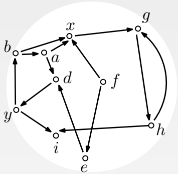
Bergen Univalent Foundations Meeting



Investigations on graph-theoretical constructions in HoTT

Jonathan Prieto-Cubides j.w.w. Håkon R. Gylterud

November 18, 2021

UNIVERSITETET I BERGEN

Setup

How to formulate planarity of graphs in HoTT?

Setup

How to formulate planarity of graphs in HoTT?

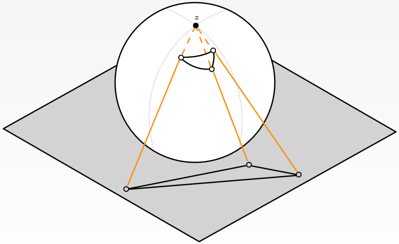
In graph theory, a planar graph is a graph that can be embedded in the **plane**.

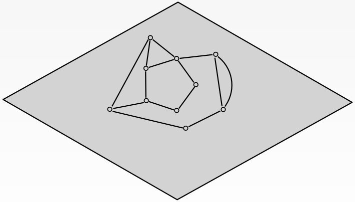
Setup

How to formulate planarity of graphs in HoTT?

In graph theory, a planar graph is a graph that can be embedded in the **plane**.

- Topological graph theory is about *the placement* of graphs on surfaces
- A graph embedding in a surface is a continuous one-to-one function from a topological representation of the graph into the surface





Overview

Graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

Graphs

- A (directed multigraph) *graph* is an object of type Graph:

$$\mathrm{Graph} :\equiv \sum_{(\mathrm{Node}:\mathcal{U})} \sum_{(\mathrm{Edge}:\mathrm{Node}\to\mathrm{Node}\to\mathcal{U})} \mathrm{isSet}(\mathrm{Node}) \times \prod_{(x,y:\mathrm{Node})} \mathrm{isSet}(\mathrm{Edge}(x,y)).$$

- $\mathrm{isFiniteGraph}(G) :\equiv \mathrm{isFinite}(\mathrm{Node}(G)) \times \prod_{x,y:\mathrm{Node}(G)} \mathrm{isFinite}(\mathrm{Edge}(G)(x,y))$

Graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

- $\mathsf{isFiniteGraph}(G) :\equiv \mathsf{isFinite}(\mathsf{Node}(G)) \times \prod_{x,y:\mathsf{Node}(G)} \mathsf{isFinite}(\mathsf{Edge}(G)(x,y))$
- The empty graph is $(\mathbb{0}, \lambda\{()\}, !, !)$

Graphs

- A (directed multigraph) *graph* is an object of type Graph:

$$\mathrm{Graph} :\equiv \sum_{(\mathrm{Node}:\mathcal{U})} \sum_{(\mathrm{Edge}:\mathrm{Node}\to\mathrm{Node}\to\mathcal{U})} \mathrm{isSet}(\mathrm{Node}) \times \prod_{(x,y:\mathrm{Node})} \mathrm{isSet}(\mathrm{Edge}(x,y)).$$

- $\mathrm{isFiniteGraph}(G) :\equiv \mathrm{isFinite}(\mathrm{Node}(G)) \times \prod_{x,y:\mathrm{Node}(G)} \mathrm{isFinite}(\mathrm{Edge}(G)(x,y))$
- The empty graph is $(\mathbb{0}, \lambda\{()\}, !, !)$
- The unit graph is $(\mathbb{1}, \lambda\, u\, v.\mathbb{0}, !, !)$

Graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

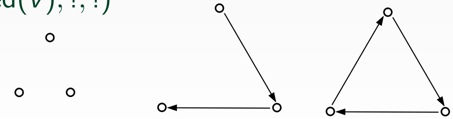
- $\mathsf{isFiniteGraph}(G) :\equiv \mathsf{isFinite}(\mathsf{Node}(G)) \times \prod_{x,y:\mathsf{Node}(G)} \mathsf{isFinite}(\mathsf{Edge}(G)(x,y))$
- The empty graph is $(\mathbb{0}, \lambda\{()\}, !, !)$
- The unit graph is $(\mathbb{1}, \lambda\, u\, v.\mathbb{0}, !, !)$
- The graph with n points and no edges $([n], \lambda\, u\, v.\mathbb{0}, !, !)$

Graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

- $\mathsf{isFiniteGraph}(G) :\equiv \mathsf{isFinite}(\mathsf{Node}(G)) \times \prod_{x,y:\mathsf{Node}(G)} \mathsf{isFinite}(\mathsf{Edge}(G)(x,y))$
- The empty graph is $(\mathbb{0}, \lambda\{()\}, !, !)$
- The unit graph is $(\mathbb{1}, \lambda\, u\, v.\mathbb{0}, !, !)$
- The graph with n points and no edges $([n], \lambda\, u\, v.\mathbb{0}, !, !)$
- The cycle graph of n points is $(\mathbb{n}, \lambda\, u\, v.u = \mathsf{pred}(v), !, !)$



The category of graphs

- A (directed multigraph) *graph* is an object of type Graph:

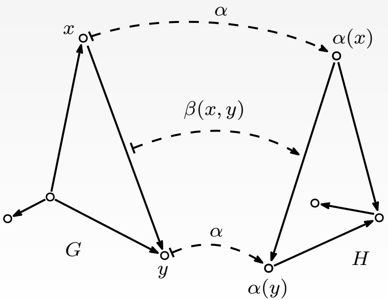
$$\mathsf{Graph} :\equiv \sum_{(\mathsf{Node}:\mathcal{U})} \sum_{(\mathsf{Edge}:\mathsf{Node}\to\mathsf{Node}\to\mathcal{U})} \mathsf{isSet}(\mathsf{Node}) \times \prod_{(x,y:\mathsf{Node})} \mathsf{isSet}(\mathsf{Edge}(x,y)).$$

The category of graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

- Structure-preserving functions are the graph homomorphisms (α, β)



The category of graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

- Structure-preserving functions are the graph homomorphisms (α, β)
- The canonical map $G = H \to G \cong H$ is an equivalence

The category of graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

- Structure-preserving functions are the graph homomorphisms (α, β)
- The canonical map $G = H \to G \cong H$ is an equivalence
- Graphs and graph homomorphisms form a category

The category of graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

- Structure-preserving functions are the graph homomorphisms (α, β)
- The canonical map $G = H \to G \cong H$ is an equivalence
- Graphs and graph homomorphisms form a category
- Isomorphic graphs hold the same properties

The category of graphs

- A (directed multigraph) *graph* is an object of type Graph:

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

- Structure-preserving functions are the graph homomorphisms (α, β)
- The canonical map $G = H \to G \cong H$ is an equivalence
- Graphs and graph homomorphisms form a category
- Isomorphic graphs hold the same properties
- The type Graph is a groupoid

How to formulate planarity for graphs in HoTT?

A graph is **planar** if and only if

- it has an *embedding into the sphere* or into the plane.
- it contains no subdivisions of K_5 or $K_{3,3}$ (Kuratowski 1930)
- it has an abstract dual (Whitney 1932)
- its cycle space has a sparse basis (Mac Lane 1937)
- the dimension of its incidence order is < 4 (Schnyder 1989)

Works on formal verification of results on planar graphs define planarity by:

- *hypermaps* as in the proof of The Four-colour theorem [4]
- inductive definitions (e.g. graph cycles [7], *near* triangulations [1], and directed face walks [2]

Ingredients:

$$\begin{tikzcd}
 & & \text{Planar map} \arrow[dll] \arrow[d] \arrow[dr] & \\
\text{Connected graph} \arrow[d] & & \text{Spherical map} \arrow[dl] \arrow[dd] & \text{Faces} \arrow[d] \\
\text{Walks} & \text{Map} \arrow[dl] \arrow[d] & & \text{Cyclic graph} \arrow[d] \\
\text{Cyclic sets} & \text{Stars} & \text{Walk homotopies} \arrow[uur] & \text{Cycle graph}
\end{tikzcd}$$

Ingredients: walks and connected graphs

$$
\begin{tikzcd}
 & & \text{Planar map} \arrow[dll] \arrow[d] \arrow[dr] & \\
\text{Connected graph} \arrow[d] & & \text{Spherical map} \arrow[dl] \arrow[dd] & \text{Faces} \arrow[d] \\
\text{Walks} & \text{Map} \arrow[dl] \arrow[d] & & \text{Cyclic graph} \arrow[d] \\
\text{Cyclic sets} & \text{Stars} & \text{Walk homotopies} \arrow[uur] & \text{Cycle graph}
\end{tikzcd}
$$

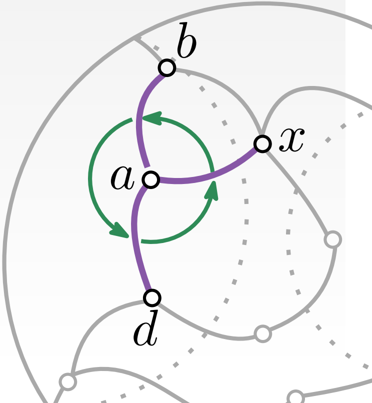
Basics:

- A walk in a graph G is build by one of the following constructors.
  - If $x : \mathsf{Node}(G)$ then $\langle x \rangle : \mathsf{W}(x, x)$
  - If $x, y, z : \mathsf{Node}(G), e : \mathsf{Edge}(G, x, y), w : \mathsf{W}(y, z)$, then $e \odot w : \mathsf{W}(x, z)$
- $\mathsf{isGraphConnected}(G) :\equiv \prod_{(x,y:\mathsf{Node}(G))} \|\mathsf{W}(x, y)\|$

Ingredients: Graph embeddings/combinatorial maps

Theorem (pp. 113 in [5])

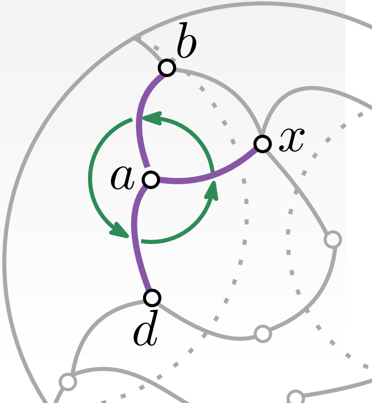
- *Every locally oriented embedding from a graph G to a surface S defines a rotation system for G.*
- *Conversely, every rotation system on a graph G defines, up to equivalence of embeddings, a unique locally oriented graph embedding from G to S.*



Ingredients: Graph embeddings/combinatorial maps

The essential information of a graph embedding is stored in the *cyclic* order of the edges incident at each node.

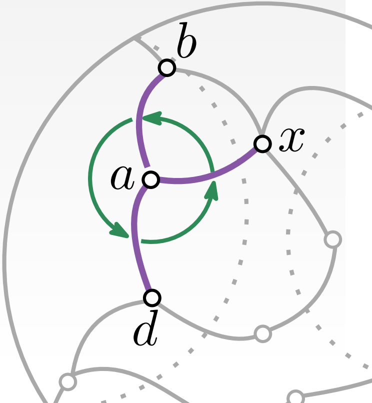
- $\mathsf{Map}(G) :\equiv \prod_{(x:\mathsf{Node}(G))} \mathsf{Cyclic}(\mathsf{Star}_{U(G)}(x))$
- $U(G) :\equiv (\mathsf{Node}(G), \mathsf{Edge}(G)(x,y) + \mathsf{Edge}(G)(y,x), !, !)$



Ingredients: Graph embeddings/combinatorial maps

The essential information of a graph embedding is stored in the *cyclic* order of the edges incident at each node.

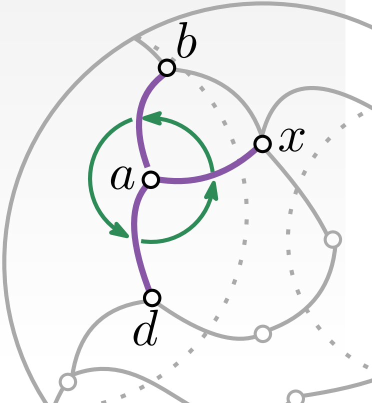
- $\mathsf{Map}(G) :\equiv \prod_{(x:\mathsf{Node}(G))} \mathsf{Cyclic}(\mathsf{Star}_{U(G)}(x))$
- $U(G) :\equiv (\mathsf{Node}(G), \mathsf{Edge}(G)(x,y) + \mathsf{Edge}(G)(y,x), !, !)$



Ingredients: Graph embeddings/combinatorial maps

The essential information of a graph embedding is stored in the *cyclic* order of the edges incident at each node.

- $\mathsf{Map}(G) :\equiv \prod_{(x:\mathsf{Node}(G))} \mathsf{Cyclic}(\mathsf{Star}_{U(G)}(x))$
- $U(G) :\equiv (\mathsf{Node}(G), \mathsf{Edge}(G)(x,y) + \mathsf{Edge}(G)(y,x), !, !)$
- $\mathsf{Star}_G(x) :\equiv \sum_{(y:\mathsf{Node}(G))} \mathsf{Edge}(H)(x,y)$

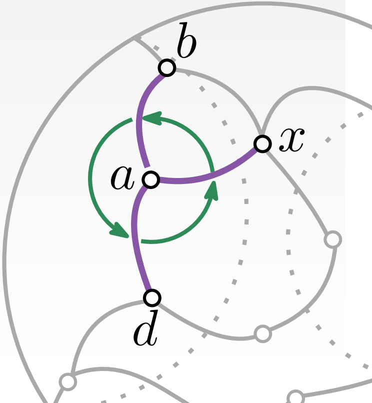


Ingredients: Graph embeddings/combinatorial maps

The essential information of a graph embedding is stored in the *cyclic* order of the edges incident at each node.

- $\mathsf{Map}(G) :\equiv \prod_{(x:\mathsf{Node}(G))} \mathsf{Cyclic}(\mathsf{Star}_{U(G)}(x))$
- $U(G) :\equiv (\mathsf{Node}(G), \mathsf{Edge}(G)(x,y) + \mathsf{Edge}(G)(y,x), !, !)$
- $\mathsf{Star}_G(x) :\equiv \sum_{(y:\mathsf{Node}(G))} \mathsf{Edge}(H)(x,y)$
- $\mathsf{Cyclic}(A) :\equiv \sum_{(\varphi:A\to A)} \sum_{(n:\mathbb{N})} \| \sum_{(e:A\simeq[n])} e \circ \varphi = \mathsf{pred} \circ e \|$

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d] & A \arrow[d] \\ {[n]} \arrow[r, "\mathsf{pred}"'] & {[n]} \end{tikzcd}$$



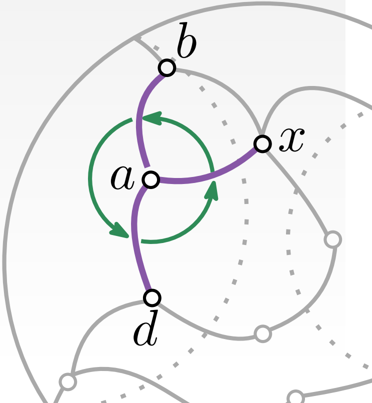
Ingredients: Graph embeddings/combinatorial maps

The essential information of a graph embedding is stored in the *cyclic* order of the edges incident at each node.

- $\mathsf{Map}(G) :\equiv \prod_{(x:\mathsf{Node}(G))} \mathsf{Cyclic}(\mathsf{Star}_{U(G)}(x))$
- $U(G) :\equiv (\mathsf{Node}(G), \mathsf{Edge}(G)(x,y) + \mathsf{Edge}(G)(y,x), !, !)$
- $\mathsf{Star}_G(x) :\equiv \sum_{(y:\mathsf{Node}(G))} \mathsf{Edge}(H)(x,y)$
- $\mathsf{Cyclic}(A) :\equiv \sum_{(\varphi:A\to A)} \sum_{(n:\mathbb{N})} \| \sum_{(e:A\simeq[n])} e \circ \varphi = \mathsf{pred} \circ e \|$

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d] & A \arrow[d] \\ {[n]} \arrow[r, "\mathsf{pred}"'] & {[n]} \end{tikzcd}$$

$$\begin{tikzcd} n-1 & 0 \arrow[l, mapsto, "\mathsf{pred}"'] & 1 \arrow[l, mapsto, "\mathsf{pred}"'] \\ i \arrow[rr, mapsto, "\mathsf{pred}"'] & & i-1 \arrow[u, dotted, "\mathsf{pred}^{i-2}"'] \arrow[from=1-1, to=2-1, dotted, "\mathsf{pred}^{n-i-1}"] \end{tikzcd}$$



How many maps does a graph have?

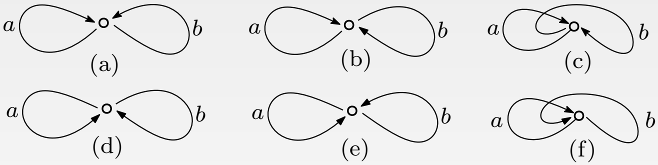
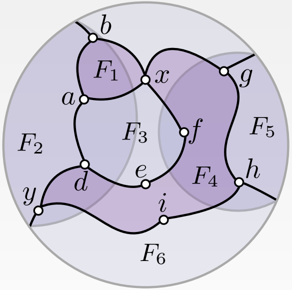


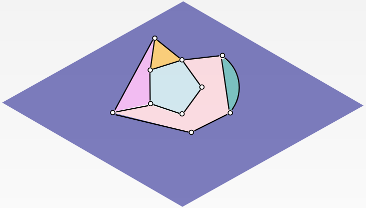
Figure: The six possible maps of the bouquet B_2.

- The surface arising from the maps M_a and M_b is the two-dimensional plane.
- For the map M_c, the surface is the topological torus.

$$M_a :\equiv (0 \mapsto (a^{\rightarrow} a^{\leftarrow} b^{\leftarrow} b^{\rightarrow})).$$
$$M_b :\equiv (0 \mapsto (a^{\rightarrow} a^{\leftarrow} b^{\rightarrow} b^{\leftarrow})).$$
$$M_c :\equiv (0 \mapsto (a^{\rightarrow} b^{\rightarrow} a^{\leftarrow} b^{\leftarrow})).$$

What maps embed a graph in the plane/sphere?





Face of a map

A face of a map $\mathcal{M}$ is a circuit in the graph with some special condition:
A face consists of the following data:

Face of a map

A face of a map $\mathcal{M}$ is a circuit in the graph with some special condition:
A face consists of the following data:

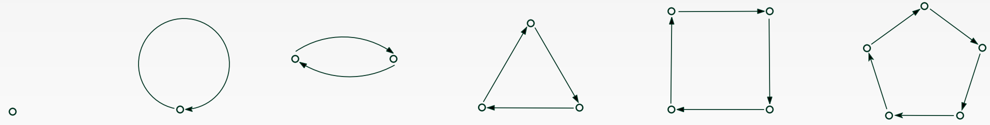
- A cyclic graph A,

Face of a map

A face of a map $\mathcal{M}$ is a circuit in the graph with some special condition:
A face consists of the following data:

- A cyclic graph A,

$$\mathsf{CyclicGraph}(G) :\equiv \sum_{(\varphi:\mathsf{Hom}(G,G))} \sum_{(n:\mathbb{N})} \underbrace{\|(G,\varphi) = (C_n, \mathsf{rot})\|}_{\mathsf{iscyclic(G,\varphi,n)}}.$$



C_0 C_1 C_2 C_3 C_4 C_5

Face of a map

A face of a map $\mathcal{M}$ is a circuit in the graph with some special condition:
A face consists of the following data:

- A cyclic graph A,
- an *edge-injective* homomorphism $h \equiv (\alpha, \beta) : \mathrm{Hom}(A, U(G))$, such that ...

Face of a map

A face of a map $\mathcal{M}$ is a circuit in the graph with some special condition:
A face consists of the following data:

- A cyclic graph A,
- an *edge-injective* homomorphism $h \equiv (\alpha, \beta) : \mathrm{Hom}(A, U(G))$, such that ...

A graph homomorphism h, $(\alpha, \beta) : \mathrm{Hom}(G, H)$, is *edge-injective* if for any $e_1, e_2 : \mathsf{E}_G(x, y)$, $x, y : \mathsf{N}_G$, when $\beta(x, y, e_1) =_{\mathsf{E}_H(\alpha(x), \alpha(y))} \beta(x, y, e_2)$ then $e_1 =_{\mathsf{E}_G(x,y)} e_2$.

Face of a map

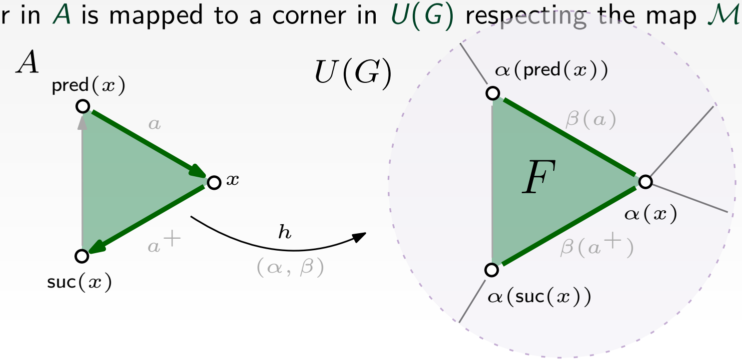
A face of a map $\mathcal{M}$ is a circuit in the graph with some special condition:
A face consists of the following data:

- A cyclic graph A,
- an *edge-injective* homomorphism $h \equiv (\alpha, \beta) : \mathsf{Hom}(A, U(G))$, such that ...
- for any node x in A, if $\|\mathsf{Star}(U(G))(h(x))\|$ then $\|\mathsf{Star}(A)(x)\|$, and

Face of a map

A face of a map $\mathcal{M}$ is a circuit in the graph with some special condition:
A face consists of the following data:

- A cyclic graph A,
- an *edge-injective* homomorphism $h \equiv (\alpha, \beta) : \mathsf{Hom}(A, U(G))$, such that ...
- for any node x in A, if $\|\mathsf{Star}(U(G))(h(x))\|$ then $\|\mathsf{Star}(A)(x)\|$, and
- any corner in A is mapped to a corner in $U(G)$ respecting the map $\mathcal{M}$.

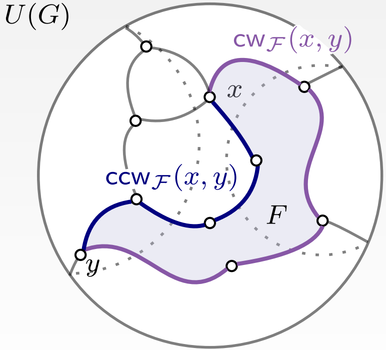


What else do we know when embedding a graph in the sphere?

- The implicit surface given by the embedding has to be *simply connected*, i.e. any loop can be contracted/deformed to one point

What else do we know when embedding a graph in the sphere?

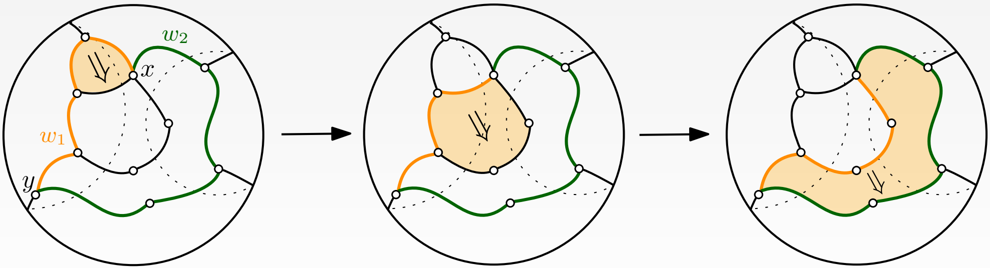
- The implicit surface given by the embedding has to be *simply connected*, i.e. any loop can be contracted/deformed to one point



Maps into a sphere: Spherical maps

A map $\mathcal{M}$ of a graph G is *spherical*, of type $\mathsf{Spherical}(\mathcal{M})$, if any pair of walks sharing the same endpoints are merely walk-homotopic.

$$\mathsf{Spherical}(\mathcal{M}) :\equiv \prod_{(x,y:\mathsf{Node}_G)} \prod_{(w_1,w_2:\mathsf{Walk}_{U(G)}(x,y))} \| \, w_1 \sim_{\mathcal{M}} w_2 \, \| \, .$$



Homotopy for walks is an equiv. relation that collapses faces

Supposing one has the following,

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

(ii) a walk w_1 of type $\mathsf{W}_{U(G)}(x, f(a))$ for $x : \mathsf{N}_G$ with one node $a : \mathsf{N}_A$, and

(iii) a walk w_2 of type $\mathsf{W}_{U(G)}(f(b), y)$ for $b : \mathsf{N}_A$ and $y : \mathsf{N}_G$.

Homotopy for walks is an equiv. relation that collapses faces

Supposing one has the following,

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

(ii) a walk w_1 of type $\mathsf{W}_{U(G)}(x, f(a))$ for $x : \mathsf{N}_G$ with one node $a : \mathsf{N}_A$, and

(iii) a walk w_2 of type $\mathsf{W}_{U(G)}(f(b), y)$ for $b : \mathsf{N}_A$ and $y : \mathsf{N}_G$.

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

Homotopy for walks is an equiv. relation that collapses faces

Supposing one has the following,

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

(ii) a walk w_1 of type $\mathsf{W}_{U(G)}(x, f(a))$ for $x : \mathsf{N}_G$ with one node $a : \mathsf{N}_A$, and

(iii) a walk w_2 of type $\mathsf{W}_{U(G)}(f(b), y)$ for $b : \mathsf{N}_A$ and $y : \mathsf{N}_G$.

$$\begin{tikzcd} \bullet_x \arrow[r, "w_1"] & \bullet_{f(a)} \arrow[r, bend left, "\mathsf{cw}_{\mathcal{F}}(a{,}b)"{name=U}] \arrow[r, bend right, "\mathsf{ccw}_{\mathcal{F}}(a{,}b)"'{name=D}] & \bullet_{f(b)} \arrow[r, "w_2"] & \bullet_y \arrow[from=U, to=D, Rightarrow, shorten=2pt] \end{tikzcd}$$

Homotopy for walks is an equiv. relation that collapses faces

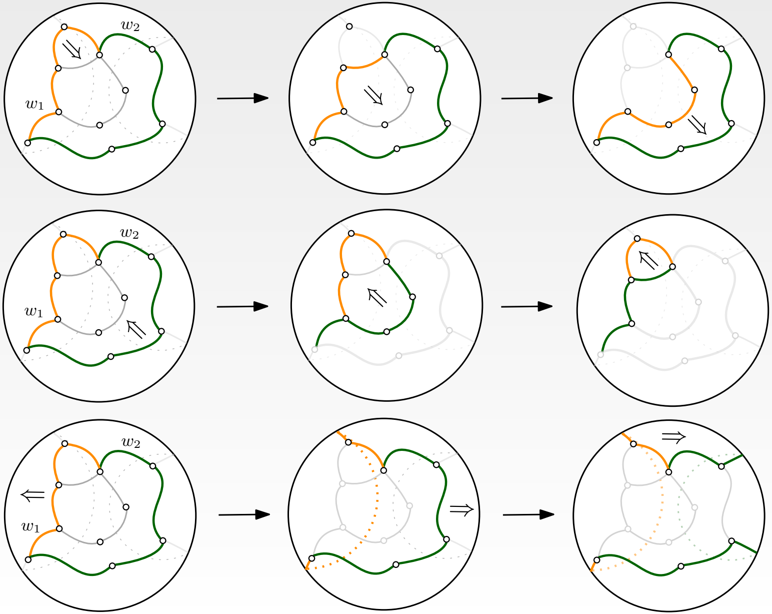
Supposing one has the following,

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

(ii) a walk w_1 of type $\mathsf{W}_{U(G)}(x, f(a))$ for $x : \mathsf{N}_G$ with one node $a : \mathsf{N}_A$, and

(iii) a walk w_2 of type $\mathsf{W}_{U(G)}(f(b), y)$ for $b : \mathsf{N}_A$ and $y : \mathsf{N}_G$.

- Whiskering lemmas are available!

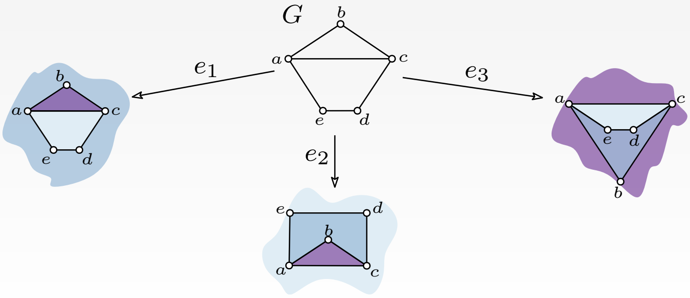


w_2
w_1

Planar map

- A planar map $\mathcal{M}$ of a connected and locally finite graph G is of type

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



Lemmas

- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- If G is a (finite) graph, then the following types are (finite) sets.

Lemmas

- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- 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

Lemmas

- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- 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

Lemmas

- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- 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

Lemmas

- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- 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

Lemmas

- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- 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.

Lemmas

- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- 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.
- The collection of planar maps of G is a (finite) set.

Lemmas

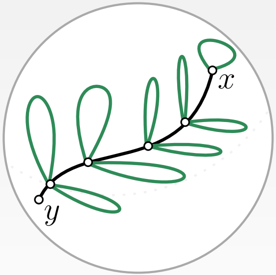
- (Finite) sets are closed under (co) products, type equivalences, Σ-types, Π-types and prop. truncation.
- If G is a (finite) graph, then the following types are (finite) sets.
  - The star at x for any node x is a (finite) set
  - The maps of G is a (finite) set
  - The spherical maps of G is a (finite) set
  - The faces of any map for G is a (finite) set, and
  - The walks without inner loops in G is a (finite) set.
- The collection of planar maps of G is a (finite) set.
- One can construct for every graph C_n a planar map, which can help us to construct more planar graphs!

A refinement for spherical maps

How to construct terms of type $\mathsf{Planar}(G)$ for some given graph G?

A refinement for spherical maps

How to construct terms of type $\mathsf{Planar}(G)$ for some given graph G?

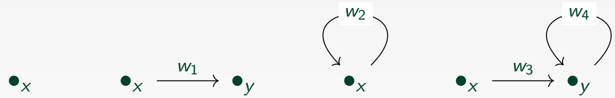


A refinement for spherical maps

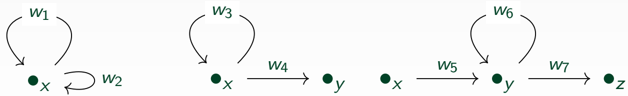
How to construct terms of type $\mathsf{Planar}(G)$ for some given graph G?

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

- 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 $x, y, z : \mathrm{Node}_G$ and $w : \mathrm{W}_G(x, z)$.

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathrm{head}(e)) + (z \in w)$.

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathrm{head}(e)) + (z \in w)$.
- $\mathrm{isQuasi}(w) :\equiv \prod_{(z:\mathrm{Node}_G)} \mathrm{isProp}(z \in w)$

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathrm{head}(e)) + (z \in w)$.
- $\mathrm{isQuasi}(w) :\equiv \prod_{(z:\mathrm{Node}_G)} \mathrm{isProp}(z \in w)$
- One can define a relation on walks $(\rightsquigarrow)$ to remove loops

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathsf{head}(e)) + (z \in w)$.
- $\mathsf{isQuasi}(w) :\equiv \prod_{(z:\mathsf{Node}_G)} \mathsf{isProp}(z \in w)$
- One can define a relation on walks $(\rightsquigarrow)$ to remove loops
- $\mathsf{Normal}(p) :\equiv \mathsf{isQuasi}(p) \times \neg \sum_{(q:\mathsf{W}_G(x,y))} (p \rightsquigarrow q)$.

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathsf{head}(e)) + (z \in w)$.
- $\mathsf{isQuasi}(w) :\equiv \prod_{(z:\mathsf{Node}_G)} \mathsf{isProp}(z \in w)$
- One can define a relation on walks $(\rightsquigarrow)$ to remove loops
- $\mathsf{Normal}(p) :\equiv \mathsf{isQuasi}(p) \times \neg \sum_{(q:\mathsf{W}_G(x,y))}(p \rightsquigarrow q)$.
- Thm. $(\in)$, $\mathsf{isQuasi}$, Normal are all decidable propositions.

Quasi-simple walks

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

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathsf{head}(e)) + (z \in w)$.
- $\mathsf{isQuasi}(w) :\equiv \prod_{(z:\mathsf{Node}_G)} \mathsf{isProp}(z \in w)$
- One can define a relation on walks $(\rightsquigarrow)$ to remove loops
- $\mathsf{Normal}(p) :\equiv \mathsf{isQuasi}(p) \times \neg \sum_{(q:\mathsf{W}_G(x,y))}(p \rightsquigarrow q)$.
- Thm. $(\in)$, isQuasi, Normal are all decidable propositions.
- Thm. The type $\sum_{(v:\mathsf{W}(x,z))}(w \rightsquigarrow^* v) \times \mathsf{Normal}(v)$ is inhabited for any walk w.

Quasi-simple walks

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

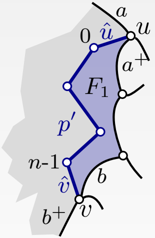
- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathrm{head}(e)) + (z \in w)$.
- $\mathrm{isQuasi}(w) :\equiv \prod_{(z:\mathrm{Node}_G)} \mathrm{isProp}(z \in w)$
- One can define a relation on walks $(\rightsquigarrow)$ to remove loops
- $\mathrm{Normal}(p) :\equiv \mathrm{isQuasi}(p) \times \neg \sum_{(q:\mathrm{W}_G(x,y))}(p \rightsquigarrow q)$.
- Thm. $(\in)$, isQuasi, Normal are all decidable propositions.
- Thm. The type $\sum_{(v:\mathrm{W}(x,z))}(w \rightsquigarrow^* v) \times \mathrm{Normal}(v)$ is inhabited for any walk w.
- Thm. Given a spherical map $\mathcal{M}$, the type $\sum_{(v:\mathrm{W}(x,z))}(w \rightsquigarrow^* v) \times \mathrm{Normal}(v) \times \|w \sim_{\mathcal{M}} v\|$ is inhabited for any walk w.

Quasi-simple walks

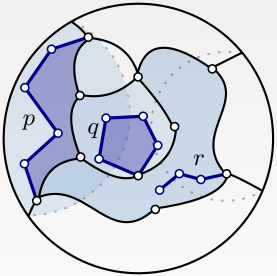
Assuming the node set is discrete, let $x, y, z : \mathsf{Node}_G$ and $w : \mathsf{W}_G(x, z)$.

- The membership relation $(\in)$ on a walk w for a node y is defined as the node y belonging to w whenever the type $(y \in w)$ is inhabited.
  - $z \in \langle y \rangle :\equiv \mathbf{0}$.
  - $z \in (e \odot w) :\equiv (z = \mathsf{head}(e)) + (z \in w)$.
- $\mathsf{isQuasi}(w) :\equiv \prod_{(z:\mathsf{Node}_G)} \mathsf{isProp}(z \in w)$
- One can define a relation on walks $(\rightsquigarrow)$ to remove loops
- $\mathsf{Normal}(p) :\equiv \mathsf{isQuasi}(p) \times \neg \sum_{(q:\mathsf{W}_G(x,y))}(p \rightsquigarrow q)$.
- Thm. $(\in)$, isQuasi, Normal are all decidable propositions.
- Thm. The type $\sum_{(v:\mathsf{W}(x,z))}(w \rightsquigarrow^* v) \times \mathsf{Normal}(v)$ is inhabited for any walk w.
- Thm. Given a spherical map $\mathcal{M}$, the type $\sum_{(v:\mathsf{W}(x,z))}(w \rightsquigarrow^* v) \times \mathsf{Normal}(v) \times \|w \sim_{\mathcal{M}} v\|$ is inhabited for any walk w.
- The two spherical definitions are locally equivalent!

Planar extensions*: planar synthesis



(a) Addition of p to G.



(b) The embedded graph $U(G \bullet p \bullet q \bullet r)$.

Planar extensions*: planar synthesis

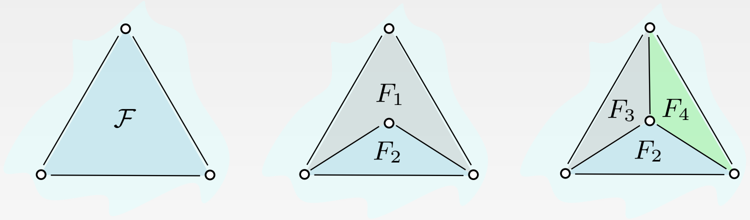


Figure: The figure is a planar synthesis of the construction of a planar map for K_4 from a planar map of C_3. One first divides the face $\mathcal{F}$ into F_1 and F_2. Then one splits F_1 into F_3 and F_4.

- In a synthesis from a connected graph, every graph in the sequence is connected.
- In a planar synthesis, every graph in the sequence is planar.

Planar extensions*: construct any biconnected planar graph

- $\mathsf{Biconnected}(G) :\equiv \prod_{x:\mathsf{N}_G} \mathsf{Connected}(G - x)$.
- If G is a cyclic graph, then $U(G)$ is 2-connected.
- The 2-connectedness of a graph is not preserved by simple path additions.
- Suppose G is a 2-connected graph, then the following claims hold.
  1. Every node in G has degree of minimum two.
  2. There exists a cyclic graph H and an injective morphism from $U(H)$ to G.
  3. The graphs $G \bullet \overline{p}$, $U(G \bullet p)$, and $U(G) \bullet \overline{p}$ are all 2-connected.
- In a non-simple Whitney synthesis of G of length n from a 2-connected cyclic graph H, every graph G_i in the sequence is a 2-connected planar graph.

Realisations of graphs

Let G be a directed multigraph. We denote by $\mathbb{G}^n(G)$ the topological realisation of G that considers the first n layers, i.e. 0-, 1-, $\cdots$, and n-cells.

Realisations of graphs

Let G be a directed multigraph. We denote by $\mathbb{G}^n(G)$ the topological realisation of G that considers the first n layers, i.e. 0-, 1-, $\cdots$, and n-cells.

- **One layer**:

$$
\begin{aligned}
&\text{data } \mathbb{G}^1(G : \text{Graph}) : \mathcal{U}\\
&\qquad \mathbb{n} : \text{Node}(G) \to \mathbb{G}^1(G)\\
&\qquad \mathbb{e} : \Pi_{(a\, b\, :\, \text{Node}(G))}\,.\,\text{Edge}(G, a, b) \to \mathbb{n}(a) = \mathbb{n}(b).
\end{aligned}
$$

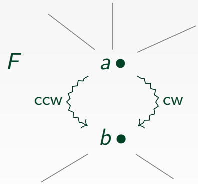
Realisations of graphs

Let G be a directed multigraph. We denote by $\mathbb{G}^n(G)$ the topological realisation of G that considers the first n layers, i.e. 0-, 1-, $\cdots$, and n-cells.

- **One layer**:

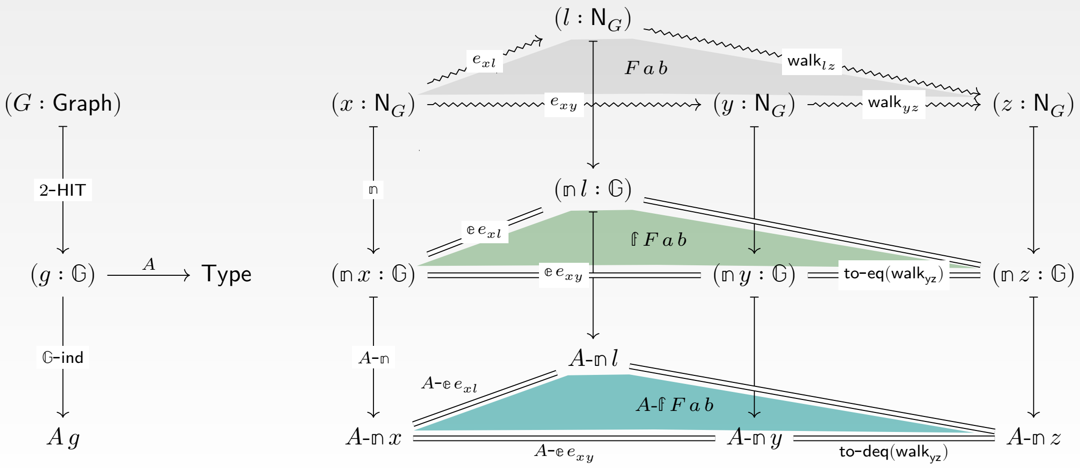
$$
\begin{aligned}
&\text{data } \mathbb{G}^1(G : \text{Graph}) : \mathcal{U}\\
&\qquad \mathbb{n} : \text{Node}(G) \to \mathbb{G}^1(G)\\
&\qquad \mathbb{e} : \Pi_{(a\, b\, :\, \text{Node}(G))}\, .\, \text{Edge}(G, a, b) \to \mathbb{n}(a) = \mathbb{n}(b).
\end{aligned}
$$

- **Two layer**: Given a *combinatorial map* $\mathcal{M}$ for G:



$$
\begin{aligned}
&\text{data } \mathbb{G}^2(G : \text{Graph}) : \mathcal{U}\\
&\qquad \mathbb{n} : \text{Node}(G) \to \mathbb{G}^2(G)\\
&\qquad \mathbb{e} : \Pi_{(a\, b\, :\, \text{Node}(G))}\, .\, \text{Edge}(G, a, b) \to \mathbb{n}(a) = \mathbb{n}(b)\\
&\qquad \mathbb{f} : \Pi_{(\mathcal{F} : \text{Face}(G, \mathcal{M}))}\, .\, \Pi_{(a\, b\, : \text{Node}(\mathcal{F})}\, .\, \mathbb{w}(\text{cw}(\mathcal{F}, a, b)) = \mathbb{w}(\text{ccw}(\mathcal{F}, a, b)).
\end{aligned}
$$

The elimination principle for the two-level top. realisation



Work in progress

Goal

Let G be a nonempty finite planar graph with n nodes. Then $\mathbb{G}^2(G) \simeq \mathbb{S}^2$.

Work in progress

Goal

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 $\mathcal{M}$.
  - Lem. 2. Face contraction preserves planarity
  - Lem. 3. H is obtained by *contracting a face* F of $\mathcal{M}$, then $\mathbb{G}^2(G) \simeq \mathbb{G}^2(H)$.

Goal

Let G be a nonempty finite planar graph with n nodes. Then $\mathbb{G}^2(G) \simeq \mathbb{S}^2$.

Proof.

- Case $n = 1$. Apply Lemma 1a. The graph is $\bullet$.
- Case $n > 1$. Let $\mathcal{M}$ be a planar map for G. Because G is a nonempty finite graph, then let m be the number of faces of $\mathcal{M}$. 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 $\mathcal{M}$. By contracting the face F, one obtains a graph G' and a map $\mathcal{M}'$ such that $(G, \mathcal{M}) \rightsquigarrow_F (G', \mathcal{M}')$. 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 $\mathcal{M}'$ 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', $\mathcal{M}'$, an equivalence $\mathbb{G}^2(G') \simeq \mathbb{S}^2$ is obtained. Finally, the conclusion follows from the chain of equivalences:

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

$\square$

Bonus slides

Notation

definitions	$:\equiv$
jugdemental equalities	$\equiv$
identity type	$=$
type equivalences	$\simeq$
univalent universe	$\mathcal{U}$
"*a* is of type *A*"	(a : A)
Σ-types	$\Sigma_{x:A}B(x)$
Π-types	$\Pi_{x:A}B(x)$
natural numbers	$\mathbb{N}$
empty type and unit type	$\mathbf{0}$ and $\mathbf{1}$
the type with n points	$[n]$ where $n : \mathbb{N}$
propositional truncation of A	$\|A\|$

References I

- G. Bauer and T. Nipkow.
  The 5 colour theorem in isabelle/isar.
  In V. A. Carreño, C. A. Muñoz, and S. Tahar, editors, *Theorem Proving in Higher Order Logics, 15th International Conference, TPHOLs 2002, Hampton, VA, USA, August 20-23, 2002, Proceedings*, pages 67–82, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- G. J. Bauer.
  Formalizing Plane Graph Theory: Towards a Formalized Proof of the Kepler Conjecture.
  PhD thesis, Technische Universität München, Germany, 2005.

References II

- P. Giblin.
  Graphs, Surfaces and Homology.
  Springer, 2010.
- G. Gonthier.
  Formal proof–the four-color theorem.
  Notices of the AMS, 55(11):1382–1393, 2008.
- J. L. Gross and T. W. Tucker.
  Topology Graph Theory.
  Dover, Ny, Usa, 1987.
- T. Univalent Foundations Program.
  Homotopy Type Theory: Univalent Foundations of Mathematics.
  https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

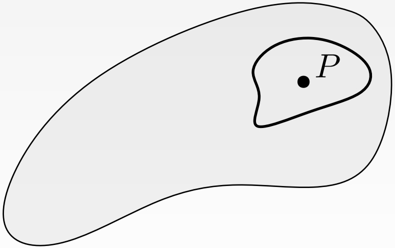
References III

- M. Yamamoto, S. Nishizaki, M. Hagiya, and Y. Toda.
  Formalization of planar graphs.
  In E. T. Schubert, P. J. Windley, and J. Alves-Foss, editors, *Higher Order Logic Theorem Proving and Its Applications, 8th International Workshop, Aspen Grove, UT, USA, September 11-14, 1995, Proceedings*, volume 971 of *Lecture Notes in Computer Science*, pages 369–384, Ut, Usa, 1995. Springer.

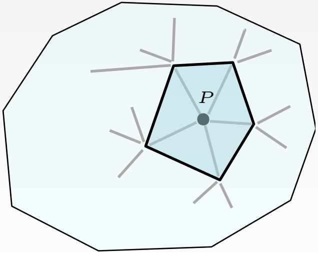
UNIVERSITY OF BERGEN

Combinatorial methods for graph embeddings

- Graph embeddings in surfaces can be analyzed by combinatorial methods. (See § 3.1.4 in [5]) e.g., rotation systems
- The generalization of the Schoenflies theorem states that for any embedding G to S, the graph G is contained in the 1-skeleton of a triangulation of the surface S



$\rightarrow$



Structure on a graph

Definition (Graph class)

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

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

Examples:

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

Definition (Homotopy levels)

Let $n : \mathbb{N}$, $n \geq 2$. A type A is called *n-type* when is-level(n, A).

$$\text{is-level}(-2, A) :\equiv \textstyle\sum_{(c:A)} \prod_{(x:A)} (c = x) \quad \text{and} \quad \text{is-level}(n+1, A) :\equiv \prod_{(x,y:A)} \text{is-level}(n, A).$$

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

Definition (Propositional truncation)

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

$$\begin{tikzcd} A \arrow[r, "|\text{-}|"] \arrow[dr, "f"'] & \|A\| \arrow[d, dashed, "\text{trunc-elim}"] \\ & P \end{tikzcd}$$

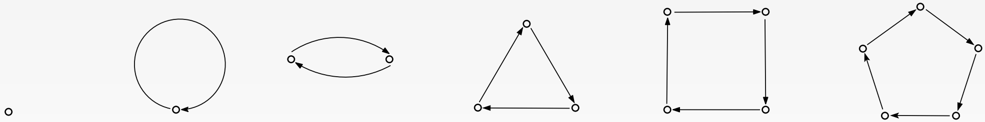
- $P \vee Q :\equiv \|P + Q\|, \quad P \wedge Q :\equiv \|P \times Q\|, \quad \textit{and} \quad \exists(x : A)P(x) :\equiv \|\Sigma_{x:A} Px\|.$

Examples of families of graphs ($\mathbb{N} \to \text{Graph}$)

- The family of *cycle* graphs:

Definition (n-cycle graph)

Given $n : \mathbb{N}$, an n-cycle graph denoted by C_n is defined by $C_n :\equiv ([n], \lambda\, u\, v. u = \mathsf{pred}(v))$ for $n \geq 1$ and C_0 as the one-point graph.



C_0 C_1 C_2 C_3 C_4 C_5

Lemmas

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

- 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 $[\![n]\!] \simeq \Sigma_{(y:\mathrm{N}_G)}(y \in w)$.
- 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 $x, y : \mathrm{N}_G$ and $n : \mathbb{N}$, the type qswalk collects all quasi-simple walks of a fixed length n.

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

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

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

Lemmas

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

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

- 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.

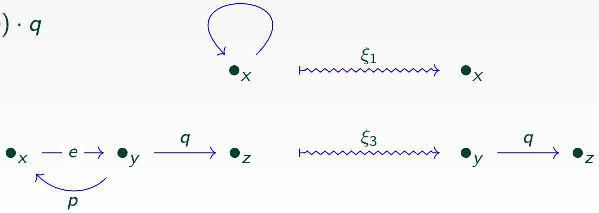
$$\sum_{(w:\mathsf{W}_G(x,y))} \mathsf{isQuasi}(w) \simeq \sum_{(m:[\![n+1]\!])} \mathsf{qswalk}(m, x, y).$$

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

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

Loop-reduction relation on walks

$$
\begin{array}{rll}
\text{data } (\rightsquigarrow) & : & \Pi\,\{x, y : \mathsf{N}_G\}.\mathsf{W}_G(x,y) \to \mathsf{W}_G(x,y) \to \mathcal{U} \\
\xi_1 & : & \Pi\,\{x\,y\}\,.\,(p : \mathsf{W}_G(x,y))\,(q : \mathsf{W}_G(x,y)) \\
 & & \quad \to \mathsf{NonTrivialLoop}(p) \to \mathsf{Trivial}(q) \\
 & & \quad \to p \rightsquigarrow q \\
\xi_2 & : & \Pi\,\{x\,y\,z\}\,.\,(e : \mathsf{E}_G(x,y))\,(p, q : \mathsf{W}_G(y,z)) \\
 & & \quad \to \neg\,\mathsf{Loop}(e \odot p) \to x \neq y \\
 & & \quad \to (p \rightsquigarrow q) \to (e \odot p) \rightsquigarrow (e \odot q) \\
\xi_3 & : & \Pi\,\{x\,y\,z\}\,.\,(e : \mathsf{E}_G(x,y))\,(p : \mathsf{W}_G(y,x))\,(q : \mathsf{W}_G(x,z)) \\
 & & \quad \to \neg\,\mathsf{Loop}((e \odot p) \cdot q) \to \mathsf{Loop}(e \odot p) \\
 & & \quad \to \mathsf{NonTrivial}(q) \\
 & & \quad \to (w : \mathsf{W}_G(x,z)) \to w = (e \odot p) \cdot q \\
 & & \quad \to w \rightsquigarrow q
\end{array}
$$



- 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.

Theorem (Normalisation)

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

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

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