Consistent geometric modeling operations
An application of graph transformations

Romain Pascual
romainpascual.fr

November 16, 2023

KIT Seminar
1. Introduction
1. Introduction
1. Introduction

```cpp
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

1. Introduction
Embedded generalized maps

- How to represent objects?
Topological cells

2. Embedded generalized maps
Topological cells

2. Embedded generalized maps
Topological cells

2. Embedded generalized maps
Topological cells

2. Embedded generalized maps
Topological cells
Generalized maps\(^1\) (topology)

Legend: 0, 1, 2

\(^1\)Damiand et al. 2014.

2. Embedded generalized maps
Generalized maps\(^1\) (topology)

Legend: 0, 1, 2

Vertices: orbits \(\langle 1, 2 \rangle\)

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

\(^1\)Damiand et al. 2014.

2. Embedded generalized maps
Generalized maps\(^1\) (topology)

Legend: 0, 1, 2

Vertices: orbits \(\langle 1, 2 \rangle\)

Faces: orbits \(\langle 0, 1 \rangle\)

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

\(^1\)Damiand et al. 2014.

2. Embedded generalized maps
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

2. Embedded generalized maps
Embeddings (geometry)

Legend: 0, 1, 2

2. Embedded generalized maps
Embeddings (geometry)

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

Legend: 0, 1, 2

position : $\langle 1, 2 \rangle \rightarrow \text{Point3}$

color : $\langle 0, 1 \rangle \rightarrow \text{ColorRGB}$

2. Embedded generalized maps
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

2. Embedded generalized maps
Graph rewriting

▶ How to formalize object transformations?
Graph transformation rules\textsuperscript{1}

3. Graph rewriting

\textsuperscript{1}Rozenberg 1997; Ehrig et al. 2006; Heckel et al. 2020.
Rewriting Gmaps

3. Graph rewriting
Orbit rewriting

Implicitly computed

3. Graph rewriting
Orbit rewriting

3. Graph rewriting
Orbit rewriting

3. Graph rewriting
Orbit rewriting

3. Graph rewriting
Orbit rewriting

Implicitly computed

Local

Instantiated rule

3. Graph rewriting
Orbit rewriting

Implicitly computed

Local

Instantiated rule

3. Graph rewriting
Modifying geometric values

Algebraic data types (attributes)

\(^1\)Bellet et al. 2017.
Modifying geometric values

Algebraic data types, three kinds of expressions

- Accessors
  - a.color = 1
  - a.position = A

1Bellet et al. 2017.

3. Graph rewriting
Modifying geometric values

Algebraic data types, three kinds of expressions

- Accessors
- Computations

\[ \text{center}(\{\circ, \bullet\}) = \bullet \]


3. Graph rewriting
Modifying geometric values

Algebraic data types, three kinds of expressions

- Accessors
- Computations
- Topological operators

\[ a@0.\text{position} = D \]
\[ \text{position}_{\langle 0,1 \rangle}(a) = \{ A, B, C, D \} \]

---

\(^1\) Bellet et al. 2017.
Extension to schemes

3. Graph rewriting
Extension to schemes

3. Graph rewriting
Extension to schemes

3. Graph rewriting
Extension to schemes

\[
\frac{1}{4}(A + B + C + D)
\]
Extension to schemes

\[ \frac{1}{4}(A + B + C + D) = \text{middle}(\{A, B, C, D\}) \]
Extension to schemes

\[ \frac{1}{4} (A + B + C + D) = \text{middle}(\{A, B, C, D\}) = \text{middle}(\text{position}_{(0,1)}(a)) \]
Extension to schemes

\[ \frac{1}{4} (A + B + C + D) = \text{middle}(\{A, B, C, D\}) = \text{middle}(\text{position}_{0,1}(a)) = \text{middle}(\text{position}_{0,1}(n0)) \]

3. Graph rewriting
A rule-based language

Topology: categorical semantics for operations
Geometry: structure-based algebraic formalisation

Formalizing the DSL of Jerboa

3. Graph rewriting
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

4. Consistent modeling operations
Breaking the topological consistency

Constraint: 0202-paths should be cycles

4. Consistent modeling operations
Breaking the topological consistency

**Constraint:** 0202-paths should be cycles

4. Consistent modeling operations
Breaking the geometric consistency

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

$$\text{mix}(a.\text{color}, b.\text{color})$$
Breaking the geometric consistency

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

```
mix(a.color, b.color)
```

4. Consistent modeling operations
Breaking the geometric consistency

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

\[
\text{mix}(a.\text{color}, b.\text{color})
\]

Rule completion

4. Consistent modeling operations
4. Consistent modeling operations
4. Consistent modeling operations
4. Consistent modeling operations
4. Consistent modeling operations
4. Consistent modeling operations
Inference of modeling operations

Theorem

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

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

Color legend: 0, 1, 2, $\kappa$.

4. Consistent modeling operations
Main contributions

Topological consistency: path analysis on rule schemes

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

5. Conclusion
Current research projects

Following up on the formalization of Jerboa

- Multi-cell query-replace approach for combinatorial maps\(^1\)
  Guillaume Damiant, Vincent Nivoliers and Jordan Goncalves (M2 intern)

- Towards a local calculus for nested conditions?\(^2\)
  Nicolas Behr and Pascale Le Gall

---

\(^1\)Damiant et al. 2022.
\(^2\)Habel et al. 2009.
5. Conclusion
(2, 2, 2)-Menger polycube\textsuperscript{1}

\textsuperscript{1}Richaume et al. 2019.

5. Conclusion
(2, 2, 2)-Menger polycube\textsuperscript{1}

\textsuperscript{1}Richaume et al. 2019.

5. Conclusion
(2, 2, 2)-Menger polycube

5. Conclusion

1Richaume 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.


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.


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


