Consistent geometric modeling operations

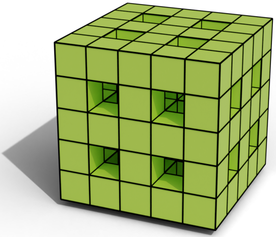
An application of graph transformations

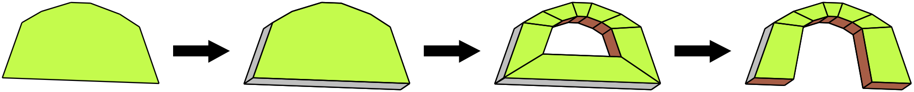
Romain Pascual

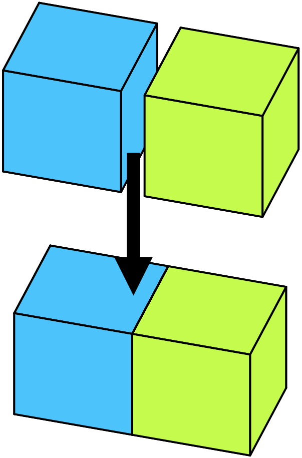
romainpascual.fr

November 16, 2023

KIT Seminar



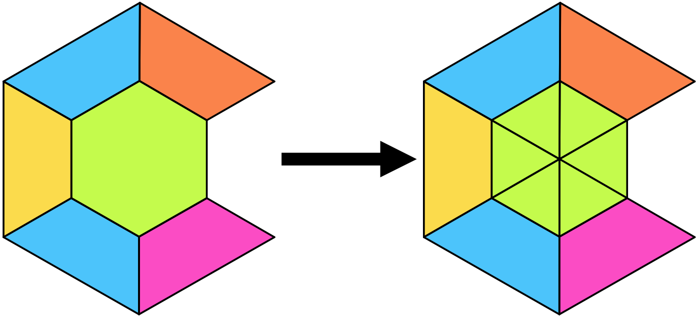




CGAL's sew operation

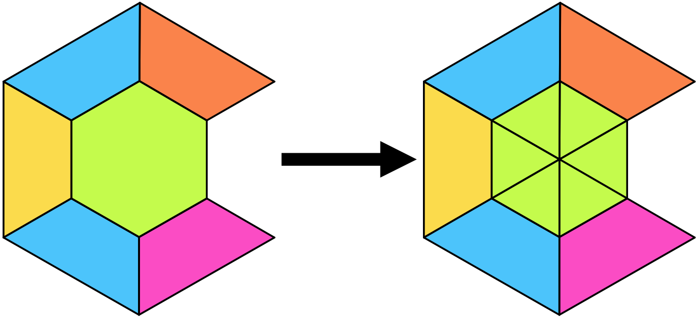
```
template<unsigned int i>
void sew(Dart_descriptor adart1, Dart_descriptor adart2)

  CGAL_assertion( i<=dimension );
  CGAL_assertion( (is_sewable<i>(adart1,adart2)) );
  size_type amark=get_new_mark();
  CGAL::GMap_dart_iterator_basic_of_involution<Self, i>
    I1(*this, adart1, amark);
  CGAL::GMap_dart_iterator_basic_of_involution<Self, i>
    I2(*this, adart2, amark);
  for ( ; I1.cont(); ++I1, ++I2 )
  {
    Helper::template Foreach_enabled_attributes_except
      <CGAL::internal::GMap_group_attribute_functor<Self, i>, i>::
      run(*this, I1, I2);
  }
  negate_mark( amark );
  for ( I1.rewind(), I2.rewind(); I1.cont(); ++I1, ++I2 )
  {
    basic_link_alpha<i>(I1, I2);
  }
  negate_mark( amark );
  CGAL_assertion( is_whole_map_unmarked(amark) );
  free_mark(amark);
}
```



Ambition : define a *domain-specific language* (DSL) for geometric modeling

Motivations : abstraction, performance, conciseness, correctness

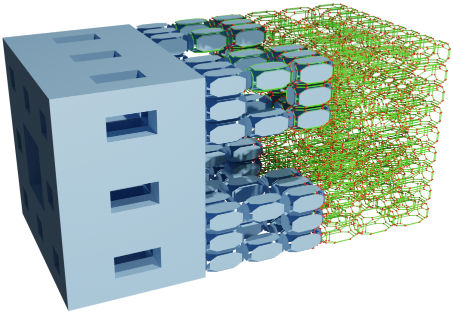


Ambition : define a *domain-specific language* (DSL) for geometric modeling

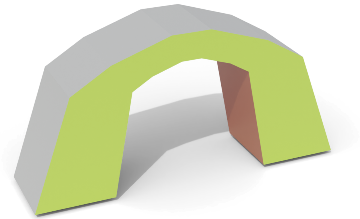
Motivations : abstraction, performance, conciseness, correctness (consistency)

Embedded generalized maps

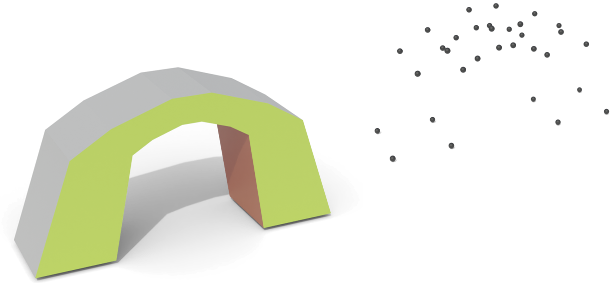
- How to represent objects?



Topological cells

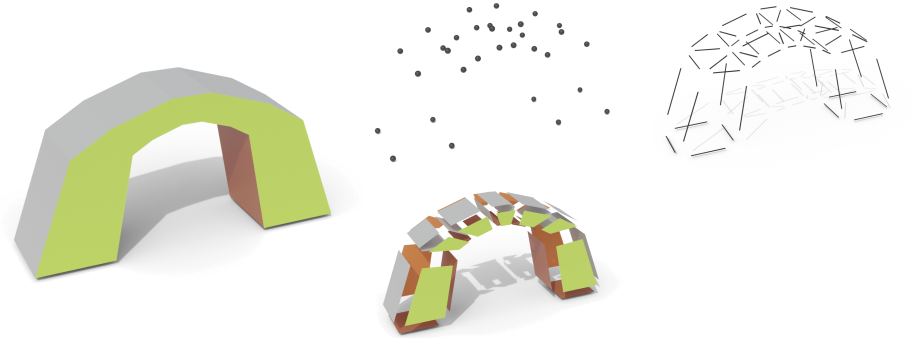


Topological cells

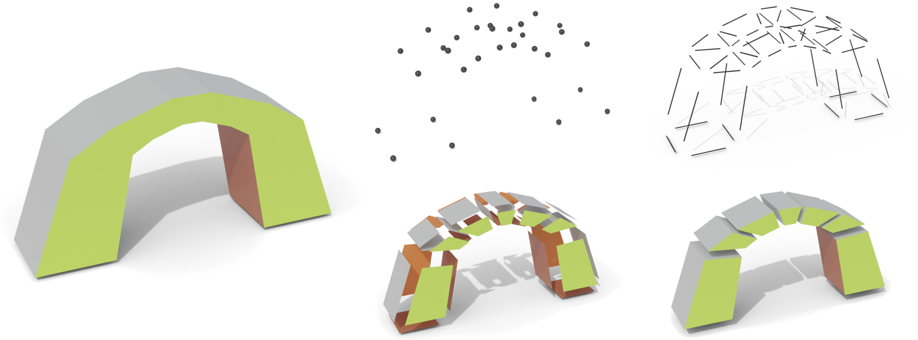


Topological cells

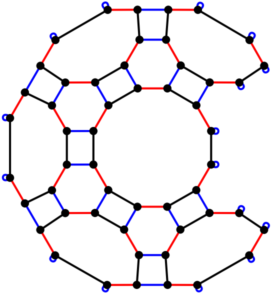
Topological cells



Topological cells



Generalized maps[1] (topology)

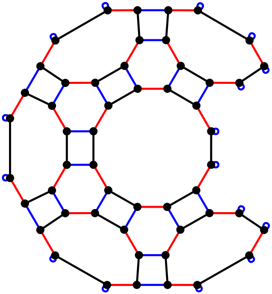


Legend: 0, 1, 2

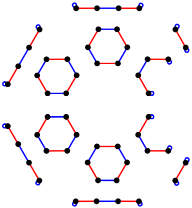
[1]Damiand et al. 2014.

Generalized maps[1] (topology)

Orbit: Sub-graph induced by a subset $\langle o \rangle$ of dimensions



Legend: 0, 1, 2

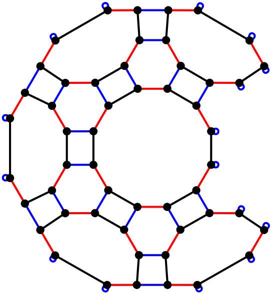


Vertices: orbits $\langle 1, 2 \rangle$

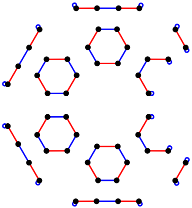
[1] Damiand et al. 2014.

Generalized maps[1] (topology)

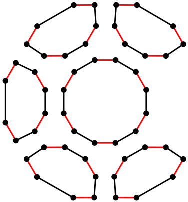
Orbit: Sub-graph induced by a subset $\langle o \rangle$ of dimensions



Legend: 0, 1, 2



Vertices: orbits $\langle 1, 2 \rangle$



Faces: orbits $\langle 0, 1 \rangle$

[1] Damiand et al. 2014.

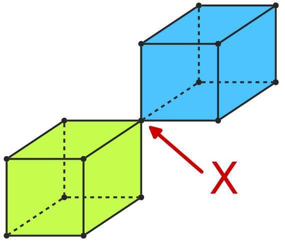
Topological consistency

Any graph with topological information is not a valid Gmap

Topological consistency

Any graph with topological information is not a valid Gmap

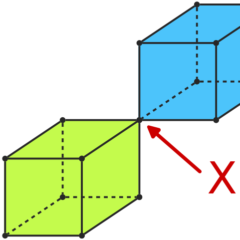
Constraint: n-cells should be glued along $(n-1)$-cells

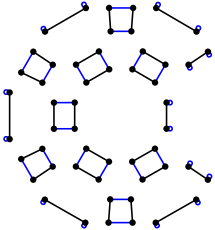


Topological consistency

Any graph with topological information is not a valid Gmap

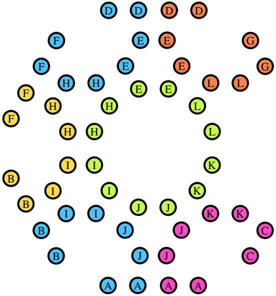
Constraint: n-cells should be glued along $(n-1)$-cells





Example of constraint: 0202-paths should be cycles

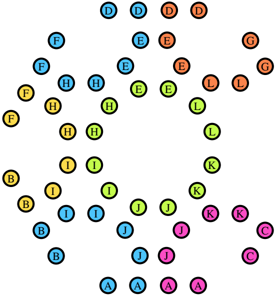
Embeddings (geometry)



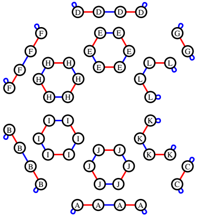
Legend: 0, 1, 2

Embeddings (geometry)

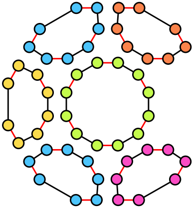
Embedding: function $\pi : \langle o_\pi \rangle \to \tau_\pi$
with τ_π an abstract data type



Legend: 0, 1, 2



position : $\langle 1, 2 \rangle \to$ `Point3`



color : $\langle 0, 1 \rangle \to$ `ColorRGB`

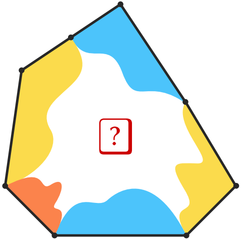
Geometric consistency

Any Gmap with embedding information is not a valid embedded Gmap

Geometric consistency

Any Gmap with embedding information is not a valid embedded Gmap

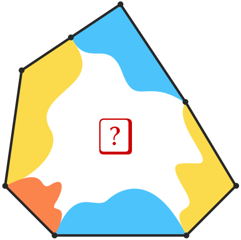
Constraint: *n*-cells can only have one value per embedding

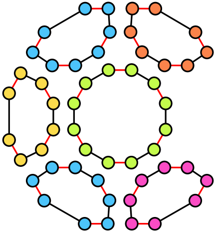


Geometric consistency

Any Gmap with embedding information is not a valid embedded Gmap

Constraint: *n*-cells can only have one value per embedding

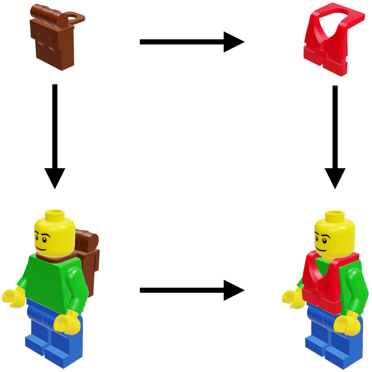




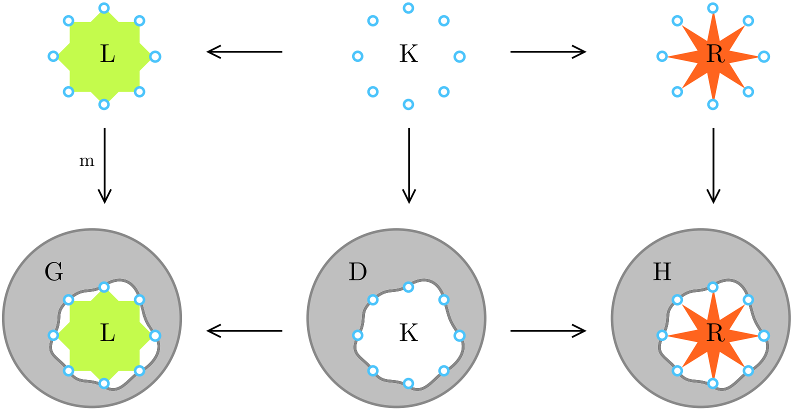
Example of constraint: nodes in a $\langle 0, 1 \rangle$-orbit should have the same color

Graph rewriting

- How to formalize object transformations?

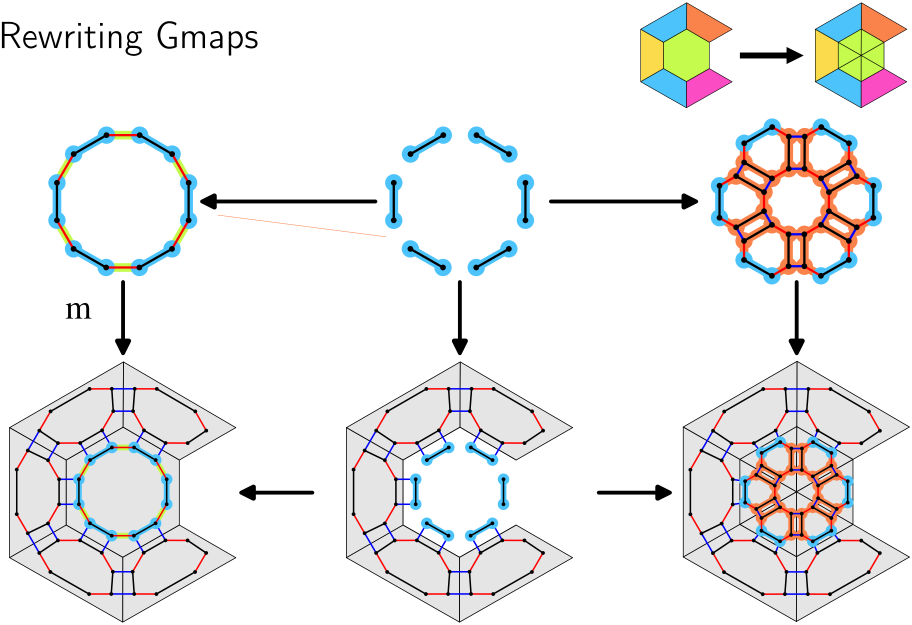


Graph transformation rules[1]

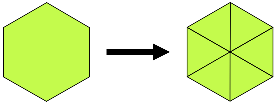


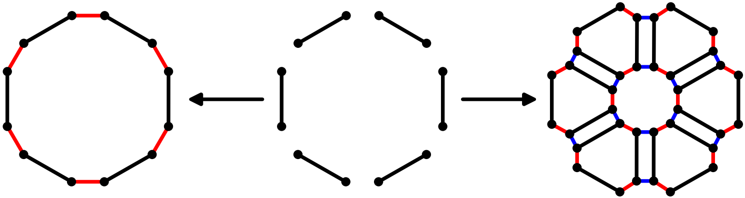
[1] Rozenberg 1997; Ehrig et al. 2006; Heckel et al. 2020.

Rewriting Gmaps

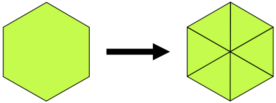


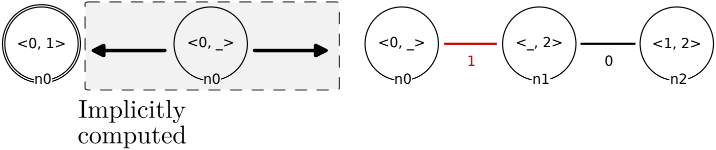
Orbit rewriting

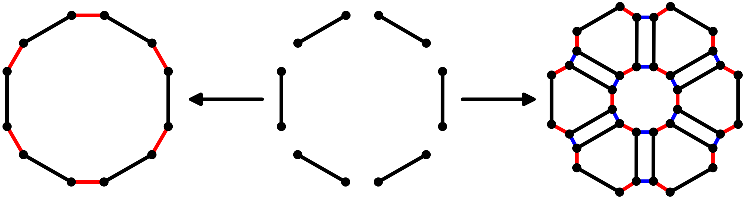




Orbit rewriting

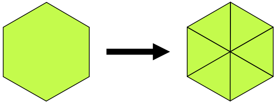


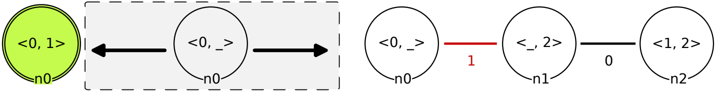




Instantiated rule

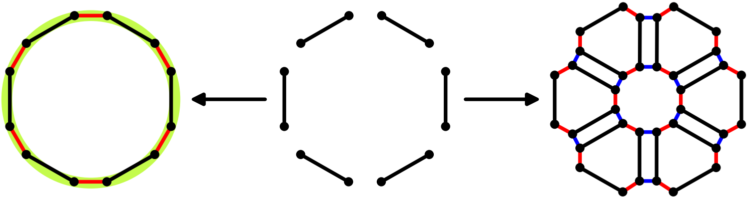
Orbit rewriting





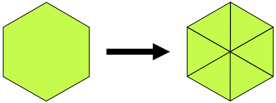
Implicitly computed

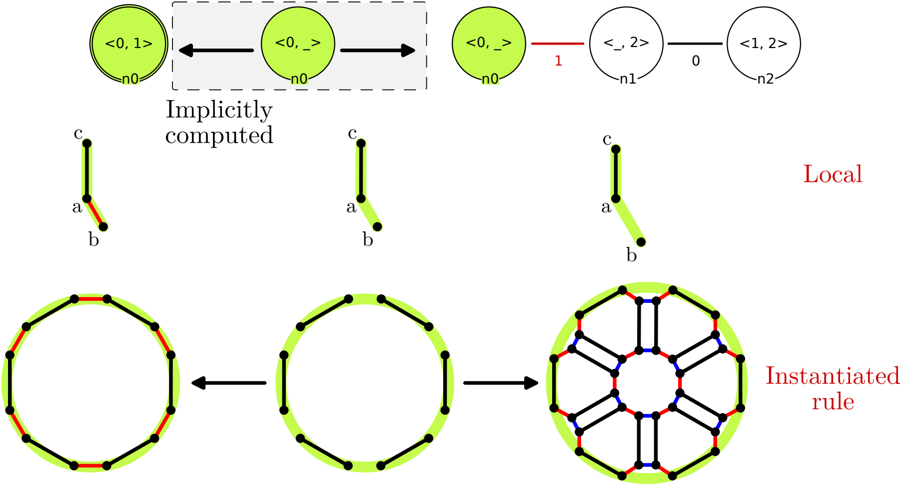
Local



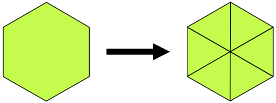
Instantiated rule

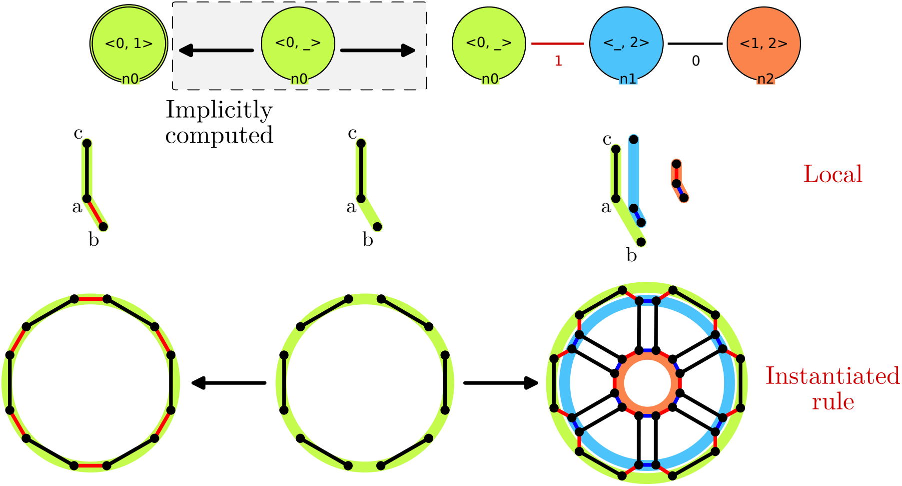
Orbit rewriting



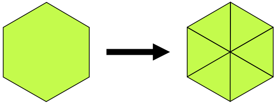


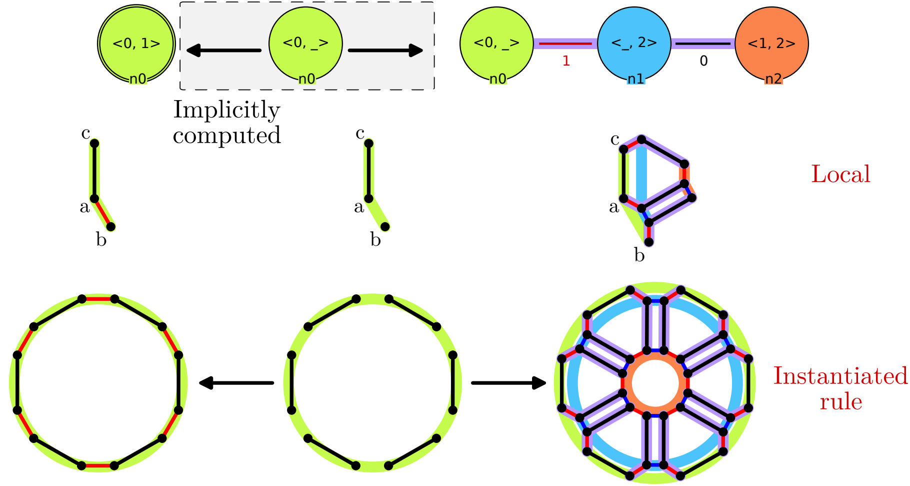
Orbit rewriting



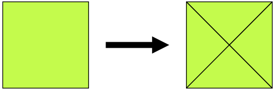


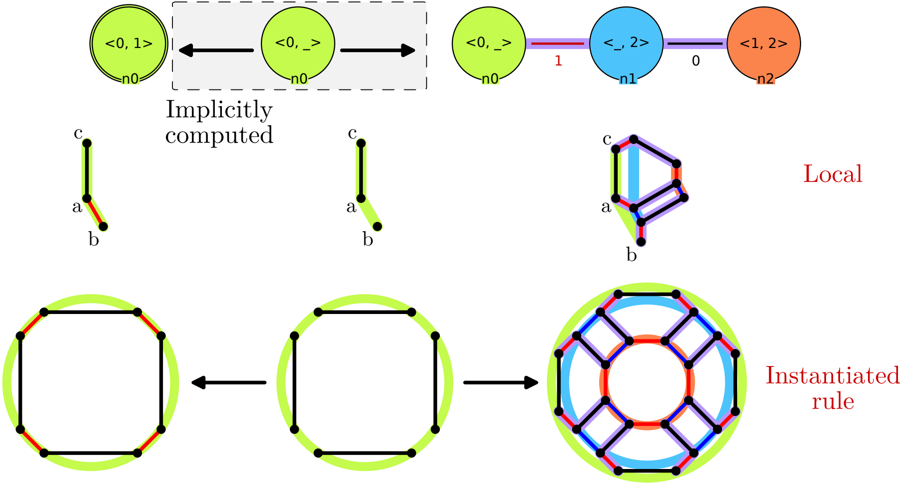
Orbit rewriting



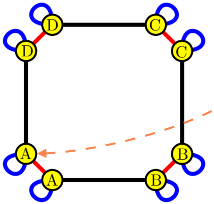


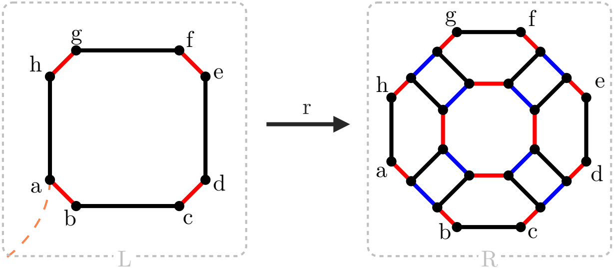
Orbit rewriting





Modifying geometric values[1]

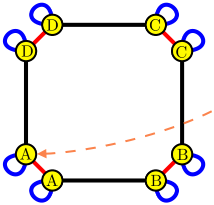


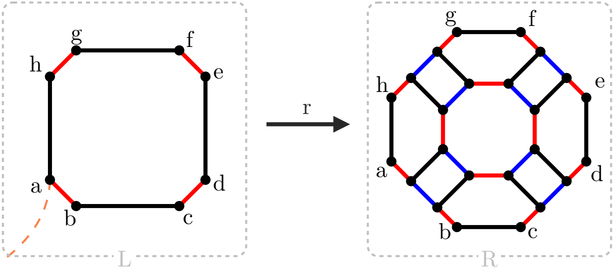


Algebraic data types (attributes)

[1] Bellet et al. 2017.

Modifying geometric values[1]





Algebraic data types, three kinds of expressions

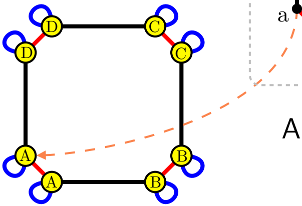
- Accessors

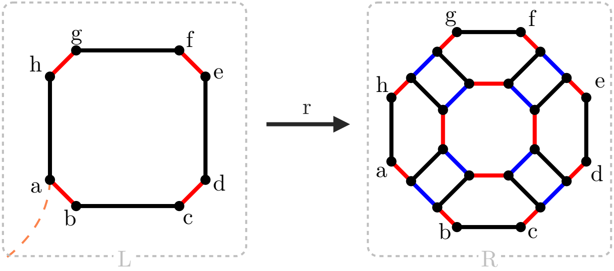
  a.color = ●

  a.position = A

[1]Bellet et al. 2017.

Modifying geometric values[1]





Algebraic data types, three kinds of expressions

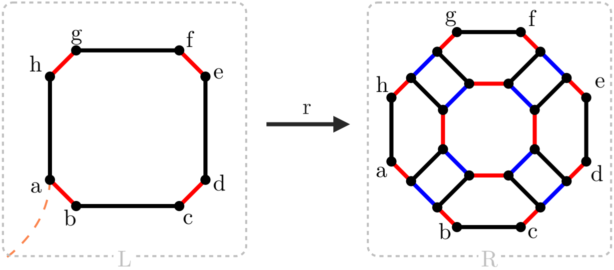
- Accessors
- Computations

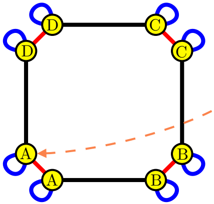
  ● + ● = ●

  center({●, ●}) = ●

[1] Bellet et al. 2017.

Modifying geometric values[1]



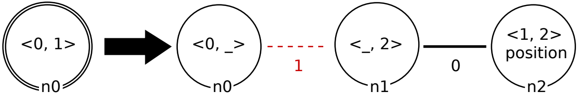


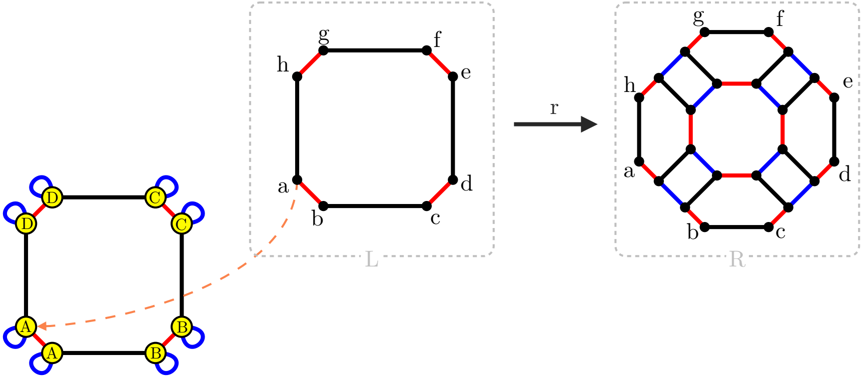
Algebraic data types, three kinds of expressions

- Accessors
- Computations
- Topological operators
  a@0.position $= D$
  $\text{position}_{\langle 0,1 \rangle}(a) = \{A, B, C, D\}$

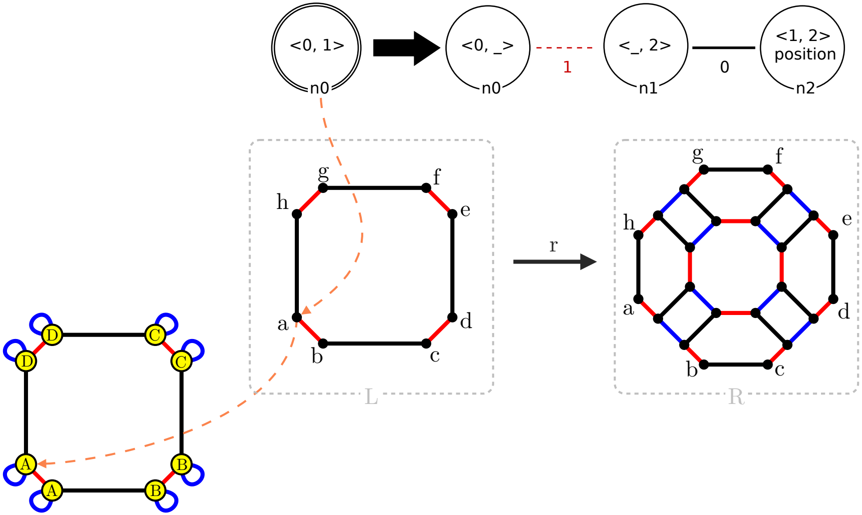
[1]Bellet et al. 2017.

Extension to schemes

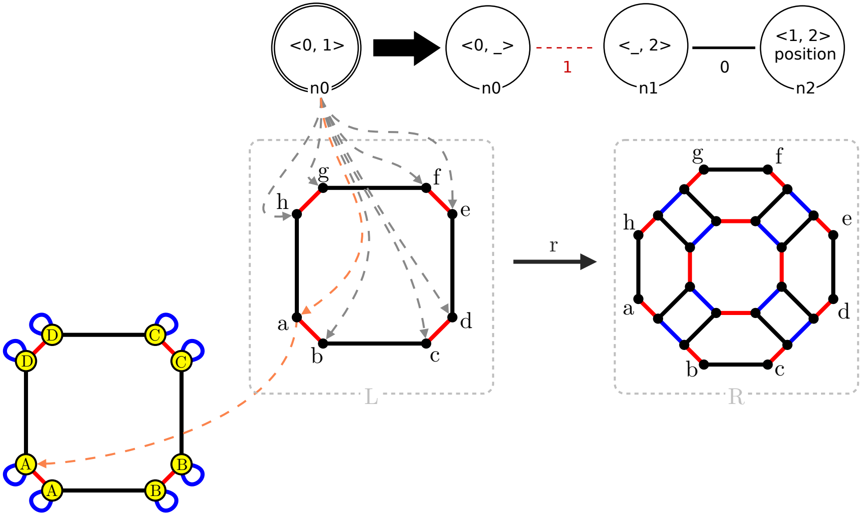




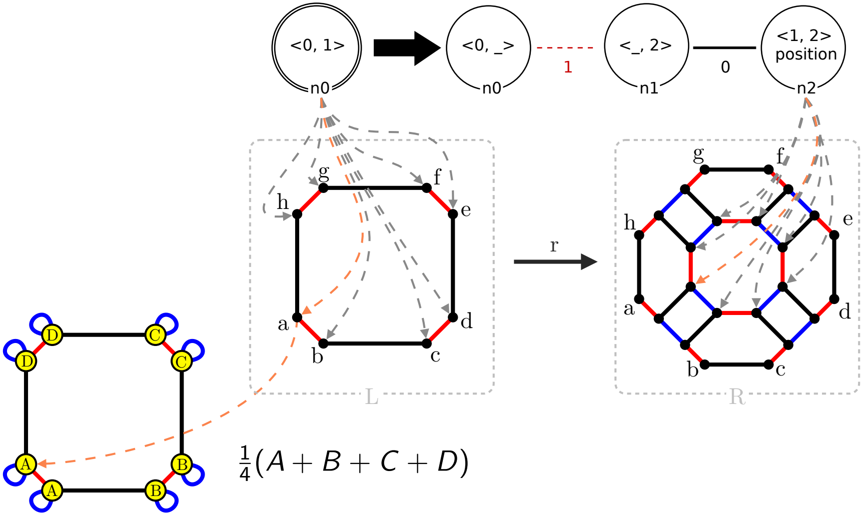
Extension to schemes



Extension to schemes

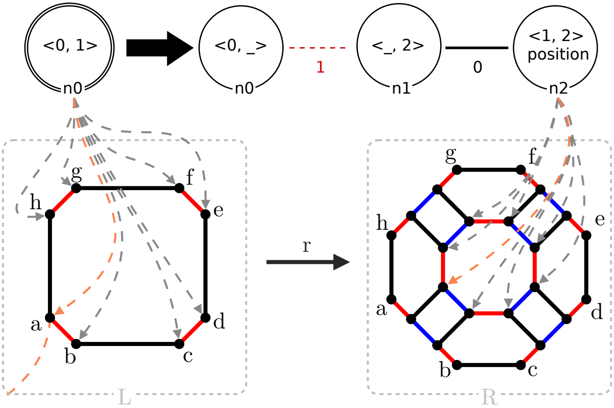


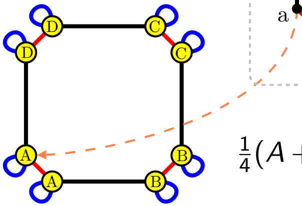
Extension to schemes



$$\frac{1}{4}(A + B + C + D)$$

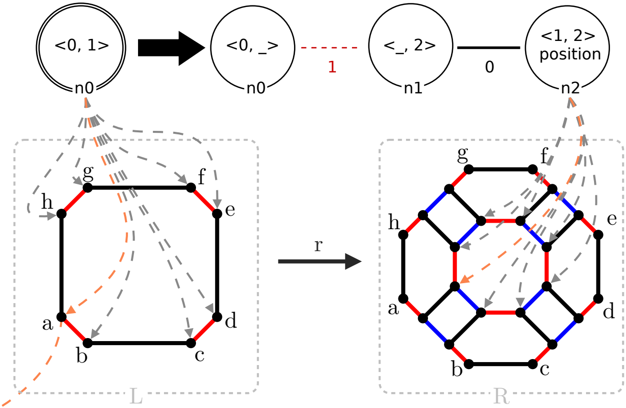
Extension to schemes

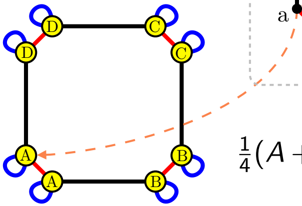




$$\frac{1}{4}(A + B + C + D) = middle(\{A, B, C, D\})$$

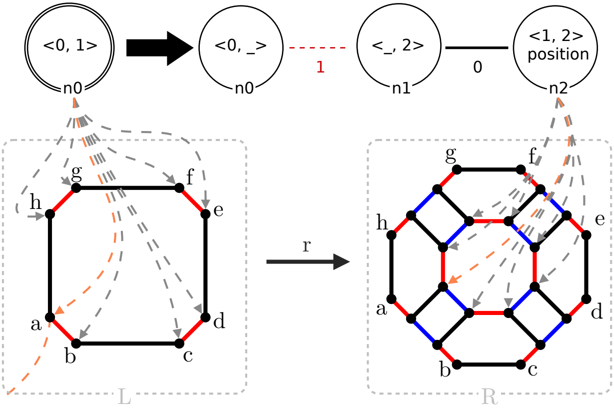
Extension to schemes

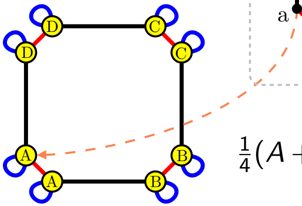




$$\frac{1}{4}(A+B+C+D) = middle(\{A,B,C,D\})$$
$$= middle(position_{\langle 0,1 \rangle}(a))$$

Extension to schemes





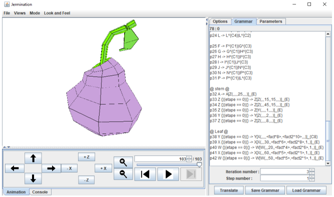
$$\frac{1}{4}(A + B + C + D) = middle(\{A, B, C, D\})$$
$$= middle(position_{\langle 0,1\rangle}(a))$$
$$= middle(position_{\langle 0,1\rangle}(n0))$$

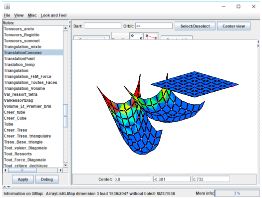
A rule-based language

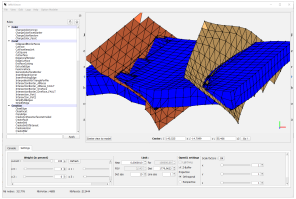
Topology : categorical semantics for operations

Geometry : structure-based algebraic formalisation

Formalizing the DSL of Jerboa

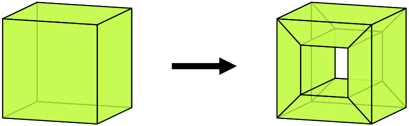


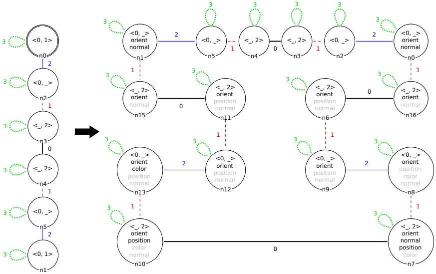




Consistent modeling operations

- How to preserve the model's constraints?





Consistency preservation

Modifications of a well-formed object should produce an equally well-formed object

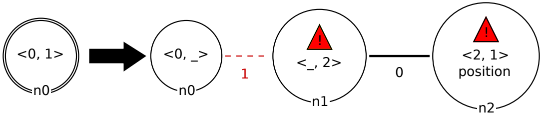
Requirement: Feedback to the rule designer

Consistency preservation

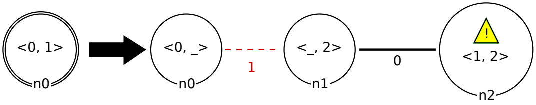
Modifications of a well-formed object should produce an equally well-formed object

Requirement: Feedback to the rule designer

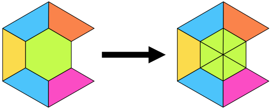
- Topological inconsistencies



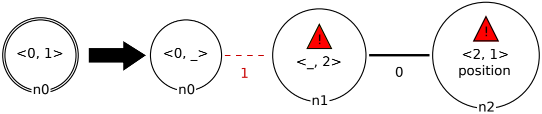
- Geometric inconsistencies



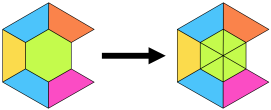
Breaking the topological consistency



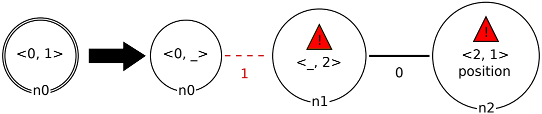
Constraint: 0202-paths should be cycles

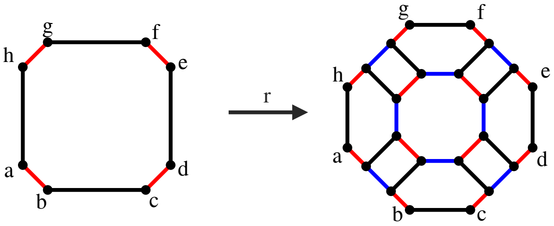


Breaking the topological consistency

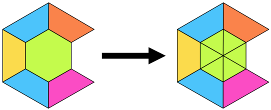


Constraint: 0202-paths should be cycles

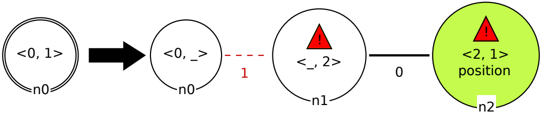


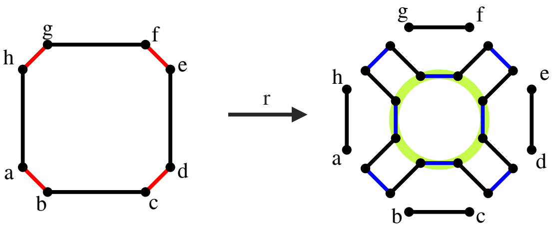


Breaking the topological consistency

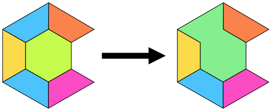


Constraint: 0202-paths should be cycles

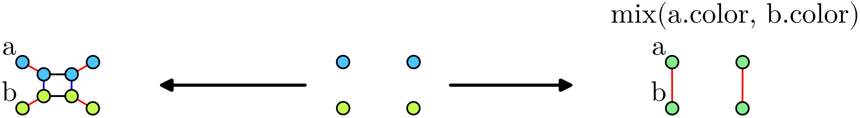




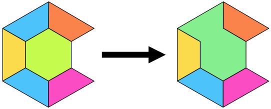
Breaking the geometric consistency



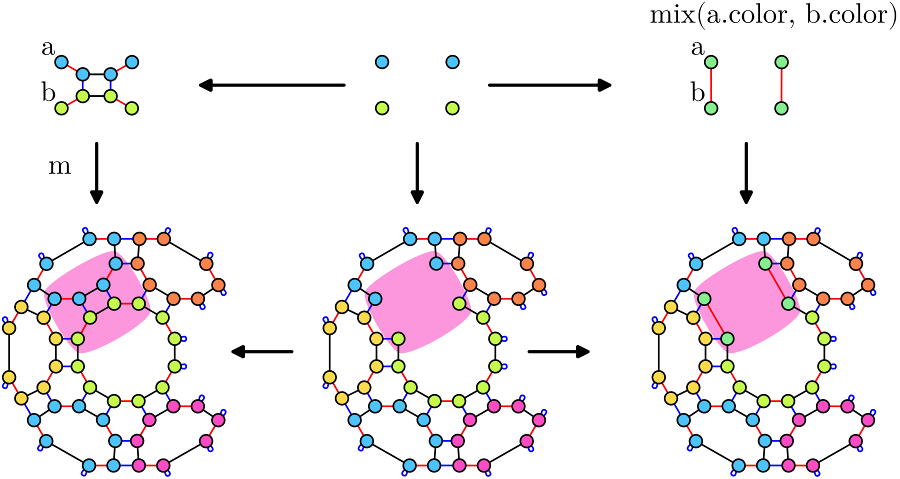
Constraint: nodes in a $\langle 0, 1 \rangle$-orbit should have the same color



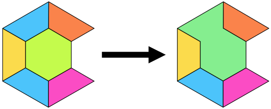
Breaking the geometric consistency



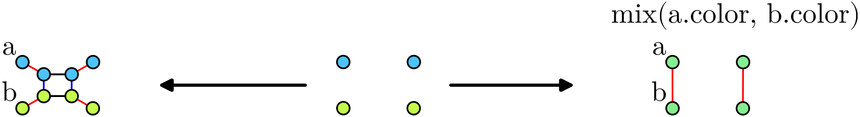
Constraint: nodes in a $\langle 0, 1 \rangle$-orbit should have the same color



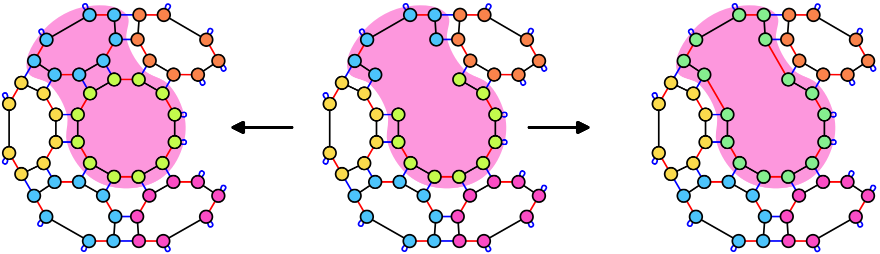
Breaking the geometric consistency

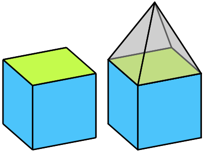


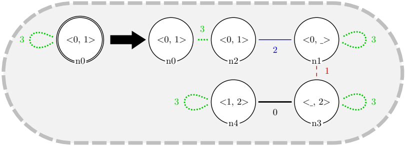
Constraint: nodes in a $\langle 0, 1 \rangle$-orbit should have the same color



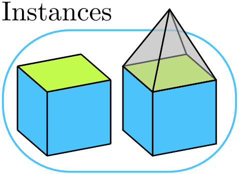
Rule completion

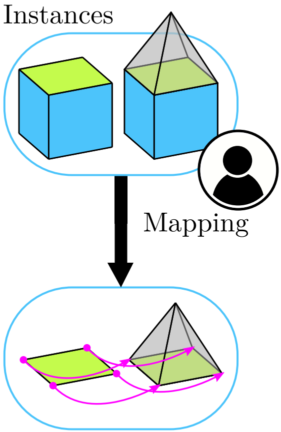


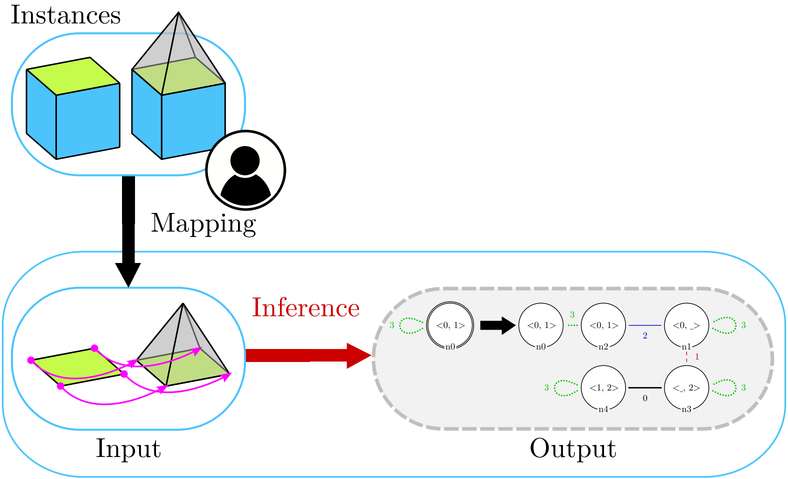


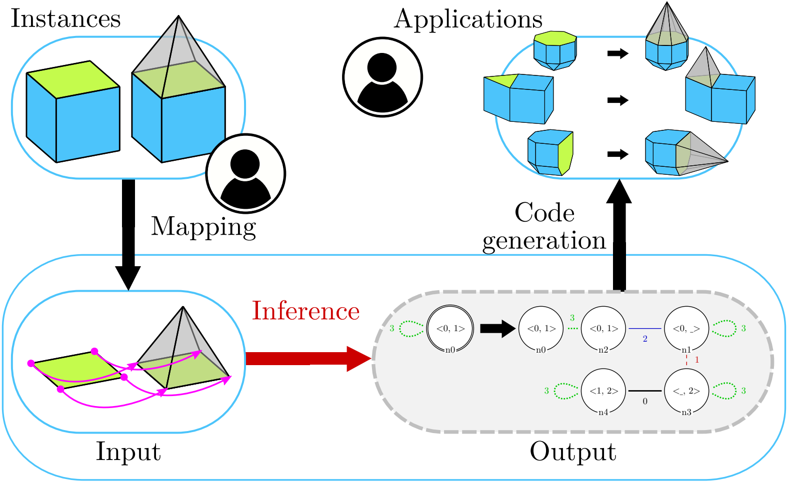

3
<0, 1>
n0
<0, 1>
n0
3
<0, 1>
n2
2
<0, _>
n1
3
1
3
<1, 2>
n4
0
<_, 2>
n3
3

Instances

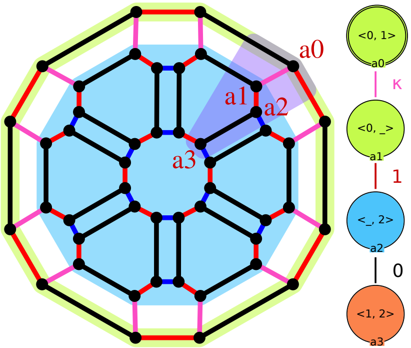



Instances
Mapping


Instances
Mapping
Inference
Input
Output
<0, 1>
n0
<0, 1>
n0
<0, 1>
n2
<0, _>
n1
<1, 2>
n4
<_, 2>
n3
3
2
1
0


Instances
Applications
Mapping
Code
generation
Inference
Input
Output
3
<0, 1>
n0
<0, 1>
n0
3
<0, 1>
n2
2
<0, _>
n1
3
1
3
<1, 2>
n4
0
<_, 2>
n3
3

Inference of modeling operations



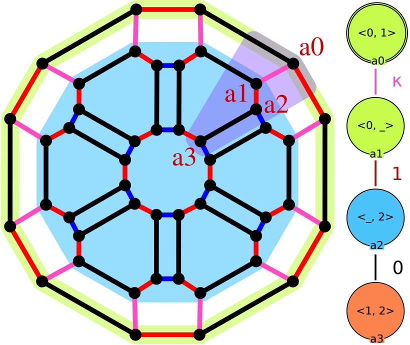
Color legend: 0, 1, 2, κ.

Graph traversal with quotient

Theorem

The algorithm produces a topological folding whenever it exists or the information that no such folding exists.

Inference of modeling operations



Color legend: 0, 1, 2, κ.

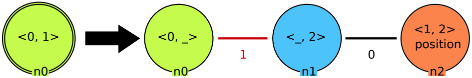
Graph traversal with quotient

Theorem

The algorithm produces a topological folding whenever it exists or the information that no such folding exists.

The consistency conditions on the rule provide a search space in which we retrieve the operation

Main contibutions



Topological consistency: path analysis on rule schemes

- Romain Pascual et al. (2022b). "Topological consistency preservation with graph transformation schemes". In: Science of Computer Programming. DOI: 10.1016/j.scico.2021.102728

Geometric consistency: rule completion

- Agnès Arnould et al. (2022). "Preserving consistency in geometric modeling with graph transformations". In: Mathematical Structures in Computer Science. DOI: 10.1017/S0960129522000226

Inference of operations: topological folding algorithm

- Romain Pascual et al. (2022a). "Inferring topological operations on generalized maps: Application to subdivision schemes". In: Graphics and Visual Computing. DOI: 10.1016/j.gvc.2022.200049

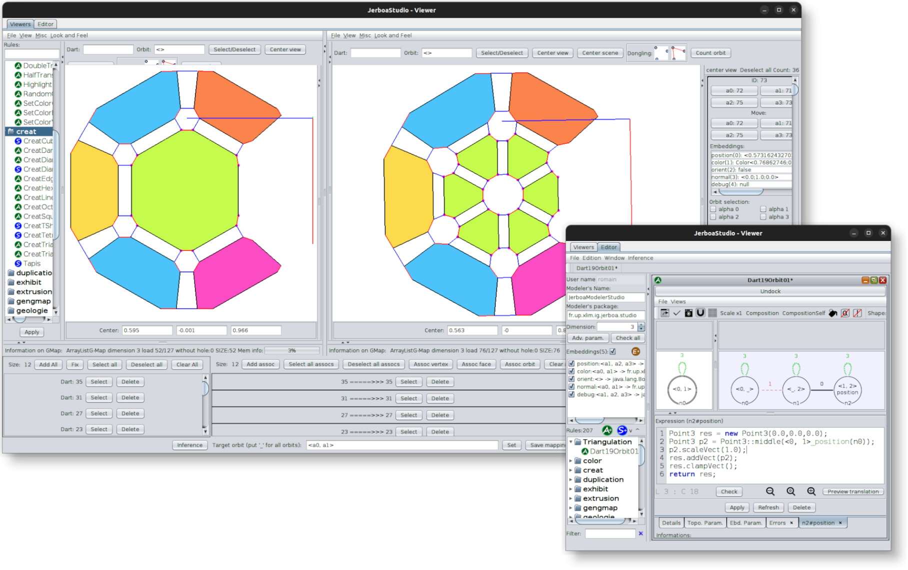
Current research projects

Following up on the formalization of Jerboa

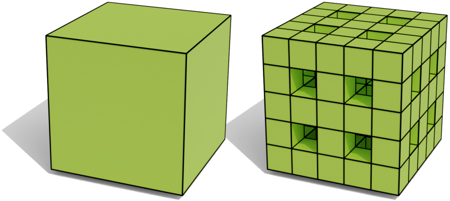
- Multi-cell query-replace approach for combinatorial maps[1]
  Guillaume Damiand, Vincent Nivoliers and Jordan Goncalves (M2 intern)

- Towards a local calculus for nested conditions?[2]
  Nicolas Behr and Pascale Le Gall

[1] Damiand et al. 2022.
[2] Habel et al. 2009.

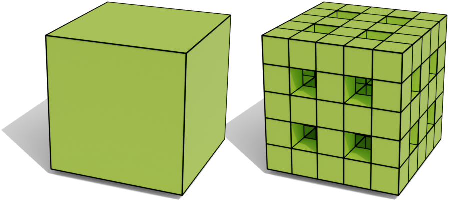

JerboaStudio - Viewer
Viewers Editor
File View Misc Look and Feel
Rules:
Dart: Orbit: <> Select/Deselect Center view
creat
duplication
exhibit
extrusion
gengmap
geologie
Apply
Center: 0.595 -0.001 0.966
Information on GMap: ArrayListG-Map dimension 3 load 52/127 without hole:0 SIZE:52 Mem info: 3%
Size: 12 Add All Fix Select all Deselect all Clear All
Dart: 35 Select Delete
Dart: 31 Select Delete
Dart: 27 Select Delete
Dart: 23 Select Delete
Dart: Orbit: <> Select/Deselect Center view Center scene Dongling: Count orbit
center view Deselect all Count: 36
ID: 73
Move:
Embeddings:
position(0): <0.57316243270
color(1): Color<0.76862746;0
orient(2): false
normal(3): <0.0;1.0;0.0>
debug(4): null
Orbit selection:
alpha 0 alpha 1
alpha 2 alpha 3
Center: 0.563 -0
Information on GMap: ArrayListG-Map dimension 3 load 76/127 without hole:0 SIZE:76
Size: 12 Add assoc Select all assocs Deselect all assocs Assoc vertex Assoc face Assoc orbit
35 =====>>> 35 Select Delete
31 =====>>> 31 Select Delete
27 =====>>> 27 Select Delete
23 =====>>> 23 Select Delete
Inference Target orbit (put '_' for all orbits): <a0, a1> Set
JerboaStudio - Viewer
Viewers Editor
File Edition Window Inference
Dart19Orbit01*
User name romain
Modeler's Name:
JerboaModelerStudio
Modeler's package:
fr.up.xlim.ig.jerboa.studio
Dimension: 3
Adv. param. Check all
Embeddings(5):
position:<a1, a2, a3>
color:<a0, a1>
orient:<> -> java.lang.Bo
normal:<a0, a1>
debug:<a1, a2, a3>
Rules:207
Triangulation
Dart19Orbit01
color
creat
duplication
exhibit
extrusion
gengmap
Filter:
Dart19Orbit01*
Undock
File Views
Scale x1 Composition CompositionSelf Shape:
Expression (n2#position)
Point3 res = new Point3(0.0,0.0,0.0);
Point3 p2 = Point3::middle(<0, 1>_position(n0));
p2.scaleVect(1.0);
res.addVect(p2);
res.clampVect();
return res;
Check Preview translation
Apply Refresh Delete
Details Topo. Param. Ebd. Param. Errors n2#position
Informations:

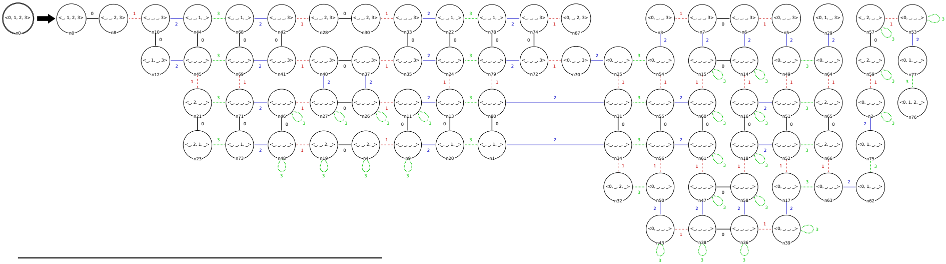
(2, 2, 2)-Menger polycube[1]



[1]Richaume et al. 2019.

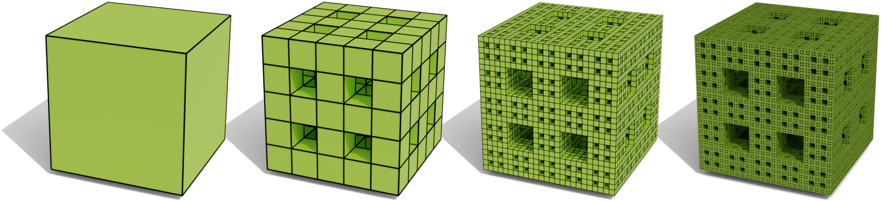
(2, 2, 2)-Menger polycube[1]

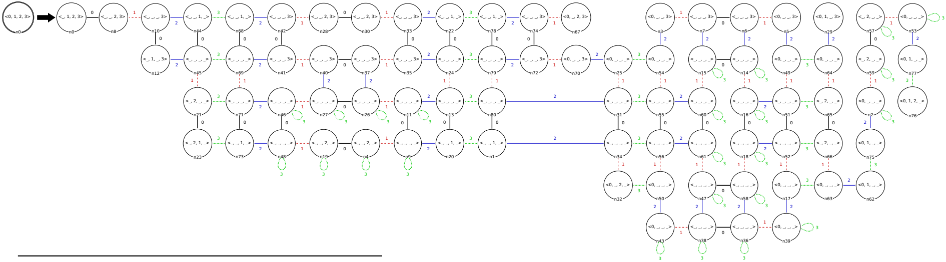




[1] Richaume et al. 2019.

(2, 2, 2)-Menger polycube[1]





[1]Richaume et al. 2019.

References I

Arnould, Agnès et al. (2022). "Preserving consistency in geometric modeling with graph transformations". In: Mathematical Structures in Computer Science. DOI: 10.1017/S0960129522000226.

Bellet, Thomas et al. (2017). "Geometric Modeling: Consistency Preservation Using Two-Layered Variable Substitutions". In: Graph Transformation (ICGT 2017). Ed. by Juan de Lara et al. Vol. 10373. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 36–53. ISBN: 978-3-319-61470-0. DOI: 10.1007/978-3-319-61470-0_3.

Damiand, Guillaume et al. (Sept. 19, 2014). Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing. CRC Press. 407 pp. ISBN: 978-1-4822-0652-4.

References II

- Damiand, Guillaume et al. (June 18, 2022). "Query-replace operations for topologically controlled 3D mesh editing". In: Computers & Graphics. ISSN: 0097-8493. DOI: 10.1016/j.cag.2022.06.008.
- Ehrig, Hartmut et al. (2006). Fundamentals of Algebraic Graph Transformation. Monographs in Theoretical Computer Science. An EATCS Series. Berlin Heidelberg: Springer-Verlag. ISBN: 978-3-540-31187-4. DOI: 10.1007/3-540-31188-2.
- Habel, Annegret et al. (Apr. 2009). "Correctness of high-level transformation systems relative to nested conditions". In: Mathematical Structures in Computer Science 19.2, pp. 245–296. ISSN: 1469-8072, 0960-1295. DOI: 10.1017/S0960129508007202.

References III

- Heckel, Reiko et al. (2020). Graph Transformation for Software Engineers: With Applications to Model-Based Development and Domain-Specific Language Engineering. Cham: Springer International Publishing. ISBN: 978-3-030-43915-6. DOI: 10.1007/978-3-030-43916-3.
- Pascual, Romain et al. (2022a). "Inferring topological operations on generalized maps: Application to subdivision schemes". In: Graphics and Visual Computing. DOI: 10.1016/j.gvc.2022.200049.
- Pascual, Romain et al. (2022b). "Topological consistency preservation with graph transformation schemes". In: Science of Computer Programming. DOI: 10.1016/j.scico.2021.102728.

References IV

Richaume, Lydie et al. (2019). “Unfolding Level 1 Menger Polycubes of Arbitrary Size With Help of Outer Faces”. In: Discrete Geometry for Computer Imagery. Ed. by Michel Couprie et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 457–468. ISBN: 978-3-030-14085-4. DOI: 10.1007/978-3-030-14085-4_36.

Rozenberg, Grzegorz, ed. (Feb. 1, 1997). Handbook of Graph Grammars and Computing by Graph Transformation: Volume I. Foundations. Vol. Foundations. 1 vols. USA: World Scientific Publishing Co., Inc. 545 pp. ISBN: 978-981-02-2884-2.