in A, also written as $x =_A y$.

Reflexivity : $\mathrm{Id}_A(a, a)$

Symmetry : $\mathrm{Id}_A(a, b) \to \mathrm{Id}_A(b, a)$

Transitivity : $\mathrm{Id}_A(a, b) \to \mathrm{Id}_A(b, c) \to \mathrm{Id}_A(a, c)$

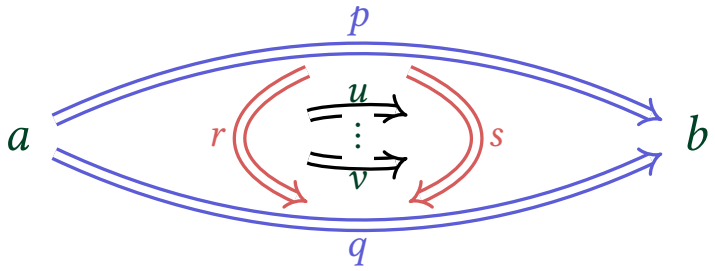


The structure of identity types

- Identity types can be understood in two ways:
  - *Extensional*: Identity types satisfy Axiom K.
    - Axiom K identifies propositional and definitional equality.
    - Identity types are propositions.
  - *Intensional*: Identity types may have multiple elements.
    - This is the approach taken in HoTT.
    - When using Agda for HoTT, we must use `--without-K` flag.

HoTT and the homotopy interpretation

- Homotopy interpretation:
  - we interpret a type A as a space,
  - an element x of type A will be seen as a point of A,
  - an identity proof p in $\mathrm{Id}_A(x, y)$ as a path from x to y, and
  - a function $f : A \to B$ will be seen as a continuous function from A to B.
- Identity types are ∞-groupoids.



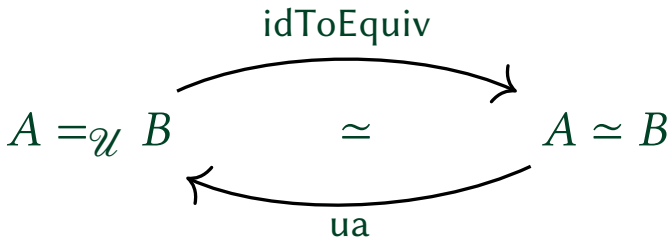
HoTT and the homotopy interpretation

- Homotopy interpretation:
  - we interpret a type A as a space,
  - an element x of type A will be seen as a point of A,
  - an identity proof p in $\mathrm{Id}_A(x, y)$ as a path from x to y, and
  - a function $f : A \to B$ will be seen as a continuous function from A to B.
- Identity types are ∞-groupoids.
- Let $n : \mathbb{N}$, $n \geq -2$. A type A is called *n-type* when is-level(n, A).
  - is-level$(-2, A) :\equiv \sum_{(c:A)} \prod_{(x:A)} (c = x)$
  - is-level$(-1, A) :\equiv \prod_{(x,y:A)} (x = y)$
  - is-level$(n + 1, A) :\equiv \prod_{(x,y:A)}$ is-level(n, A).

n	-2	-1	0	1
is-level(n, A)	isContr(A)	isProp(A)	isSet(A)	isGroupoid(A)

HoTT := MLTT + UA + HITs

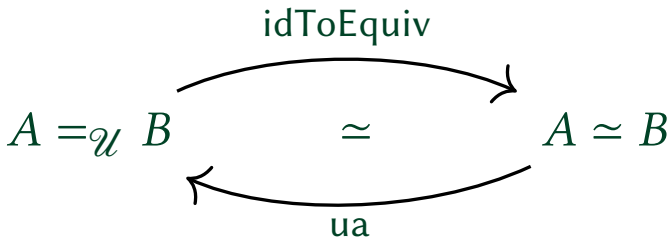
- Under UA, equivalences between types are promoted to equalities.
- In particular, data structures that are equivalent are considered equal (SIP).



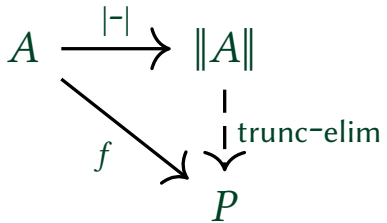
- Some HITs include:
  - Propositional truncation of a type A denoted by $\|A\|$ is the *universal solution* to the problem of mapping A to a proposition P.
  - The circle, the interval, the 2-sphere, suspension.

HoTT := MLTT + UA + HITs

- Under UA, equivalences between types are promoted to equalities.
- In particular, data structures that are equivalent are considered equal (SIP).



- Some HITs include:
  - Propositional truncation of a type A denoted by $\|A\|$ is the *universal solution* to the problem of mapping A to a proposition P.



  - $P \vee Q :\equiv \|P + Q\|, \quad P \wedge Q :\equiv \|P \times Q\|, \quad \exists(x : A)P(x) :\equiv \|\Sigma_{(x:A)}P(x)\|.$

Chapter 3: Graphs

Graphs

- A (directed multigraph) *graph* is a term of type Graph:

$$\mathsf{Graph} :\equiv \sum_{(\mathsf{N} : \mathcal{U})} \sum_{(\mathsf{E} : \mathsf{N}\to\mathsf{N}\to\mathcal{U})} \mathsf{isSet}(\mathsf{N}) \times \prod_{(x,y : \mathsf{N})} \mathsf{isSet}(\mathsf{E}(x,y)).$$

- A walk in a graph G from x to y is a sequence of connected edges.

$$\begin{aligned}
&\textbf{data}\ \mathsf{W} &&: \mathsf{N}_G \to \mathsf{N}_G \to \mathcal{U} \\
&\quad \langle _ \rangle &&: (x : \mathsf{N}_G) \to \mathsf{W}_G(x,x) \\
&\quad (_ \odot _) &&: \Pi\{x\,y\,z : \mathsf{N}_G\}.(e : \mathsf{E}_G(x,y))\,(w : \mathsf{W}_G(y,z)) \to \mathsf{W}_G(x,z).
\end{aligned}$$

- A connected graph is a graph where there is a walk between any two nodes.

$$\mathsf{Connected}(G) :\equiv \prod_{(x,y : \mathsf{N}_G)} \|W_G(x,y)\|.$$

Path graphs

- The *path graph* with n nodes is the non-connected graph P_n, defined as

$$P_n :\equiv (\llbracket n \rrbracket, \lambda\, u\, v.\mathrm{toNat}(u) + 1 = \mathrm{toNat}(v)),$$

where

$$\begin{aligned} &\mathrm{toNat} : \llbracket n \rrbracket \to \mathbb{N}. \\ &\mathrm{toNat}(k, !) :\equiv k. \end{aligned}$$

P_0 : $\varnothing$

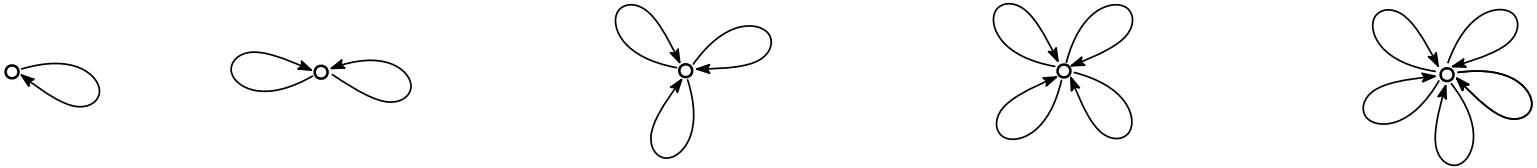
P_1 : $\bullet$

P_2 : $\bullet \longrightarrow \bullet$

P_{n+1} : $\bullet \longrightarrow \bullet \longrightarrow \cdots \longrightarrow \bullet$

Bouquet

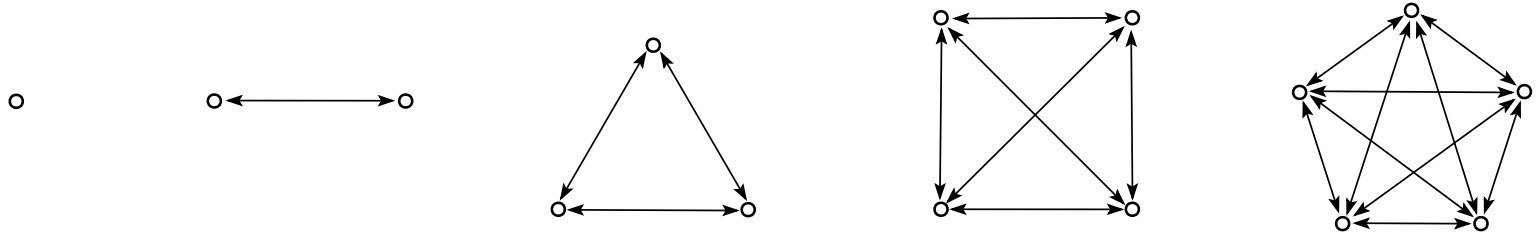
- The family of *bouquet* graphs $B_n :\equiv (\mathbb{1}, \lambda\, u\, v.[\![n]\!])$, consists of graphs obtained by considering a single point with n self-loops.



B_1 B_2 B_3 B_4 B_5

- A graph of n nodes is called *complete* when every pair of distinct nodes is joined by an edge. The complete standard graph with node set $[\![n]\!]$ is denoted by K_n.

$$K_n :\equiv ([\![n]\!], \lambda\, u\, v.u \neq v).$$

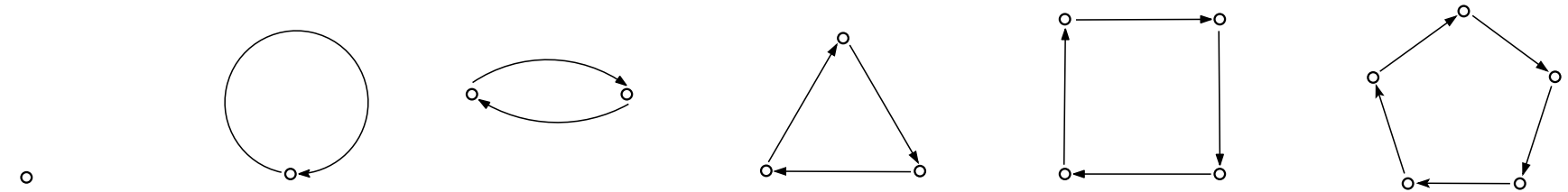


K_1 K_2 K_3 K_4 K_5

n-cycle graphs and cyclic graphs

- An *n-cycle graph* denoted by C_n is a graph with n edges defined as

$$C_n :\equiv (\llbracket n \rrbracket, \lambda\, u\, v.u = \mathrm{pred}(v), !, !).$$



- A graph G is considered to be *cyclic* if the type

$$\mathrm{CyclicGraph}(G) :\equiv \sum_{(\varphi\ :\ \mathrm{Hom}(G,G))} \sum_{(n\ :\ \mathbb{N})} \mathrm{isCyclic}(G, \varphi, n), \tag{1}$$

is inhabited, where

$$\mathrm{isCyclic}(G, \varphi, n) :\equiv \|(G, \varphi) = (C_n, \mathrm{rot})\|. \tag{2}$$

Graph homomorphisms

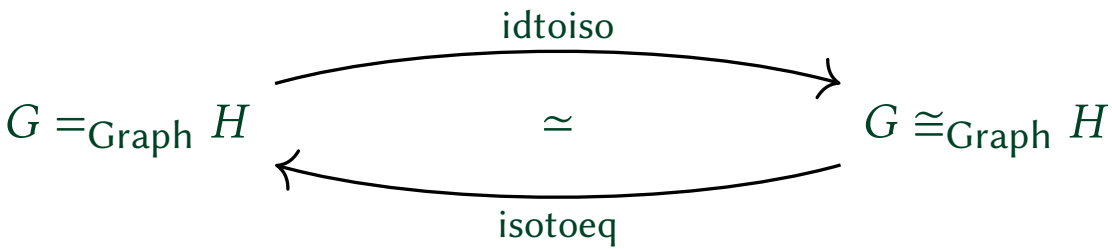
- The type of graph homomorphisms is given by:

$$\mathsf{Hom}(G, H) :\equiv \sum_{(\alpha\ :\ \mathsf{N}_G \to \mathsf{N}_H)} \prod_{(x,y\ :\ \mathsf{N}_G)} \mathsf{E}_G(x, y) \to \mathsf{E}_H(\alpha(x), \alpha(y)).$$

- The type of graph isomorphisms is given by:

$$(G \cong H) :\equiv \sum_{(\alpha\ :\ \mathsf{N}_G \simeq \mathsf{N}_H)} \prod_{(x,y\ :\ \mathsf{N}_G)} \mathsf{E}_G(x, y) \simeq \mathsf{E}_H(\alpha(x), \alpha(y)).$$

- Graph isomorphism as defined above adheres to SIP:



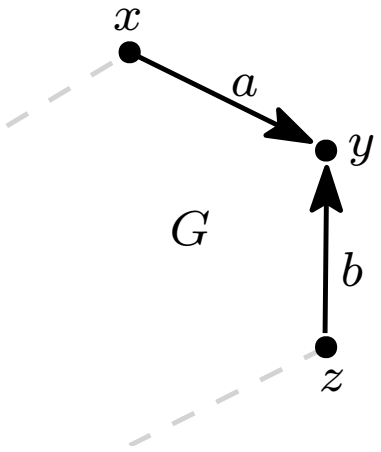
Chapter 4: Graph maps

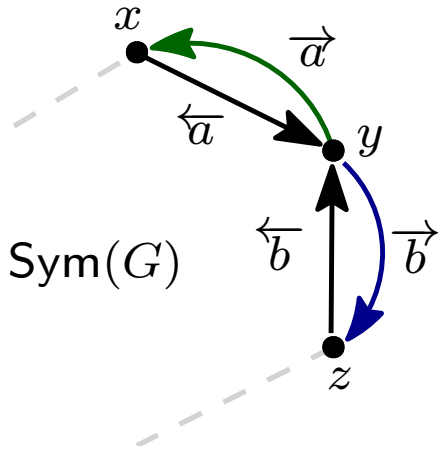
Symmetrisation

- The *symmetrisation* of a graph G is the graph $\mathsf{Sym}(G)$ defined as follows.

$$\mathsf{Sym} : \mathsf{Graph} \to \mathsf{Graph}.$$
$$\mathsf{Sym}\,(G) :\equiv (\mathsf{N}_G, \lambda xy.\mathsf{E}_G(x,y) + \mathsf{E}_G(y,x), p_G, r(q_G)),$$

where r is a proof that the coproduct $\mathsf{E}_G(x,y) + \mathsf{E}_G(y,x)$ is a set using q_G as a proof that $\mathsf{E}_G(x,y)$ is a set for all $x, y : \mathsf{N}_G$.





Stars and maps

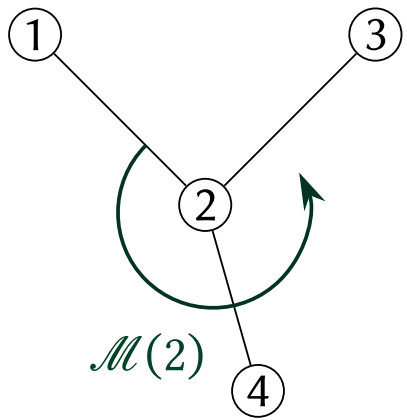
- The *star* at a node x in a graph G is the type $\mathsf{Star}_G(x)$.

$$\mathsf{Star}_G(x) :\equiv \sum_{(y\,:\,\mathsf{N}_G)} \mathsf{E}_{\mathsf{Sym}(G)}(x, y).$$

- Encode embeddings of graphs into surfaces via rotation systems:

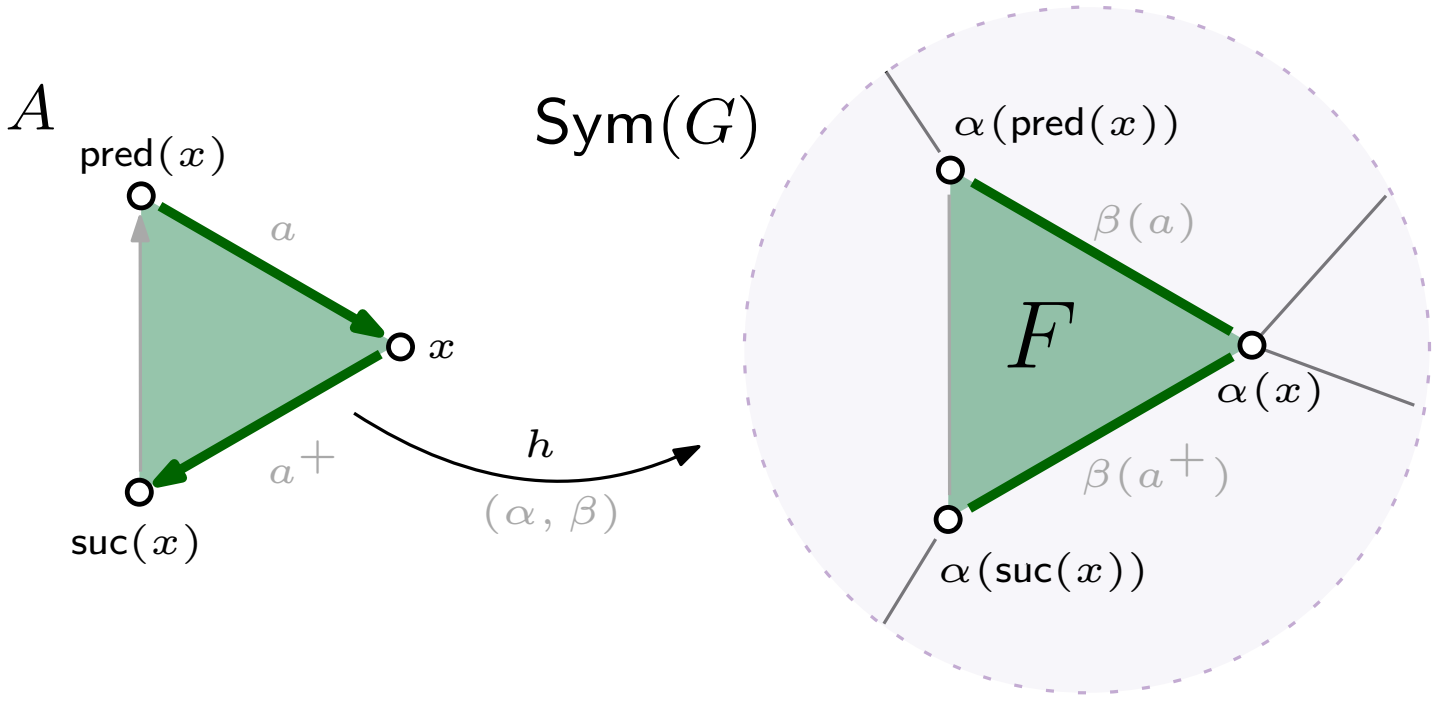
$$\mathsf{Map}(G) :\equiv \prod_{(x\,:\,\mathsf{N}_G)} \mathsf{Cyclic}(\mathsf{Star}_G(x)).$$

Node	Adj. nodes	$\mathscr{M}$
1	2, 3	[2, 3]
2	1, 3, 4	[1, 4, 3]
3	1, 2, 5	[1, 2, 5]
4	2, 5	[2, 5]
5	3, 4	[3, 4]



Faces of a map

- Regions bounded by edges in an embedded graph.
- Determined by tracing boundary walks.



The type of faces of a map

A face of type $\mathrm{Face}(G, \mathscr{M})$ consists of:

- a cyclic graph A,
- a graph homomorphism h given by (α, β) of type $\mathrm{Hom}(A, \mathrm{Sym}(G))$, such that
  - h is *edge-injective*,if the function f defined below is an embedding.

$$f : \sum_{(x,y\ :\ \mathsf{N}_G)} \mathsf{E}_G(x,y) \to \sum_{(x,y\ :\ \mathsf{N}_H)} \mathsf{E}_H(x,y).$$

$$f((x,y,e)) :\equiv (\alpha(x), \alpha(y), \beta(x,y,e)).$$

  - h is *map-compatible*, denoted by $\mathrm{isMapComp}(h)$.
    - h is *star-compatible*: $\mathrm{isStarComp}(h)(x)$ defined as

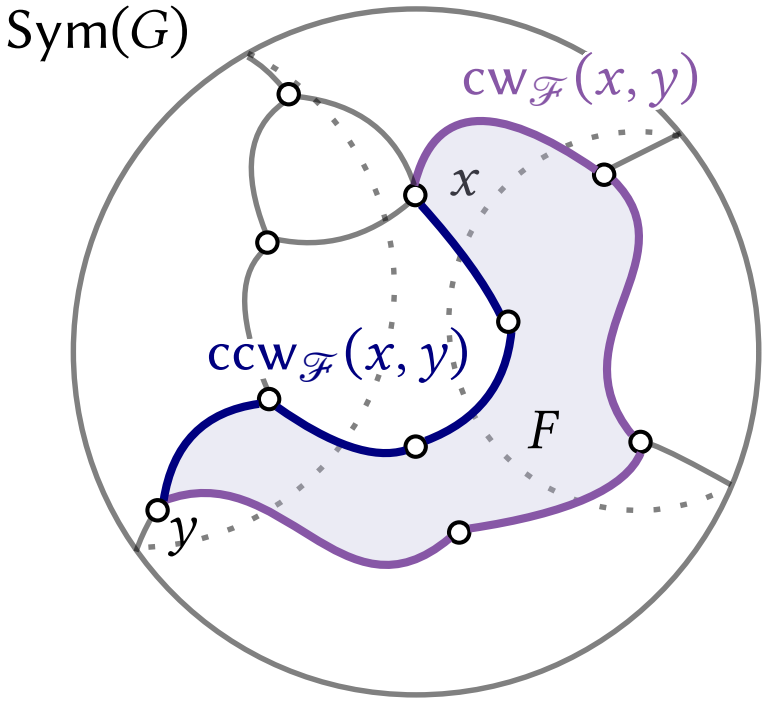
$$\|\mathrm{Star}_G(\alpha(x))\| \to \|\mathrm{Star}_A(x)\|.$$

    - h is *corner-compatible*: $\mathrm{isCornerComp}(h)(x)$ defined as

$$\mathscr{M}(\alpha(x))((\alpha(\mathrm{pred}(x)), \mathrm{flip}(\beta(\mathrm{pred}(x), x, a)))) =_{\mathrm{Star}_G(\alpha(x))} (\,\alpha(\mathrm{suc}(x))\,,\ \beta(x, \mathrm{suc}(x), a^+)\,).$$

Walking around a face

- On a face, we can always walk around in two directions and form a walk.
- In particular, we can walk around in a clockwise direction and form the walk $\mathrm{cw}_{\mathcal{F}}(a, b)$.
- We can also walk around in a counterclockwise direction and form the walk $\mathrm{ccw}_{\mathcal{F}}(a, b)$.



Chapter 5: Spherical maps and walk homotopy

The notion of walk homotopy $(\sim_{\mathscr{M}})$

- Let $w_1, w_2 : \mathrm{W}_{\mathrm{Sym}(G)}(x, y)$. Let $\mathscr{M}$ be a map. The expression $w_1 \sim_{\mathscr{M}} w_2$ denotes that one can *deform* w_1 into w_2 along the faces of $\mathscr{M}$.
- The relation $(\sim_{\mathscr{M}})$ has the following four constructors:

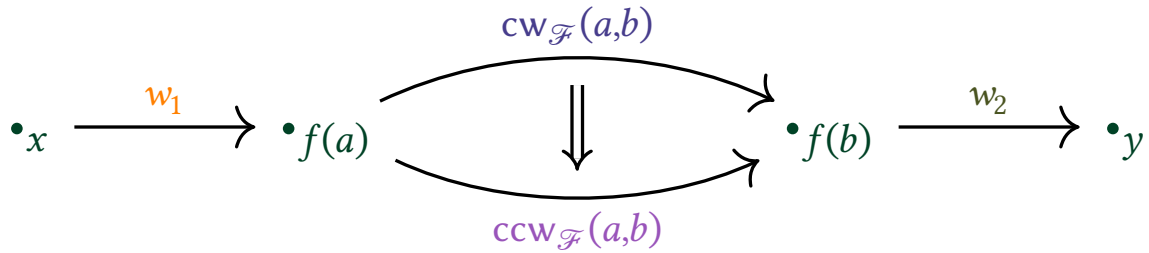
$$\mathrm{hrefl} : \prod_{(w_1 : \mathrm{W}_{\mathrm{Sym}(G)}(x,y))} w_1 \sim_{\mathscr{M}} w_1.$$

$$\mathrm{hsym} : \prod_{(w_1, w_2 : \mathrm{W}_{\mathrm{Sym}(G)}(x,y))} w_1 \sim_{\mathscr{M}} w_2 \to w_2 \sim_{\mathscr{M}} w_1.$$

$$\mathrm{htrans} : \prod_{(w_1, w_2, w_3 : \mathrm{W}_{\mathrm{Sym}(G)}(x,y))} w_1 \sim_{\mathscr{M}} w_2 \to w_2 \sim_{\mathscr{M}} w_3 \to w_1 \sim_{\mathscr{M}} w_3.$$

$$\mathrm{hcollapse} : \cdots \text{(walks along faces are walk homotopic)}$$

The hcollapse constructor in $(\sim_{\mathscr{M}})$



$$(w_1 \cdot \mathrm{cw}_{\mathscr{F}}(a,b) \cdot w_2) \sim_{\mathscr{M}} (w_1 \cdot \mathrm{ccw}_{\mathscr{F}}(a,b) \cdot w_2),$$

supposing one has the following,

(i) a face $\mathscr{F}$ given by $\langle A, f \rangle$ of the map $\mathscr{M}$,

(ii) a walk w_1 of type $\mathrm{W}_{\mathrm{Sym}(G)}(x, f(a))$ for a node x in G with a node a in A, and

(iii) a walk w_2 of type $\mathrm{W}_{\mathrm{Sym}(G)}(f(b), y)$ for a node b in A with a node y in G.

Whiskering

1. Right whiskering: If $w_1 \sim_{\mathscr{M}} w_2$ then $(w_1 \cdot w_3) \sim_{\mathscr{M}} (w_2 \cdot w_3)$.

2. Left whiskering: If $p_1 \sim_{\mathscr{M}} p_2$ then $(w \cdot p_1) \sim_{\mathscr{M}} (w \cdot p_2)$.

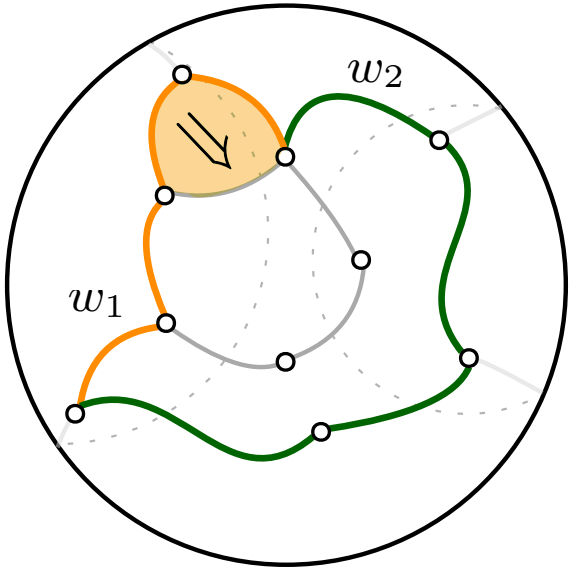
3. Horizontal composition: If $w_1 \sim_{\mathscr{M}} w_2$ and $p_1 \sim_{\mathscr{M}} p_2$, then $(w_1 \cdot p_1) \sim_{\mathscr{M}} (w_2 \cdot p_2)$.

Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 1:

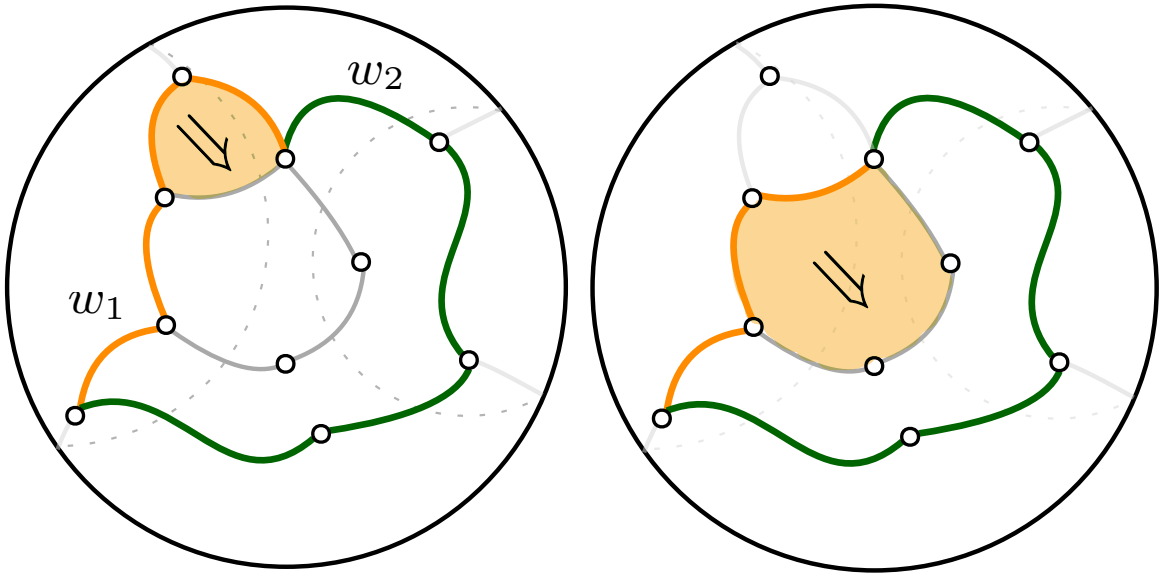


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 1:

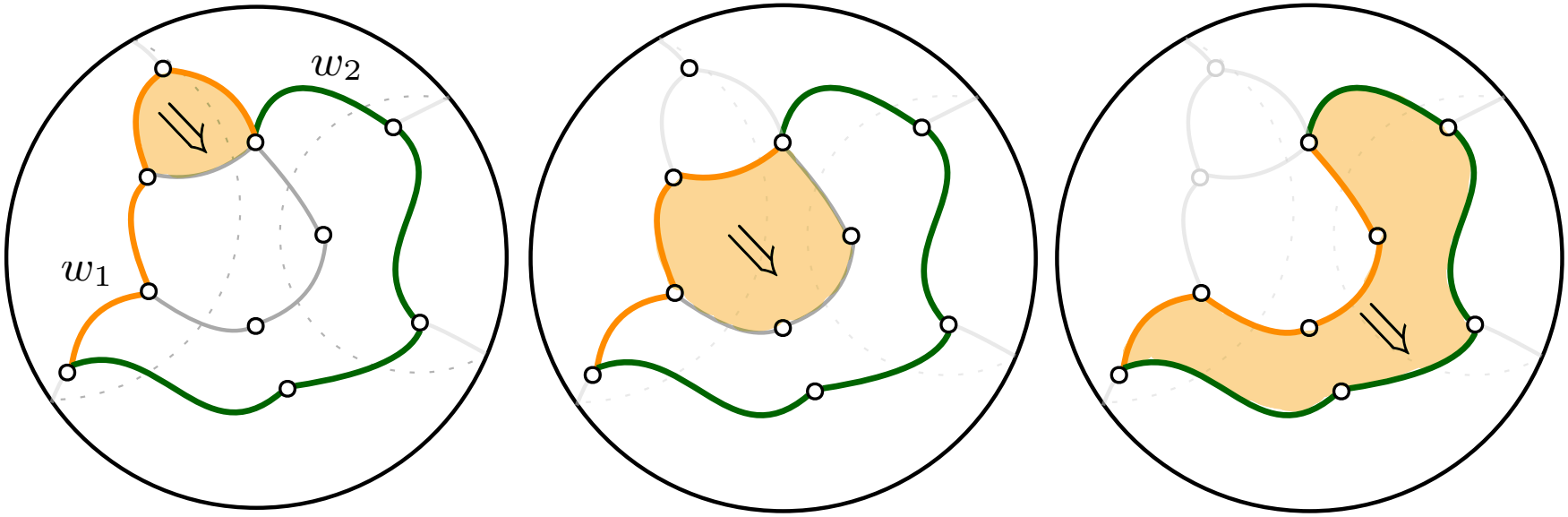


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 1:

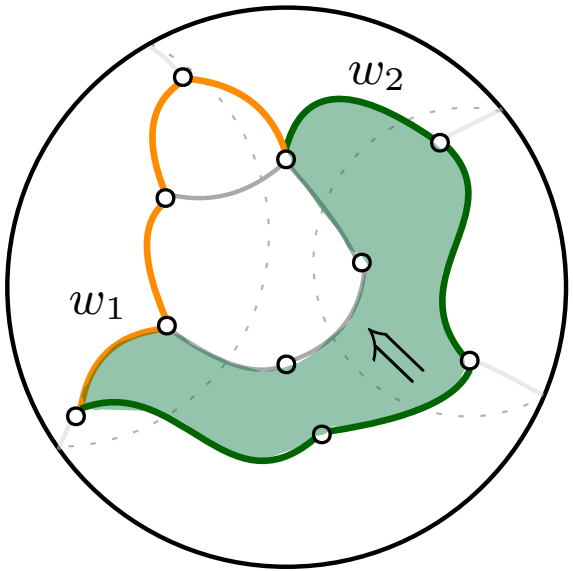


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 2:

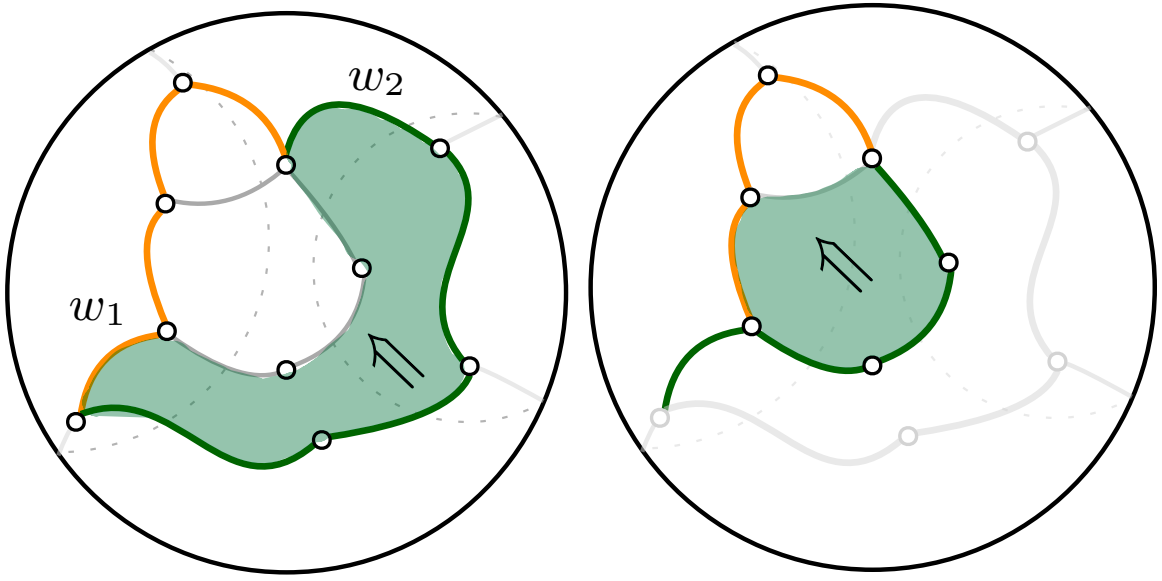


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 2:

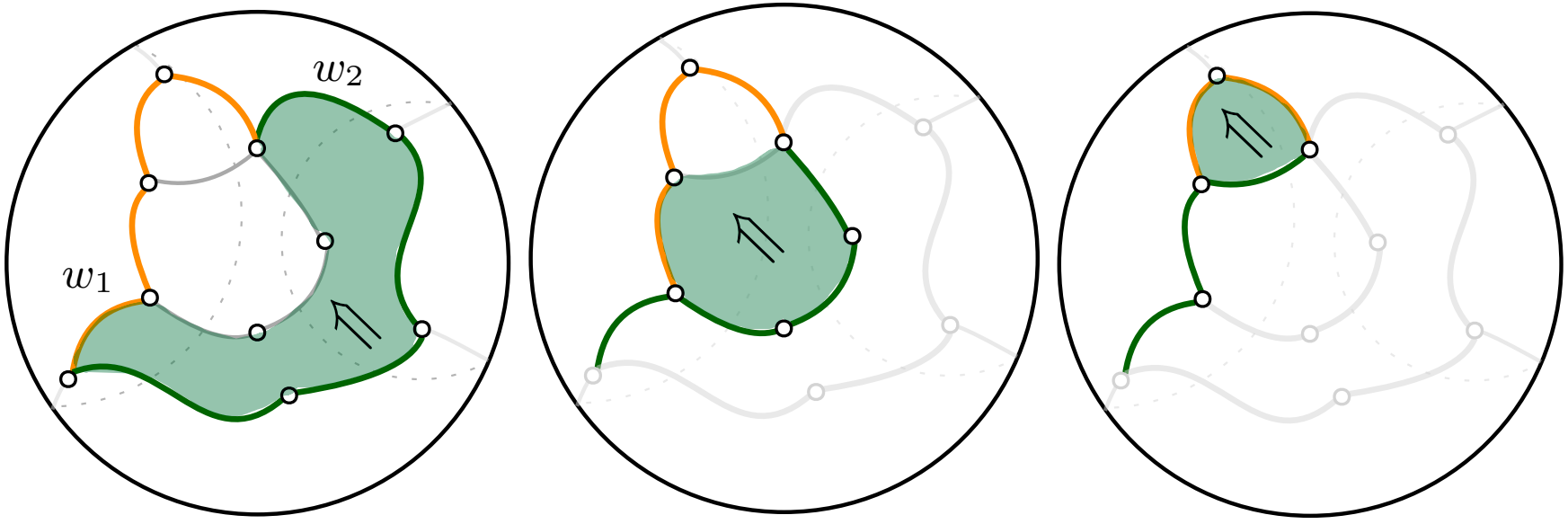


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 2:

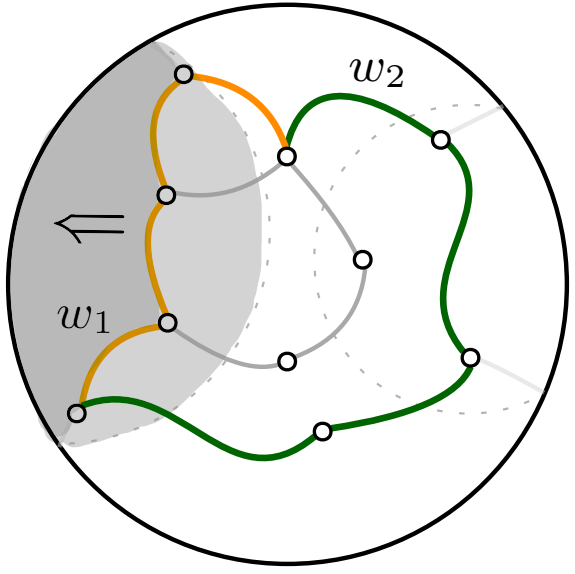


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 3:

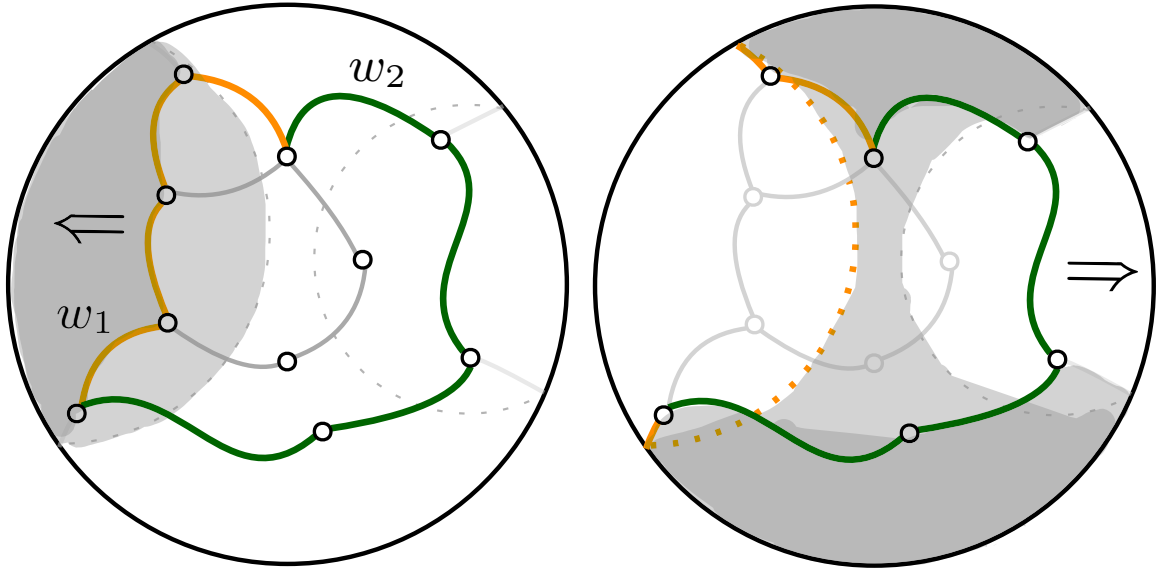


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 3:

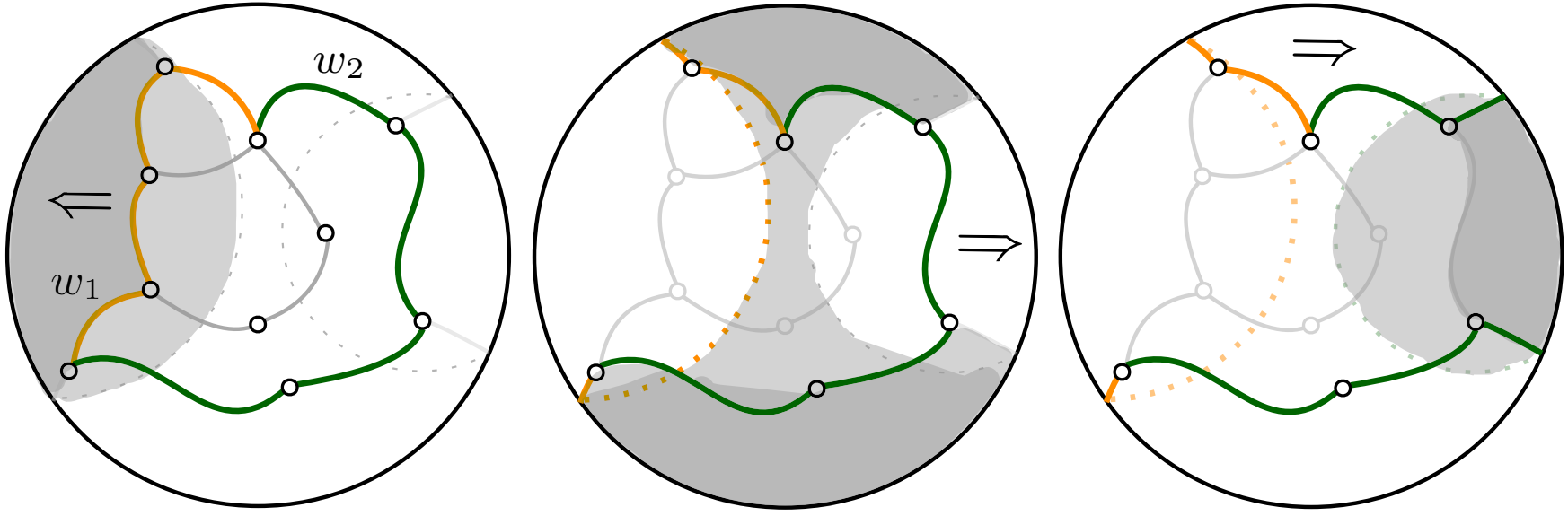


Spherical maps

- A map $\mathscr{M}$ for a graph G is said to be *spherical* if isSpherical($\mathscr{M}$) is inhabited.

$$\mathsf{isSpherical}(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_{\mathsf{Sym}(G)})} \prod_{(w_1,w_2:\mathsf{W}_{\mathsf{Sym}(G)}(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

Option 3:

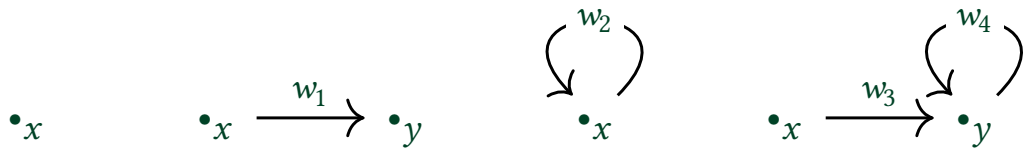


A refinement for spherical maps

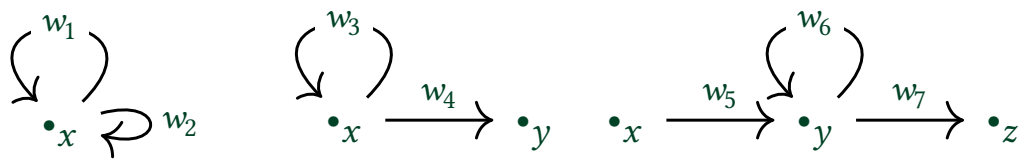
How to construct terms of type Planar(G) for some given graph G?

$$\mathsf{isSpherical}_2(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_G)} \prod_{(w_1,w_2:\mathsf{W}_G(x,y))} \mathsf{isQuasi}(w_1) \times \mathsf{isQuasi}(w_2) \to \|w_1 \sim_{\mathscr{M}} w_2\|.$$

- Examples of walks that are quasi-simple



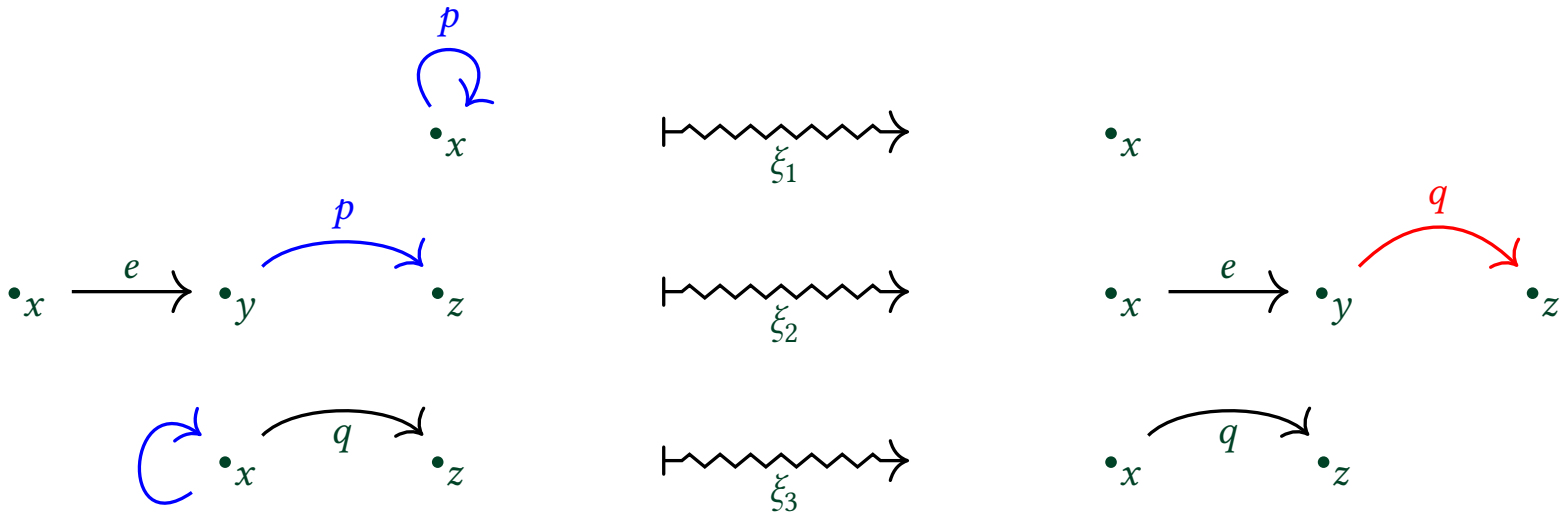
- Examples of walks that are **not** quasi-simple



Quasi-simple walks

Assuming the node set is discrete, let $x, y, z : \mathrm{N}_G$ and $w : \mathrm{W}_G(x, z)$.

- Being quasi-simple is: $\mathrm{isQuasi}(w) :\equiv \prod_{(z:\mathrm{N}_G)} \mathrm{isProp}(z \in w)$, where the relation $(\in)$ is defined as:
  - $z \in \langle y \rangle :\equiv \bot$.
  - $z \in (e \odot w) :\equiv (z = \mathrm{source}(e)) + (z \in w)$.
- One can define a relation on walks $(\rightsquigarrow)$ to remove loops.



Some results on quasi-simple walks

- Each walk p is normal if it is quasi-simple and not a loop.

$$\mathrm{Normal}(p) :\equiv \mathrm{isQuasi}(p) \times \neg \sum_{(q : \mathrm{W}_G(x,y))} (p \rightsquigarrow\!\!\!\!\!\rightarrow q).$$

- Thm. $(\in)$, isQuasi, Normal are all decidable propositions when the node set is discrete.
- Thm. Normal forms for any walk. The type $\sum_{(v : \mathrm{W}_G(x,z))} (w \rightsquigarrow\!\!\!\!\!\rightarrow^* v) \times \mathrm{Normal}(v)$ is inhabited for any walk w.
- Thm. Given a spherical map $\mathscr{M}$, the type

$$\sum_{(v : \mathrm{W}_G(x,z))} (w \rightsquigarrow\!\!\!\!\!\rightarrow^* v) \times \mathrm{Normal}(v) \times \|w \sim_{\mathscr{M}} v\|$$

  is inhabited for any walk w.

Loops are redundant

- Thm. Both characterisations of being a spherical map are equivalent if the node set is discrete, that is, for any map $\mathscr{M}$ of a graph G,

$$\mathsf{isSpherical}_1(\mathscr{M}) \simeq \mathsf{isSpherical}_2(\mathscr{M})$$

where

$$\mathsf{isSpherical}_1(\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_G)} \prod_{(w_1,w_2:\mathsf{W}_G(x,y))} \| w_1 \sim_{\mathscr{M}} w_2 \|$$

and

$$\mathsf{isSpherical}_2(G,\mathscr{M}) :\equiv \prod_{(x,y:\mathsf{N}_G)} \prod_{(w_1,w_2:\mathsf{W}_G(x,y))} \mathsf{isQuasi}(w_1)\times\mathsf{isQuasi}(w_2) \to \| w_1 \sim_{\mathscr{M}} w_2 \| .$$

- Thm. The set of quasi-simple walks is a finite set if the graph is finite and the node set is discrete.

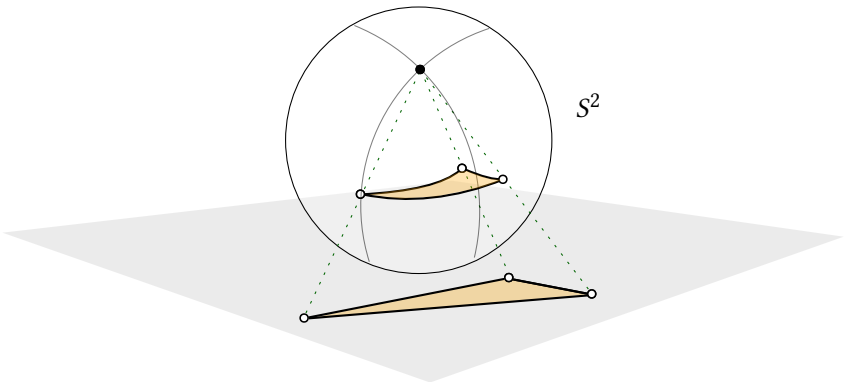
Chapter 6: Planarity

Planarity

- A connected and locally finite graph G is *planar* if the type $\mathsf{Planar}(G)$ is inhabited.

$$\mathsf{Planar}(G) :\equiv \sum_{(\mathscr{M} \,:\, \mathsf{Map}(G))} \mathsf{isSpherical}(\mathscr{M}) \times \underbrace{\mathsf{Face}(G, \mathscr{M})}_{\text{outer face}}$$

- Thm. The type of all planar maps of a (finite) graph forms a (finite) set!

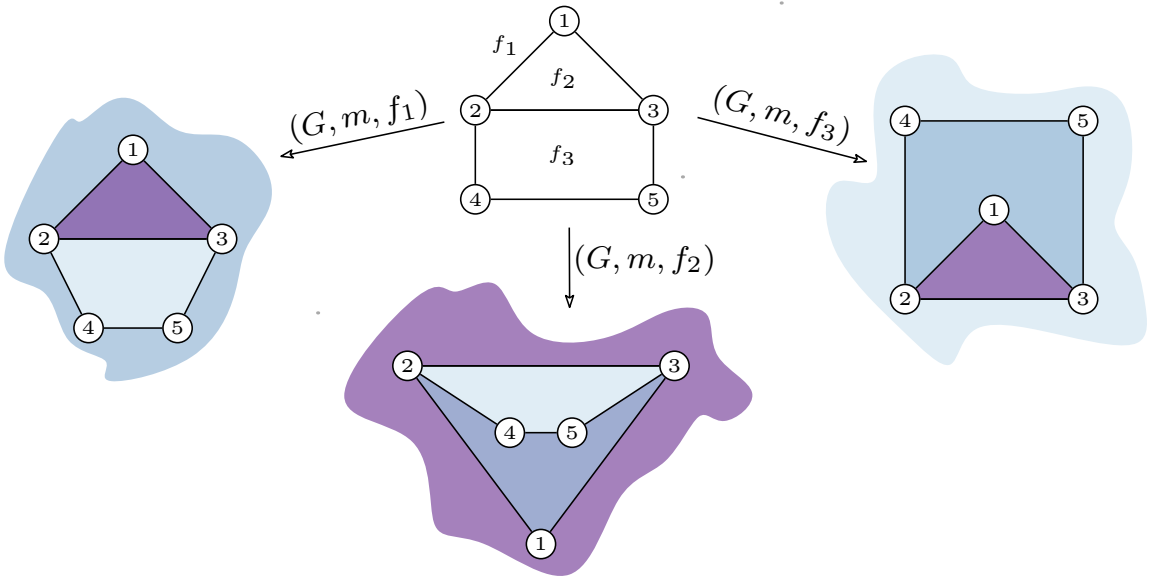


Planarity

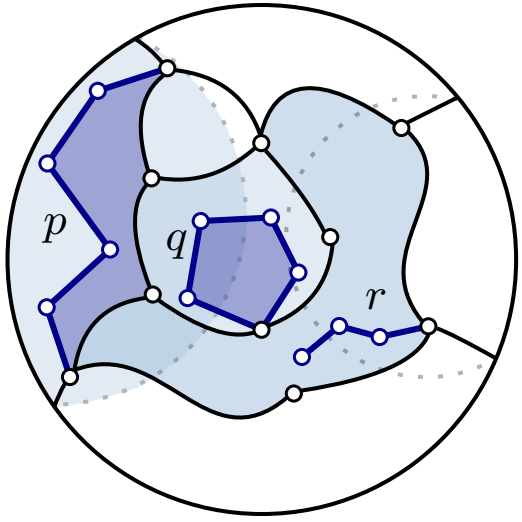
- A connected and locally finite graph G is *planar* if the type $\mathsf{Planar}(G)$ is inhabited.

$$\mathsf{Planar}(G) :\equiv \sum_{(\mathscr{M} \,:\, \mathsf{Map}(G))} \mathsf{isSpherical}(\mathscr{M}) \times \underbrace{\mathsf{Face}(G, \mathscr{M})}_{\text{outer face}}$$

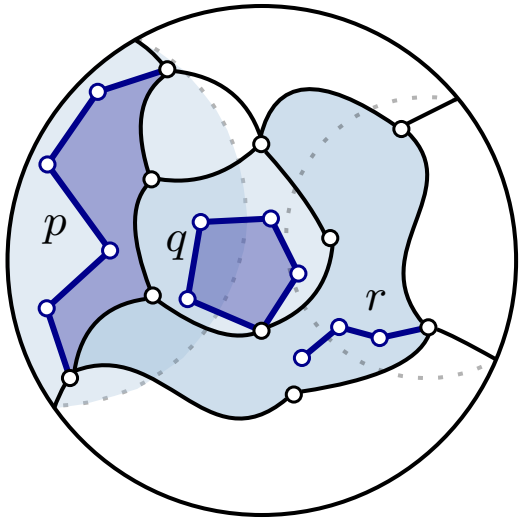
- Thm. The type of all planar maps of a (finite) graph forms a (finite) set!

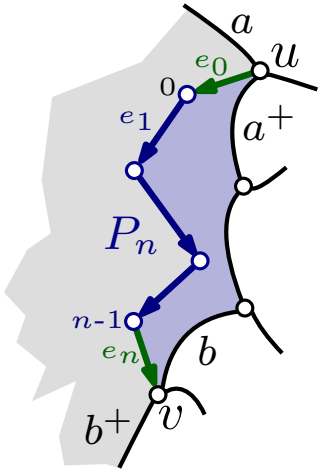


Planar extensions via path additions

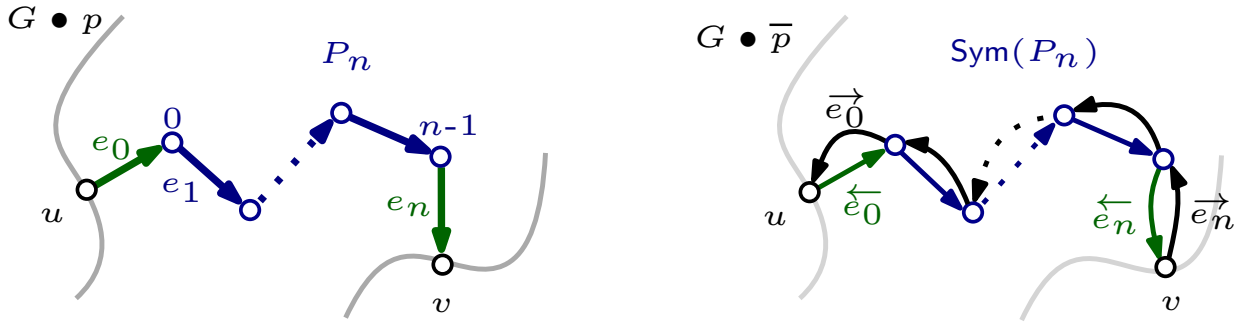


Planar extensions via path additions

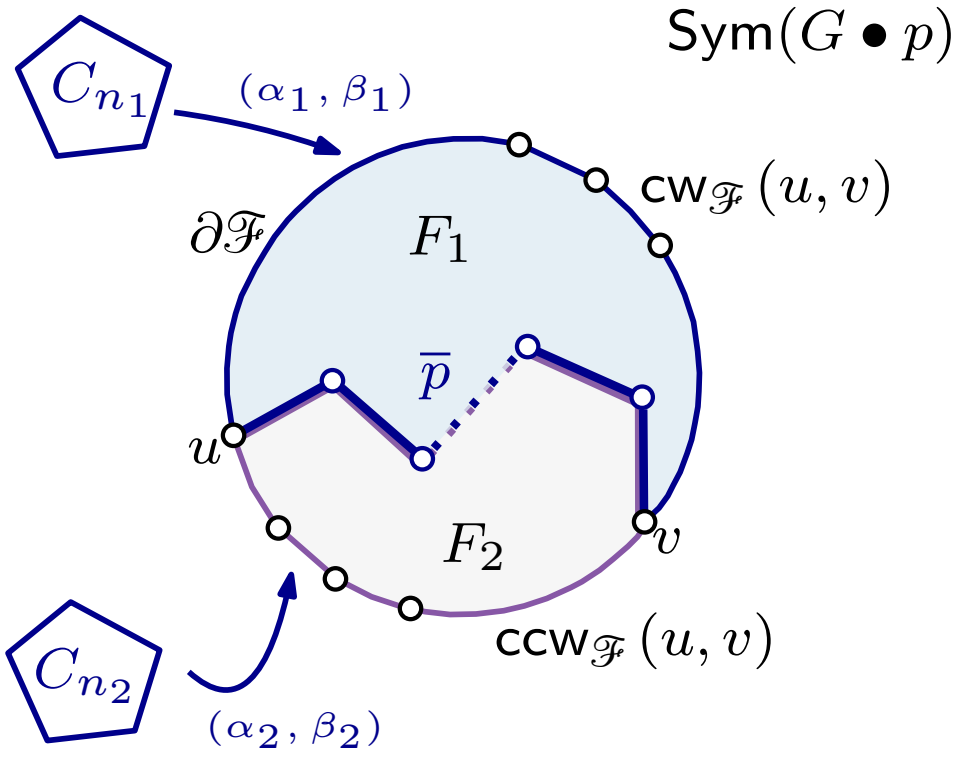




Planar extensions via path additions

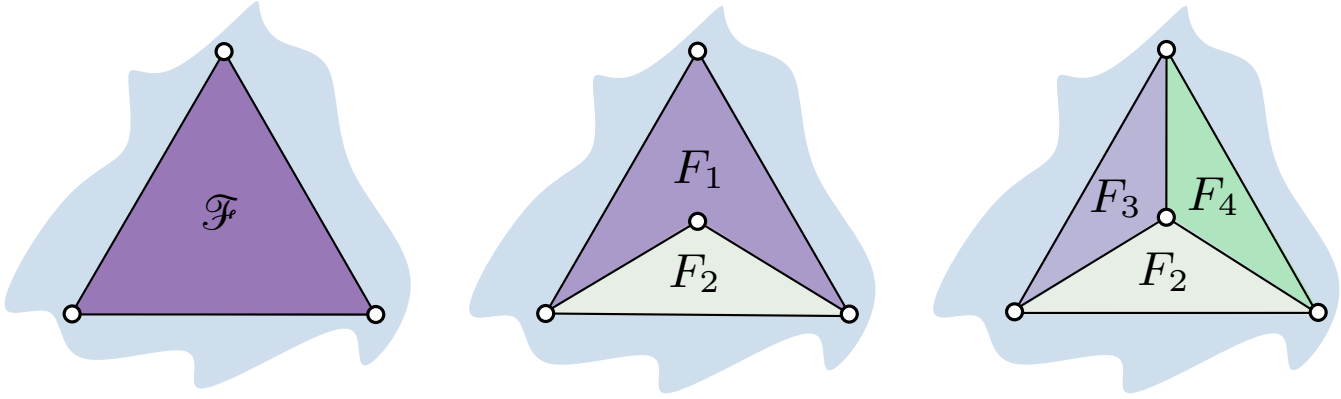


Planar extensions: division of a face



Planar extensions

- A planar map for K_4 can be constructed from a planar map for K_3.



- If G is a finite graph with a map $\mathscr{M}$, then we refer to the Euler characteristic of G by $\mathscr{M}$, denoted by $\chi_{\mathscr{M}}$, as the number associated with the cardinal of the set of nodes (v), edges (e), and faces (f).

$$\chi_{\mathscr{M}} :\equiv v - e + f.$$

- Lem: For a graph G with planar map $\mathscr{M}$, any planar extension of $\mathscr{M}$ maintains Euler's characteristic.

A few concluding remarks

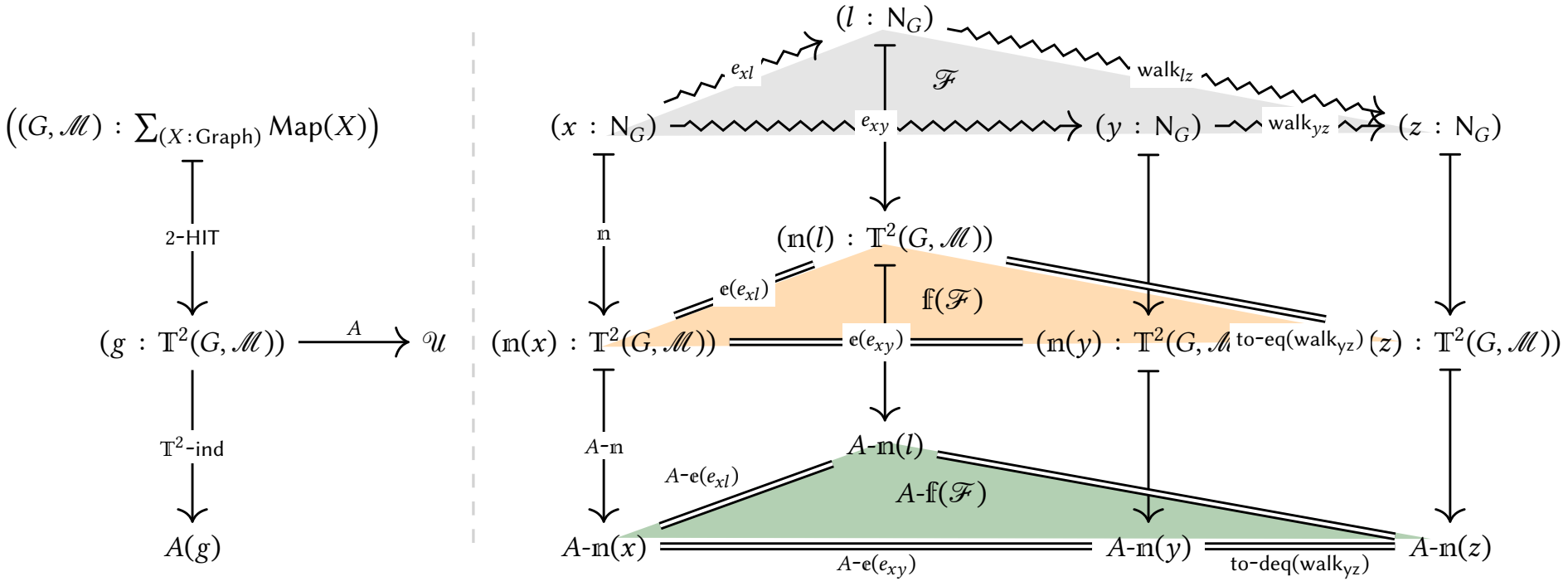
- Novel characterisation of planarity for directed multigraphs using graph maps, encoding topological intuition.
- Defines planarity as a structure rather than a property on graphs.
- The collection of planar maps of G is a (finite) set, because:
  - If G is a (finite) graph, then the following types are (finite) sets.
    - The star at x for any node x is a (finite) set,
    - the maps of G is a (finite) set,
    - the spherical maps of G is a (finite) set,
    - the faces of any map for G is a (finite) set, and
    - the walks without inner loops in G is a (finite) set.

What's next?

- Equivalence between different characterisations of planarity, e.g., Kuratowski's theorem (Forbidden minors characterisation).
- The thesis contains lemmas and conjectures about geometric realisations of graphs using higher-inductive types.
- Proving those conjectures (in Agda), in particular, would be a great step forward.
- More formalisations in Agda!

Bonus

The elimination principle for the two-level top. realisation



$\mathbb{G}^2(G) \simeq \mathbb{S}^2$

- Let G be a nonempty finite planar graph with n nodes. Then $\mathbb{G}^2(G) \simeq \mathbb{S}^2$.
- Lem. 1.
  a. $\mathbb{G}^2(\bullet) \simeq \mathbb{S}^2$.
  b. $\mathbb{G}^2(T) \simeq \mathbb{G}^2(\bullet)$ for a tree T.
- Let G be a graph with a map $\mathscr{M}$.
  - Lem. 2. Face contraction preserves planarity
  - Lem. 3. H is obtained by *contracting a face* F of $\mathscr{M}$, then $\mathbb{G}^2(G) \simeq \mathbb{G}^2(H)$.

$\mathbb{G}^2(G) \simeq \mathbb{S}^2$

- Thm. Let G be a non-empty finite planar graph with n nodes. Then $\mathbb{G}^2(G) \simeq \mathbb{S}^2$.
- Proof. Because G is a non-empty finite graph, then let m be the number of faces of $\mathscr{M}$. We proceed by induction on m.
  - Case $m = 0$. Impossible.
  - Case $m = 1$. Apply Lemma 1b. The graph G is a tree.
  - Case $m > 1$. Let F be a face of $\mathscr{M}$. By contracting the face F, one obtains a graph G' and a map $\mathscr{M}'$ such that $(G, \mathscr{M}) \rightsquigarrow_F (G', \mathscr{M}')$. Therefore, G' has $m - 1$ faces and by Lemma 3, one gets that $\mathbb{G}^2(G) \simeq \mathbb{G}^2(G')$. By Lemma 2, the map $\mathscr{M}'$ is planar. Now, if n' and k denote the number of nodes of G' and F, respectively, then $n' = n - (k - 1)$ and $k > 0$. By applying the induction hypothesis to G', $\mathscr{M}'$, an equivalence $\mathbb{G}^2(G') \simeq \mathbb{S}^2$ is obtained. Finally, the conclusion follows from the chain of equivalences:

$$\mathbb{G}^2(G) \simeq \mathbb{G}^2(G') \simeq \mathbb{S}^2.$$

Structure on a graph

Definition (Graph class)

A *class* C of graphs is given by the collection of graphs that holds some given structure $P : \mathrm{Graph} \to \mathcal{U}$

$$C :\equiv \sum_{(G : \mathrm{Graph})} P(G)$$

Examples:

- $\mathrm{isUndirected}(G) :\equiv \Pi_{x,y : \mathsf{N}_G} \mathsf{E}_G(x,y) \to \mathsf{E}_G(y,x)$
- $\mathrm{isFiniteGraph}(G) :\equiv \mathrm{isFinite}(\mathsf{N}_G) \times \prod_{x,y : \mathsf{N}_G} \mathrm{isFinite}(\mathsf{E}_G(x,y))$

Lemmas

Given $x, y, z : \mathsf{N}_G$, $e : \mathsf{E}_G(x, y)$ and a quasi-simple walk $w : \mathsf{W}_G(y, z)$,

- if $x \notin w$ then the walk $(e \odot w)$ is quasi-simple.
- if the walk $(e \odot w)$ is a quasi-simple walk then w is also a quasi-simple walk.
- if the lenght of w is n, then $[\![n]\!] \simeq \Sigma_{(y:\mathsf{N}_G)}(y \in w)$.
- If the node-set of G is discrete then
  - being quasi-simple for a walk is a decidable proposition.
  - the type $(x \in w)$ is a finite set.
- Given $x, y : \mathsf{N}_G$ and $n : \mathbb{N}$, the type qswalk collects all quasi-simple walks of a fixed length n.

$$\mathrm{qswalk}(n, x, y) :\equiv \sum_{(w:\mathsf{W}_G(x,y))} \mathrm{isQuasi}(w) \times (\mathrm{length}(w) = n).$$

- Given a graph G, $n : \mathbb{N}$, and $x, z : \mathsf{N}_G$, the following equivalence holds.

$$\mathrm{qswalk}(S(n), x, z) \simeq \sum_{(y:\mathsf{N}_G)} \sum_{(e:\mathsf{E}_G(x,y))} \sum_{(w:\mathrm{qswalk}(n,y,z))} (x \notin w)$$

Lemmas

- Given a finite graph, $x, y : \mathsf{N}_G$ and $n : \mathbb{N}$, the type $\mathrm{qswalk}(n, x, y)$ is a finite set.
- Let G be a finite graph. Then the following type is a finite set.

$$\sum_{(x,y:\mathsf{N}_G)} \sum_{(m:[\![n+1]\!])} \mathrm{qswalk}(m, x, y).$$

- Given a graph G with finite node-set, $x, y : \mathsf{N}_G$ and a quasi-simple walk $w : \mathsf{W}_G(x, y)$ of length m, then it holds that $m \leq n$.
- Given a graph G with finite node-set and $x, y : \mathsf{N}_G$, the following equivalence holds.

$$\sum_{(w:\mathsf{W}_G(x,y))} \mathrm{isQuasi}(w) \simeq \sum_{(m:[\![n+1]\!])} \mathrm{qswalk}(m, x, y).$$

- The quasi-simple walks of a finite graph G forms a finite set.

$$\sum_{(x,y:\mathsf{N}_G)} \sum_{(w:\mathsf{W}(x,y))} \mathrm{isQuasi}(w).$$

Definition

Let $x, y : \mathsf{N}_G$ and $w : \mathsf{W}_G(x, y)$.

1. The walk w is a loop whenever the head is equal to the end, i.e. $\mathsf{Loop}(w)$.

$$\begin{aligned} \text{data Loop} &: \prod \{x, y\}.\, \mathsf{W}_G(x, y) \to \mathcal{U} \\ \text{is-loop} &: x = y \to \mathsf{Loop}(w) \end{aligned}$$

2. The walk w is trivial if its length is zero, i.e. $\mathsf{Trivial}(w)$.

$$\begin{aligned} \text{data Trivial} &: \prod \{x, y\}.\, \mathsf{W}_G(x, y) \to \mathcal{U} \\ \text{is-trivial} &: \mathsf{length}(w) = 0 \to \mathsf{Trivial}(w) \end{aligned}$$

3. A walk w is not trivial, if it has one edge at least, i.e. $\mathsf{NonTrivial}(w)$.

$$\begin{aligned} \text{data NonTrivial} &: \prod \{x, y\}.\, \mathsf{W}_G(x, y) \to \mathcal{U} \\ \text{has-edge} &: \prod \{x\, y\, z\} \prod \{w : \mathsf{W}_G(y, z)\}.\, (e : \mathsf{E}_G(x, y)) \\ &\to \mathsf{NonTrivial}(e \odot w). \end{aligned}$$

- The relation $(\rightsquigarrow^*)$ is the reflexive and transitive closure of the relation $(\rightsquigarrow)$.
- Given $x, y : \mathsf{N}_G$ and $p, q : \mathsf{W}_G(x, y)$, the following claims hold:
  1. If $x \in q$ and $p \rightsquigarrow^* q$ then $x \in p$.
  2. If $p \rightsquigarrow q$ then $\mathsf{length}(q) < \mathsf{length}(p)$.
- Given a walk $p : \mathsf{W}_G(x, y)$, $\mathsf{Reduce}(p) :\equiv \Sigma_{(q:\mathsf{W}_G(x,y))}(p \rightsquigarrow q)$.
- Given a walk p, one states that p is in *normal form* if $\mathsf{Normal}(p)$. If $p \rightsquigarrow q$ and q is in normal form, we refer to q as the normal formal of p.

$$\mathsf{Normal}(p) :\equiv \mathsf{isQuasi}(p) \times \neg\, \mathsf{Reduce}(p).$$

- Being in normal form for a walk is a proposition.

Given nodes x and y in a graph G, we encode the fact a walk can reduce or not by using the inductive data type Progress.

$$\begin{aligned}\text{data Progress}\,\{x\,y\}\,(p\,:\,\mathrm{W}_G(x,y))\,:\,\mathcal{U}\\ \text{step}\;:\;\text{Reduce}(p)\to\text{Progress}(p)\\ \text{done}\;:\;\text{Normal}(p)\to\text{Progress}(p)\end{aligned}$$

Theorem (Normalisation)

- *Given a graph G with discrete node-set, there exists a reduction for each walk to one of its normal forms.*

$$\Sigma_{(v:\mathrm{W}_G(x,z))}(w \rightsquigarrow^* v)\times\text{Normal}(v).$$

- *Given a graph G and a walk w of type $\mathrm{W}_G(x,y)$ for two $x,y:\mathrm{N}_G$, the following claims hold.*
  1. *The type Reduce(w) is decidable.*
  2. *The proposition Normal(w) is decidable.*
  3. *The walk w progresses.*

Planar synthesis of graphs

A Whitney *synthesis* (synthesis for short) of graph G from graph H is defined as a sequence of graphs $G_0, G_1, \cdots, G_n$, where G_0 is H, G_n is G, and each G_i results from the path addition of p_i to G_{i-1} for i in the range 1 to n. Consequently, G can be viewed as the result of adding paths $p_1, p_2, \cdots, p_n$ to H:

$$G \equiv H \bullet p_1 \bullet p_2 \bullet \cdots \bullet p_n.$$

The *length* of this synthesis is n. A *simple synthesis* refers to a sequence containing only simple additions. Conversely, a sequence composed solely of non-simple additions is termed a *non-simple synthesis*.

Planar synthesis of graphs

- Syntheses preserve graph connectedness. Specifically, if a graph H is connected and G is synthesised from H, then each intermediate graph G_i in the synthesis sequence is also connected.
- Given a planar map $\mathscr{M}$ of the graph H with outer face $\mathscr{F}$, we define a *planar synthesis* of G from H of length n as a sequence

$$(G_0, \mathscr{M}_0, \mathscr{F}_0), (G_1, \mathscr{M}_1, \mathscr{F}_1) \cdots, (G_n, \mathscr{M}_n, \mathscr{F}_n),$$

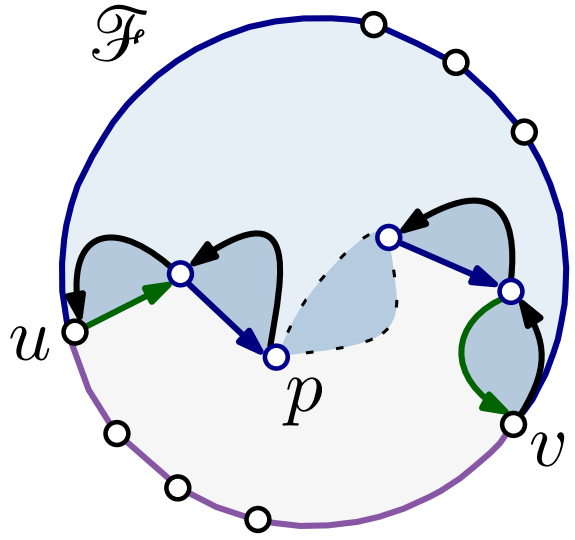
where:

  - $(G_0, \mathscr{M}_0, \mathscr{F}_0)$ is equivalent to $(H, \mathscr{M}, \mathscr{F})$, and
  - $(G_n, \mathscr{M}_n)$ corresponds to $(G, E(\mathscr{M}_{n-1}, \mathscr{F}_{n-1}, p_{n-1}))$.

For each i in the range 1 to n, the graph G_i is $G_{i-1} \bullet p_i$, and the map $\mathscr{M}_i$ is $E(\mathscr{M}_{i-1}, \mathscr{F}_{i-1}, p_{i-1})$, where $\mathscr{F}_{i-1}$ is a face of $\mathscr{M}_{i-1}$.

Planar synthesis of graphs

- If a graph G is synthesised from a planar graph H via planar synthesis, then G and every graph in the corresponding sequence are planar.



- The number of faces increases by $n + 2$.
- The number of edges increases by $2(n + 1)$
- The Euler characteristic of $E(\mathscr{M}, \mathscr{F}, \overline{p})$ remains equal to that of the original map $\mathscr{M}$.

$$\chi_{E(\mathscr{M},\overline{p})} :\equiv (v + n + 1) - (e + 2 \cdot (n + 1)) + (f + n + 2) \equiv \chi_{\mathscr{M}}.$$