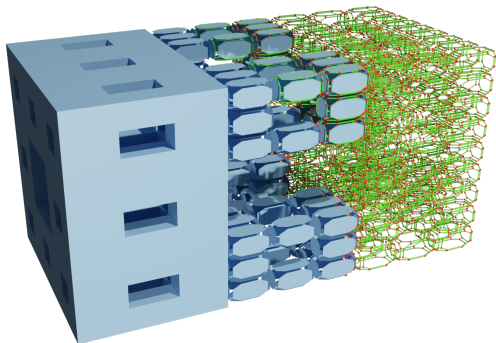


Graph rewriting for topology-based geometric modeling

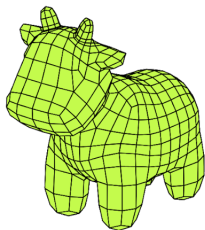


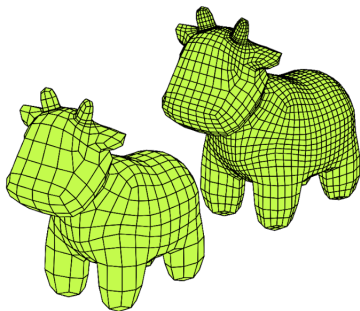
Romain Pascual

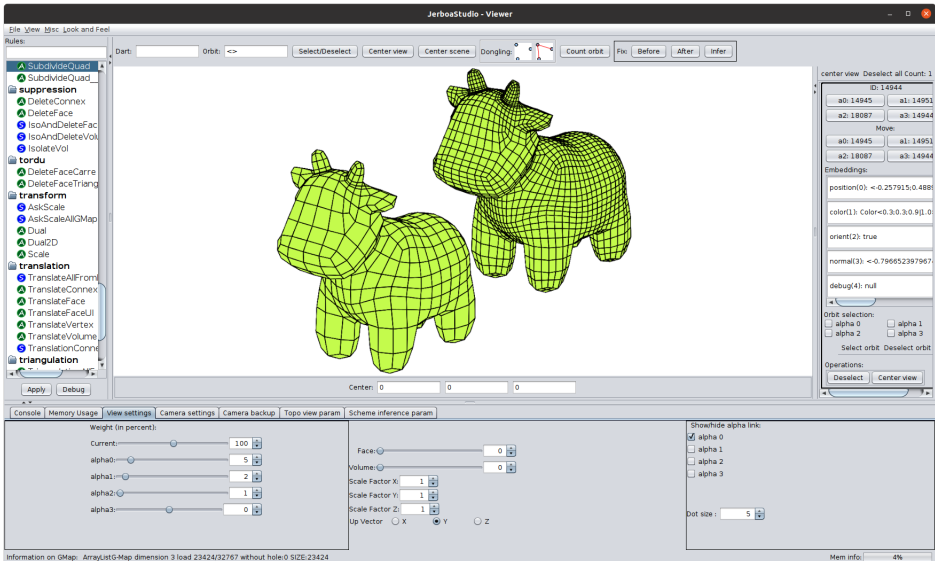
December 7th, 2021



université
PARIS-SACLAY









JerboaStudio - Viewer

File View Misc Look and Feel

Rules:

- SubdivideQuad
- suppression
 - DeleteConnex
 - DeleteFace
 - IsoAndDeleteFac
 - IsoAndDeleteVol
 - IsolateVol
- tordu
 - DeleteFaceCarre
 - DeleteFaceTriang
- transform
 - AskScale
 - AskScaleAllGMap
 - Dual
 - Dual2D
 - Scale
- translation
 - TranslateAllFrom
 - TranslateConnex
 - TranslateFace
 - TranslateFaceUl
 - TranslateVertex
 - TranslateVolume
 - TranslationConne
- triangulation

Apply Debug

Console Memory Usage View settings Camera settings Camera backup Te

Weight (in percent):

Current: 100

alpha0: 5

alpha1: 2

alpha2: 1

alpha3: 0

Information on GMap: ArrayListG-Map dimension 3 load 23424/32767 without hole:0 SIZE:23424

JerboaStudio - Editor

File Edit Window Inference

SubdivideQuad

User name: roman

Modeler's Name: JerboaModelerStudio

Modeler's package: fr.up.xim.ig.jerboa.studio

Dimension: 3

Adv. param. Check all

Embeddings(5):

- position: <a1, a2, a3> ->
- color: <a0, a1> -> fr.up.xi
- orient: <-> -> java.lang.B
- normal: <a0, a1> -> fr.up
- debug: <a1, a2, a3> -> j

Rules: 109

- color
- creat
- duplication
- extrusion
- hexadominant
- infer
- infertool
- modélisation
- normal
- rotation
- sew
- subdivision

Filter:

SubdivideQuad

Unlock

File Views

Scale x1 Composition CompositionSelf Shape: CIRCLE

CRITIQUE (0)

WARNING (0)

INFO (0)

Details Topo. Param. Ebd. Param. Errors

Informations:

Scale Factor X: 1

Scale Factor Y: 1

Scale Factor Z: 1

Up Vector: ☐ X ☒ Y ☐ Z

center view Deselect all Count: 1

ID: 14944

a0: 14945 a1: 14951

a2: 18087 a3: 14944

Move:

a0: 14945 a1: 14951

a2: 18087 a3: 14944

Embeddings:

position(0): <0.257915;0.488

color(1): Color<0.3;0.3;0.9>[0

orient(2): true

normal(3): <-0.796652397967

debug(4): null

Orbit selection:

☐ alpha 0 ☐ alpha 1

☐ alpha 2 ☐ alpha 3

Select orbit Deselect orbit

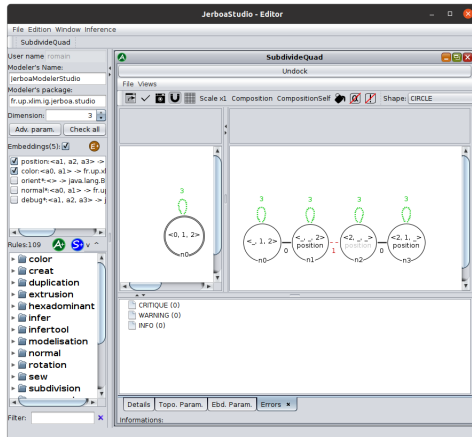
Operations:

Deselect Center view

alpha 3

Dot size: 5

Mem info: 4%



Graph rewriting

- ▶ Generalising string (or term) rewriting to non-linear structures.

What is a graph ?

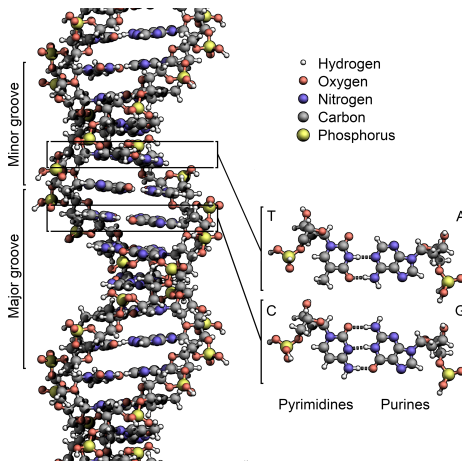
What is a graph ?

- A set of objects, or elements and a set of relations between them.



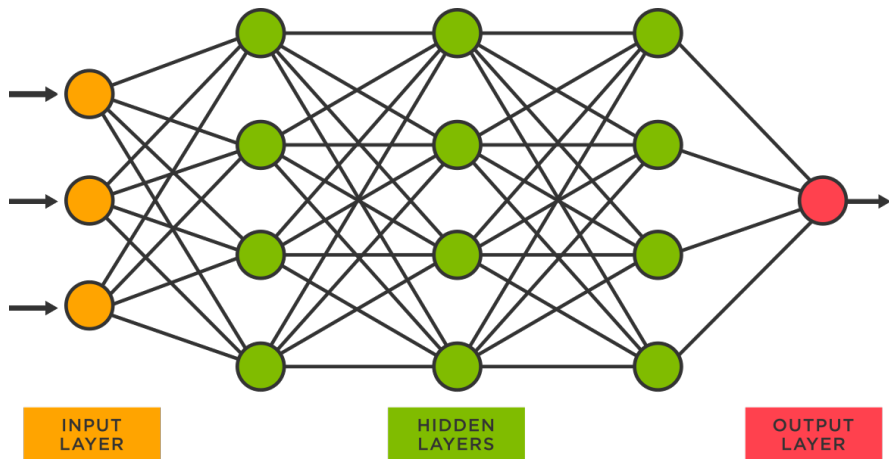
What is a graph ?

- A set of objects, or elements and a set of relations between them.



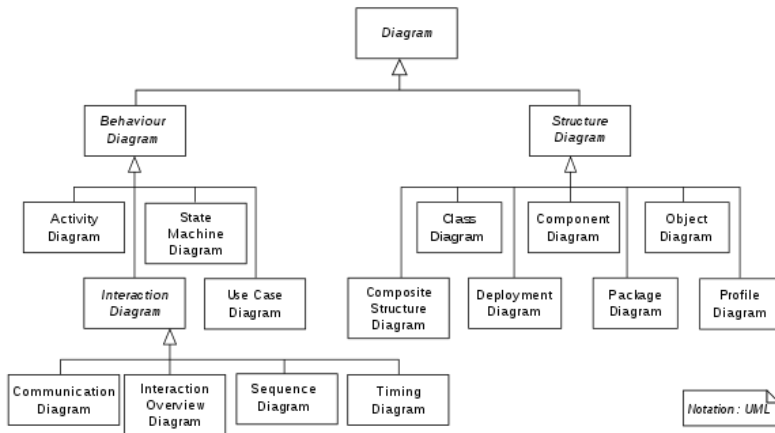
What is a graph ?

- ▶ A set of objects, or elements and a set of relations between them.



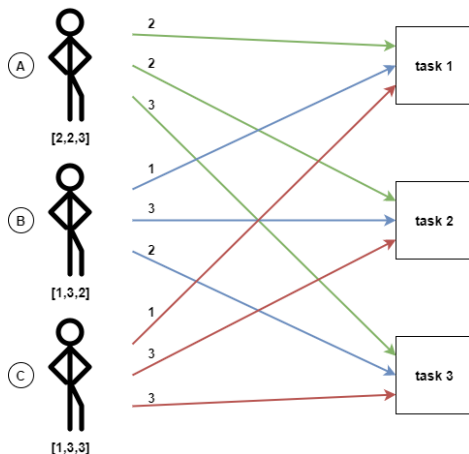
What is a graph ?

- A set of objects, or elements and a set of relations between them.



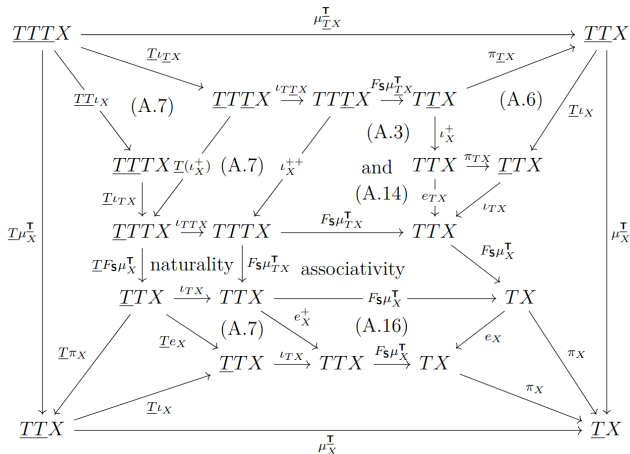
What is a graph ?

- A set of objects, or elements and a set of relations between them.



What is a graph ?

- A set of objects, or elements and a set of relations between them.



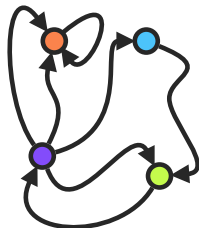
What is a graph ?

- A set of objects, or elements and a set of relations between them.

Graph (Directed multigraph with loops)

A **graph** $G = (V, E, s, t)$ consists of

- a set of **nodes** V ,
- a set of **arcs** E ,
- a **source function** $s : E \rightarrow V$,
- a **target arrow** $t : E \rightarrow V$,



Rewriting

Rewriting

String rewriting :

- an alphabet Σ
- a set of rewriting rules $u \rightarrow v$ (u and v are words on Σ^*)

Rewriting

String rewriting :

- an alphabet Σ
- a set of rewriting rules $u \rightarrow v$ (u and v are words on Σ^*)

sleep $\rightarrow$ follow

Rewriting

String rewriting :

- an alphabet Σ
- a set of rewriting rules $u \rightarrow v$ (u and v are words on Σ^*)

sleep $\rightarrow$ follow

You are all sleeping!

Rewriting

String rewriting :

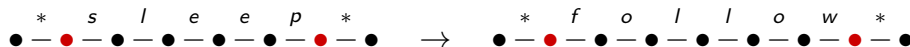
- an alphabet Σ
- a set of rewriting rules $u \rightarrow v$ (u and v are words on Σ^*)

sleep $\rightarrow$ follow

Youareallsleeping! $\rightarrow$ Youareallfollowing!

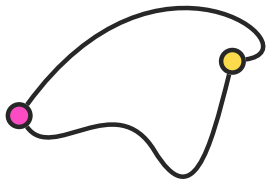
How to rewrite graphs?

How to rewrite graphs? (based on [Ehrig 1979])



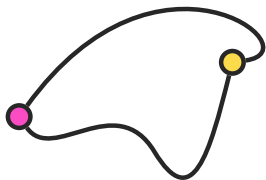
How to rewrite graphs? (based on [Ehrig 1979])

$\bullet - \overset{*}{\bullet} - \overset{s}{\bullet} - \overset{l}{\bullet} - \overset{e}{\bullet} - \overset{e}{\bullet} - \overset{p}{\bullet} - \overset{*}{\bullet} - \bullet \rightarrow \bullet - \overset{*}{\bullet} - \overset{f}{\bullet} - \overset{o}{\bullet} - \overset{l}{\bullet} - \overset{l}{\bullet} - \overset{o}{\bullet} - \overset{w}{\bullet} - \overset{*}{\bullet} - \bullet$



How to rewrite graphs? (based on [Ehrig 1979])

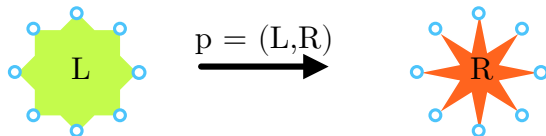
$\bullet - \overset{*}{\bullet} - \overset{s}{\bullet} - \bullet - \overset{l}{\bullet} - \bullet - \overset{e}{\bullet} - \bullet - \overset{e}{\bullet} - \bullet - \overset{p}{\bullet} - \overset{*}{\bullet} - \bullet$
 $\rightarrow$
 $\bullet - \overset{*}{\bullet} - \overset{f}{\bullet} - \bullet - \overset{o}{\bullet} - \bullet - \overset{l}{\bullet} - \bullet - \overset{l}{\bullet} - \bullet - \overset{o}{\bullet} - \bullet - \overset{w}{\bullet} - \overset{*}{\bullet} - \bullet$



- ▶ No notion of beginning and end in a graph.
- Identify the "gluing" elements.

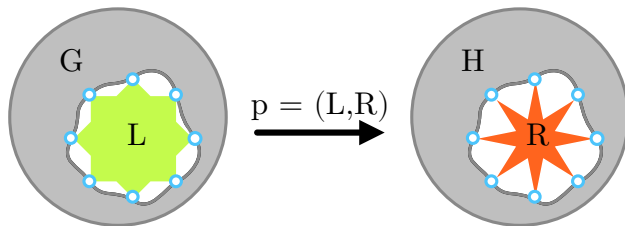
Graph transformation rules

- **Goal** : Generalise from strings to graphs.



Graph transformation rules

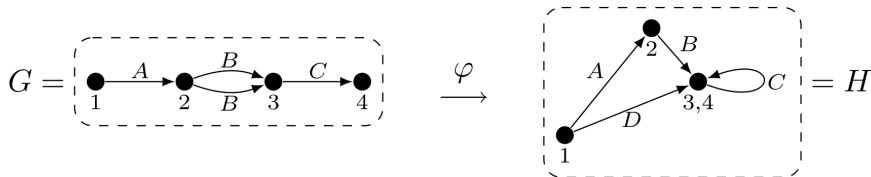
- **Goal** : Generalise from strings to graphs.



- Find in G a graph similar to L ,
- Remove it from G ,
- Reconnect R within the more global context.

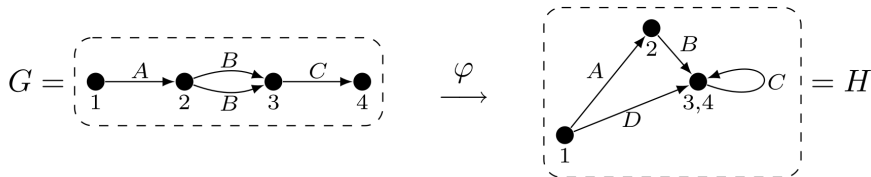
How to map graphs? (from [König 18])

Graph **morphism** :



How to map graphs? (from [König 18])

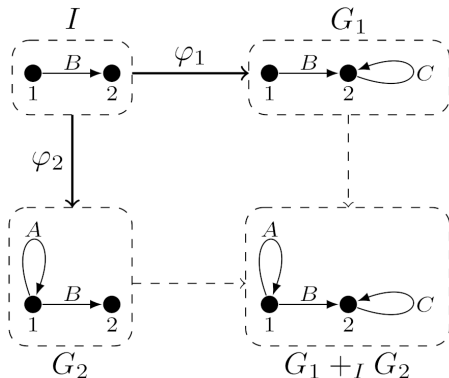
Graph **morphism** :



► Functions on nodes and arcs that preserve structure.

How to glue graphs? (from [König 18])

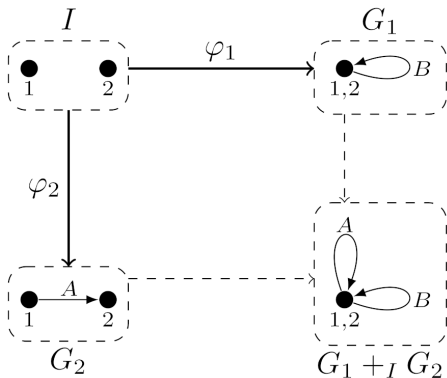
Graph **gluing** :



► $\sim$ Quotiented disjoint union.

How to glue graphs? (from [König 18])

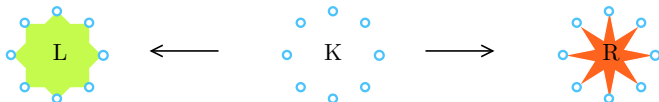
Graph **gluing** :



► $\sim$ Quotiented disjoint union.

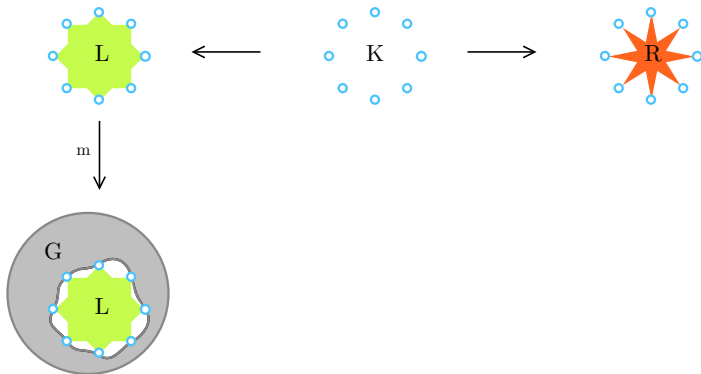
Back to graph transformation rules

- ▶ L , K , R , G , D , H are graphs.
- ▶ Arrows are graph morphisms.
- ▶ Squares are graph gluings.



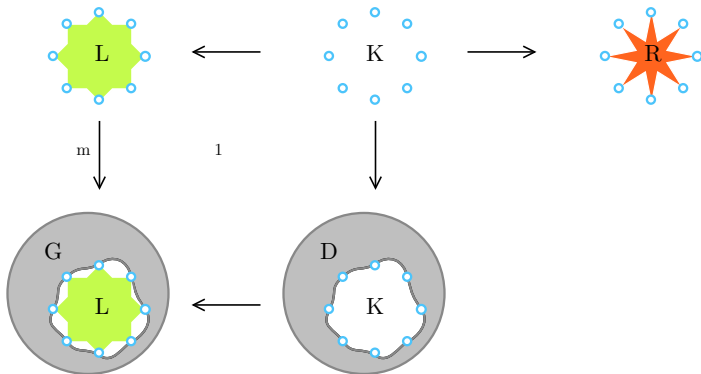
Back to graph transformation rules

- ▶ L , K , R , G , D , H are graphs.
- ▶ Arrows are graph morphisms.
- ▶ Squares are graph gluings.



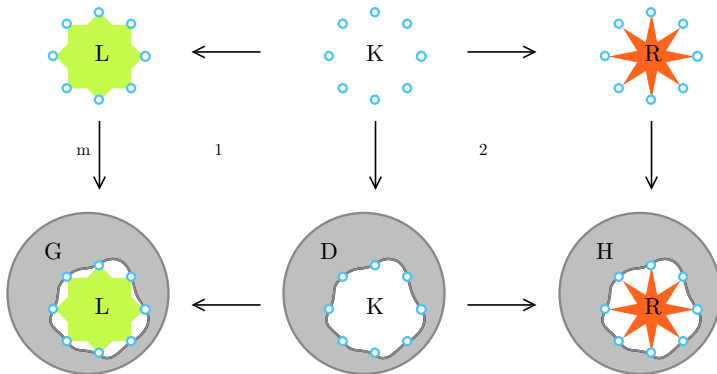
Back to graph transformation rules

- ▶ L , K , R , G , D , H are graphs.
- ▶ Arrows are graph morphisms.
- ▶ Squares are graph gluings.

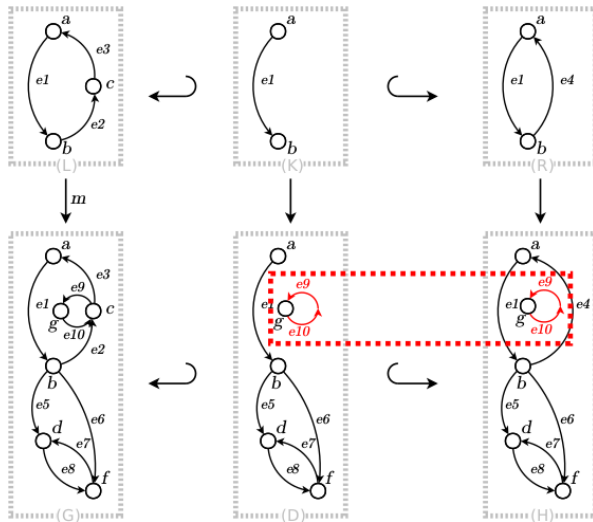


Back to graph transformation rules

- ▶ L , K , R , G , D , H are graphs.
- ▶ Arrows are graph morphisms.
- ▶ Squares are graph gluings.



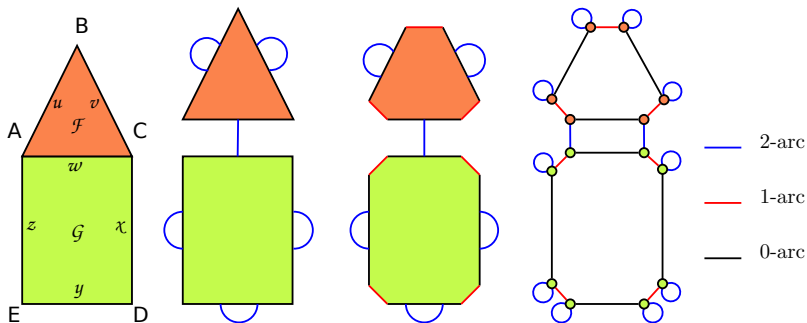
Why does it has to be so complicated?



Topology-based geometric modeling

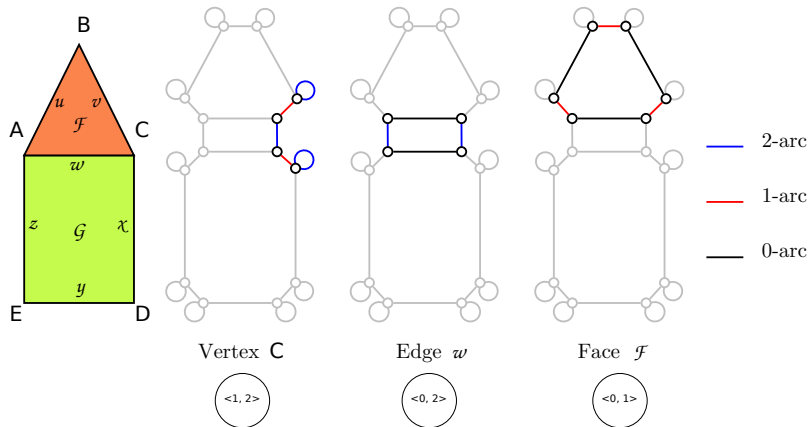
- ▶ Description of object via topological cells.

Generalized maps [Lienhardt 89]



► Defined as graphs using labels, attributes and constraints.

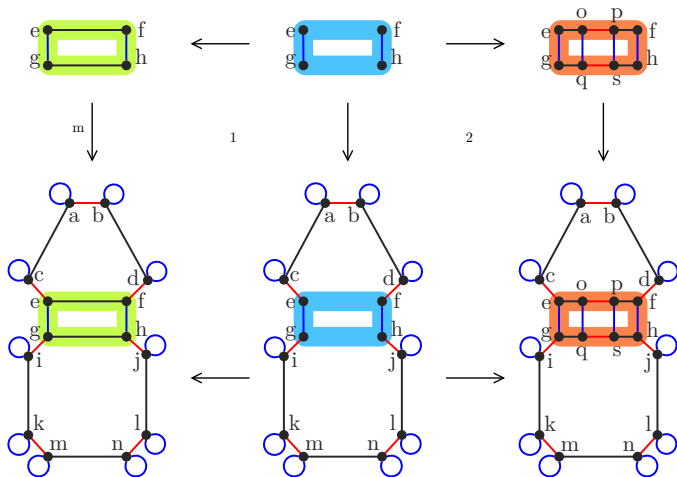
Orbits



► Defined as subgraph induced by a subset of dimensions.

Rewriting G-maps

- Most conservative framework (all morphisms are injectives).



Research Questions

- ▶ What am I doing with all of this?

How to preserve the consistency of the model?

- ▶ **Topological constraints** : structure,
 - E.g., vertices are incident to edges that are incident to faces.

How to preserve the consistency of the model?

- ▶ **Topological constraints** : structure,
 - E.g., vertices are incident to edges that are incident to faces.
- ▶ **Embedding constraints** : geometry,
 - E.g., all elements defining the same vertex should share the same position.

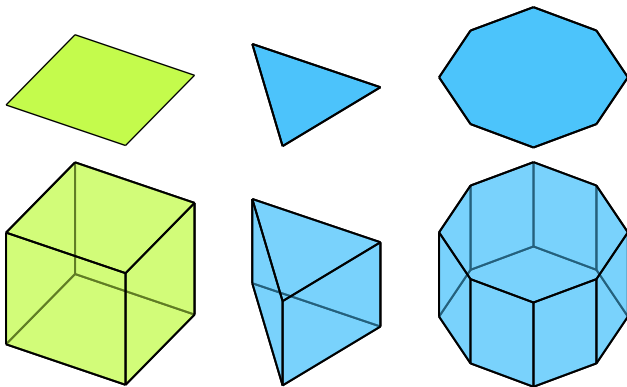
How to preserve the consistency of the model?

- ▶ **Topological constraints** : structure,
 - E.g., vertices are incident to edges that are incident to faces.
- ▶ **Embedding constraints** : geometry,
 - E.g., all elements defining the same vertex should share the same position.

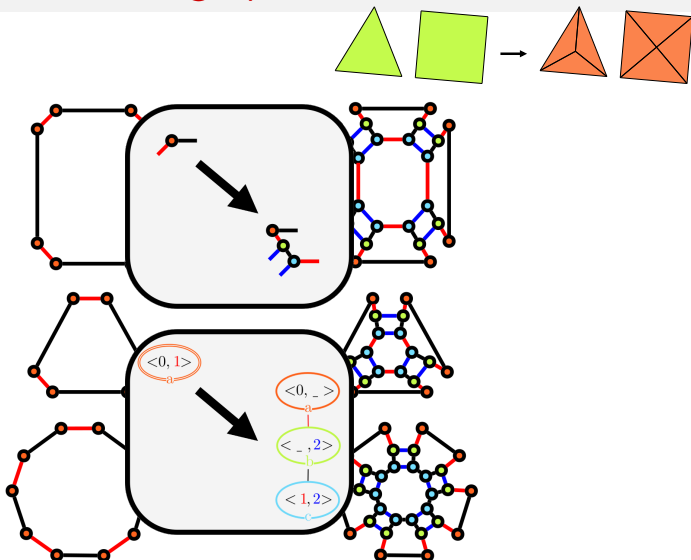
Goal : The modification of a well-formed object should provide a well-formed object.

How to define modeling operations?

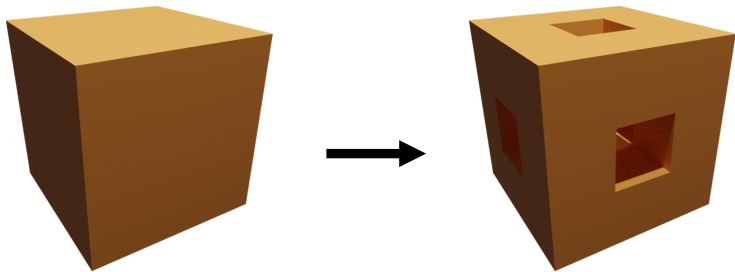
$\langle 0, 1 \rangle$



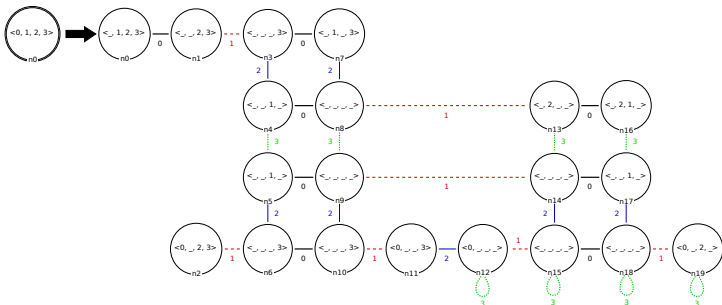
How to define modeling operations?



How to simplify the creation of new rules?



How to simplify the creation of new rules?



How to simplify the creation of new rules?

