

Graph transformation for geometric modeling

Romain Pascual

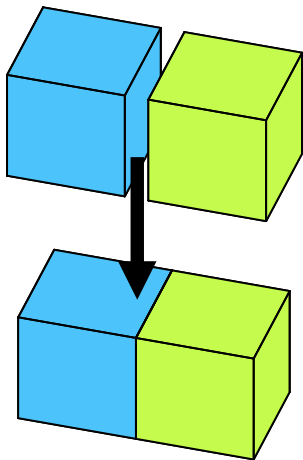
December 14, 2023

Xmas MICS





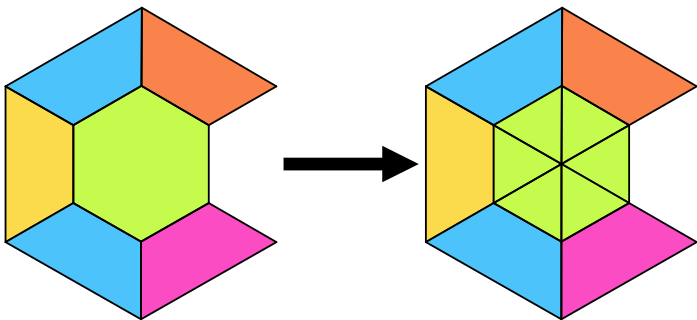




CGAL's sew operation

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

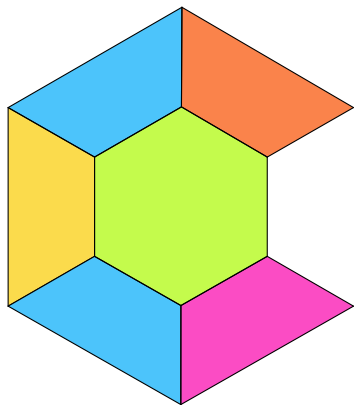
{
    CGAL_assertion( i<=dimension );
    CGAL_assertion( (is_sewable<i>(adart1,adart2)) );
    size_type amark=get_new_mark();
    CGAL::GMap_dart_iterator_basic_of_involution<Self, i>
        I1(*this, adart1, amark);
    CGAL::GMap_dart_iterator_basic_of_involution<Self, i>
        I2(*this, adart2, 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(amarck) );
    free_mark(amarck);
}
```



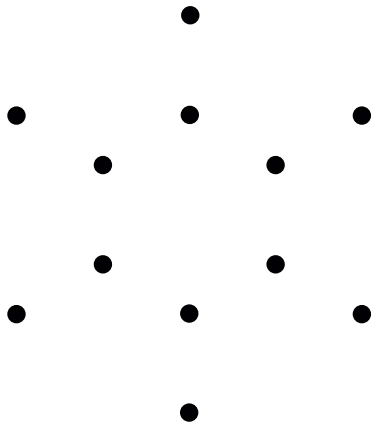
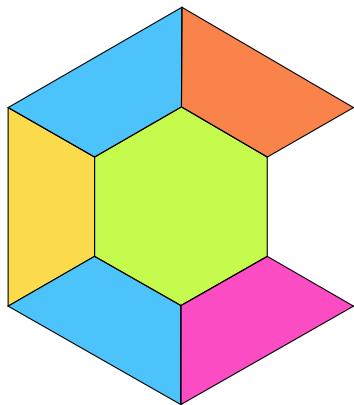
Ambition : define a *domain-specific language* (DSL) for geometric modeling

Motivations : abstraction, performance, conciseness, correctness, simplicity

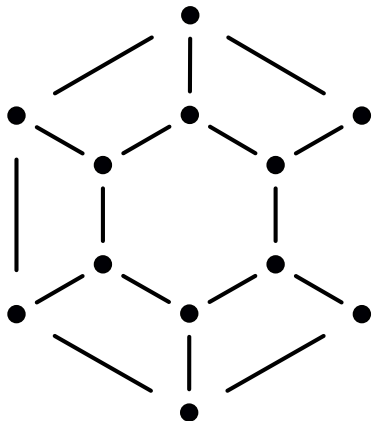
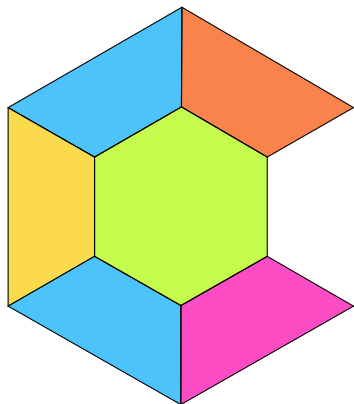
Cellular decomposition



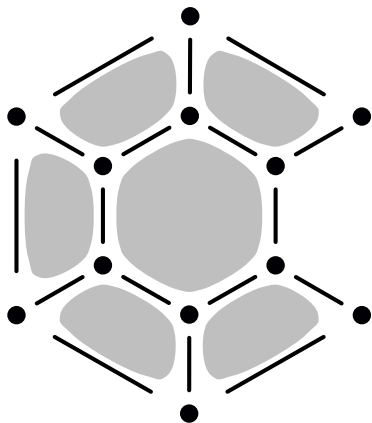
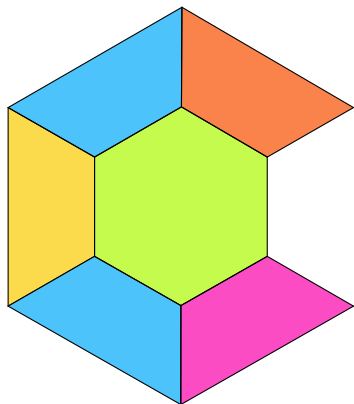
Cellular decomposition



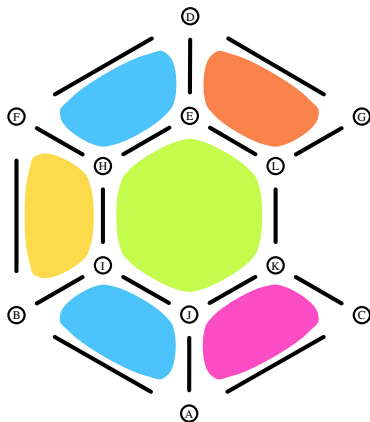
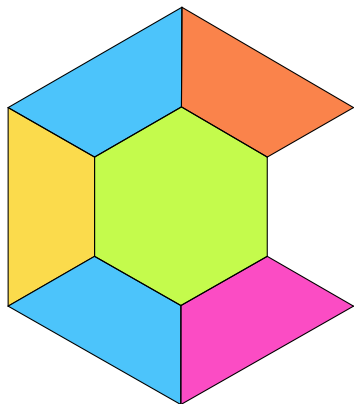
Cellular decomposition



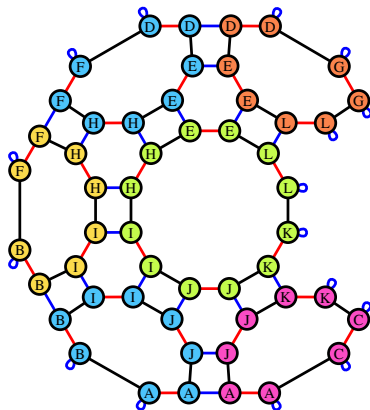
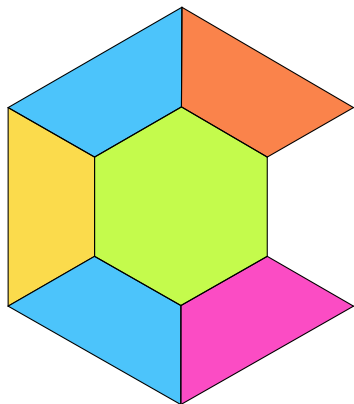
Cellular decomposition



Cellular decomposition



Cellular decomposition: embedded generalized maps



A graph ?

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



A graph ?

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

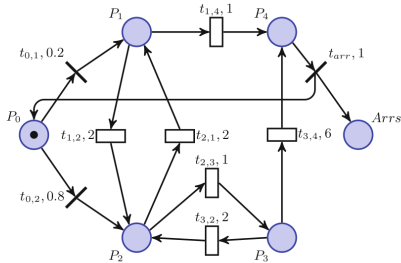


Fig. 12. eGSPN encoding of PH arrival model with PH specified in (1) in Sect. 3.1

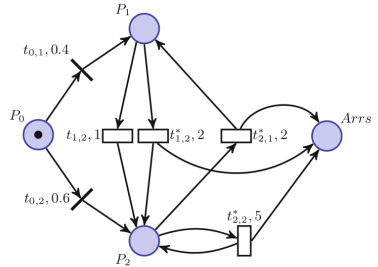
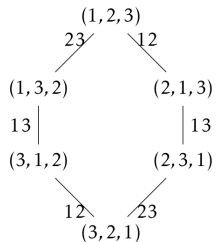


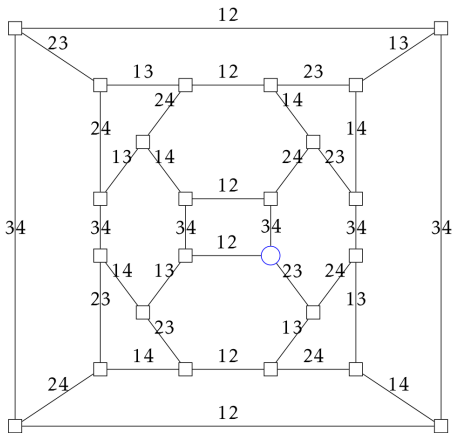
Fig. 13. eGSPN encoding of MAP arrival model with MAP given in (2) in Sect. 3.2

A graph ?

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



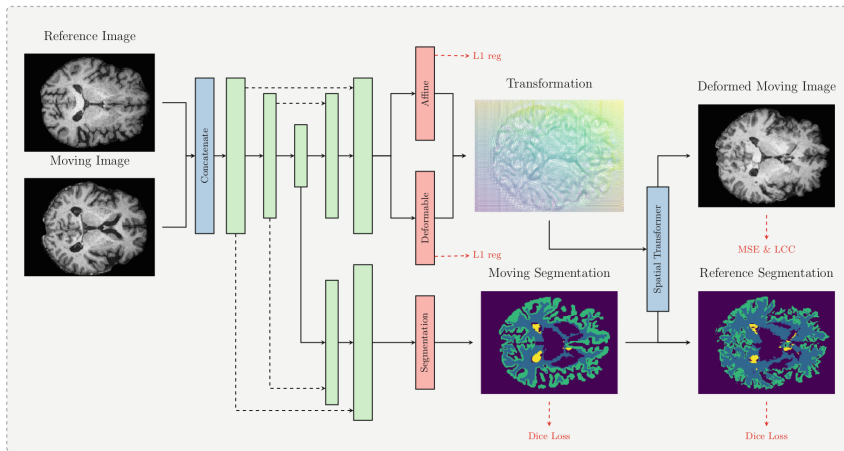
(a) permutohedron of order 3



(b) permutohedron of order 4

A graph ?

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



A graph ?

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

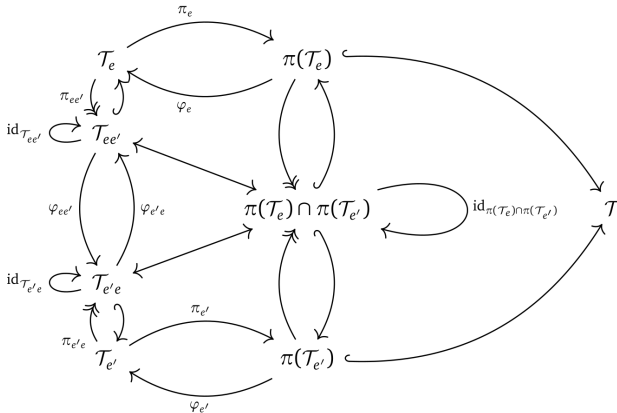
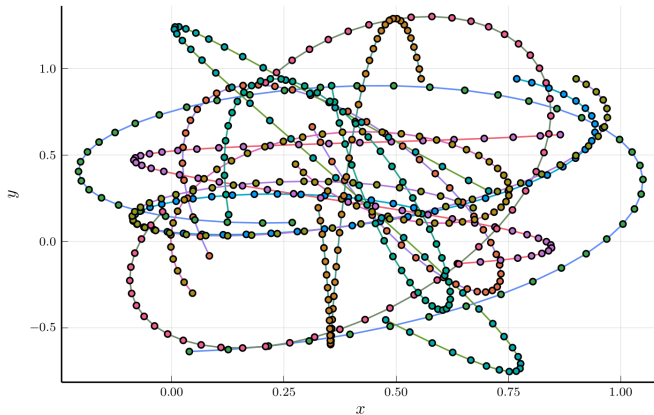


Figure 1.3: Diagram illustrating the parallel gluing from Proposition 1.2.26.

A graph ?

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

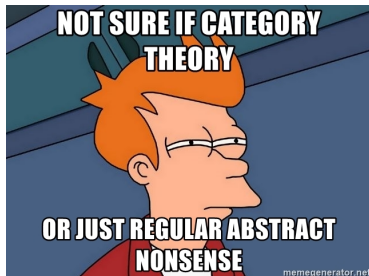


(a) $\sigma = 0$

A graph ?

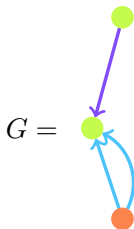
A graph is a presheaf on

$$E \begin{array}{c} \xrightarrow{s} \\ \xrightarrow{t} \end{array} V$$



The category **Graph**

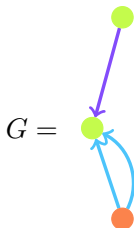
A graph



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

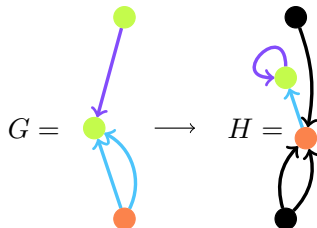
The category **Graph**

A graph



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

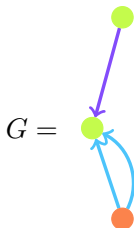
A graph morphism



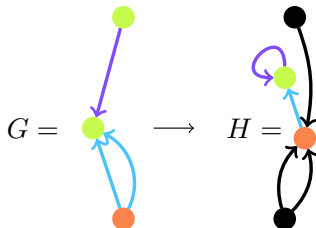
- a node function $V_G \rightarrow V_H$,
 - an arc function $E_G \rightarrow E_H$,
- preserving structure

The category **Graph**

A graph



A graph morphism



Form an **adhesive** category

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

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^*)

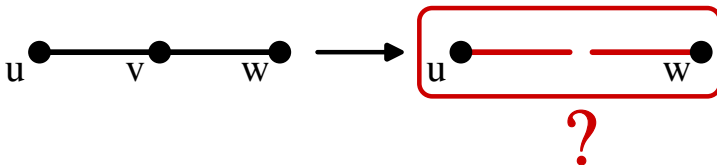
sleep $\rightarrow$ follow

You are all sleeping! $\rightarrow$ You are all following!

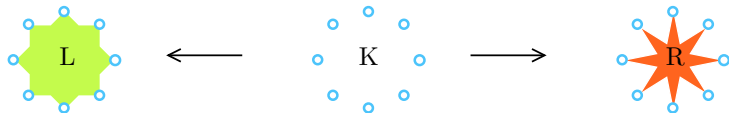
On the difficulty of rewriting graphs



On the difficulty of rewriting graphs

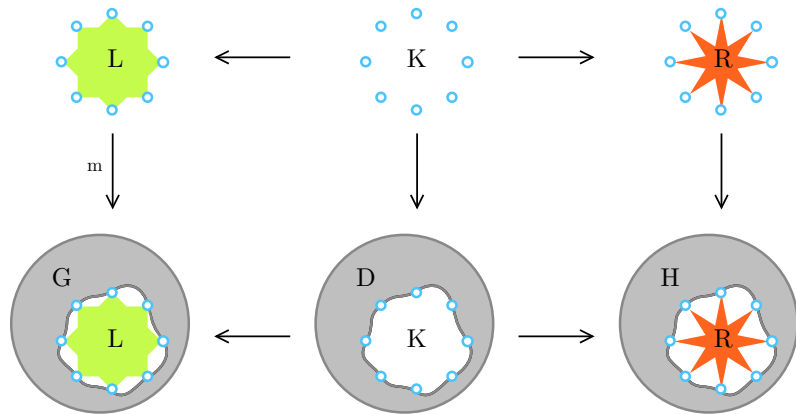


Graph transformation rules¹



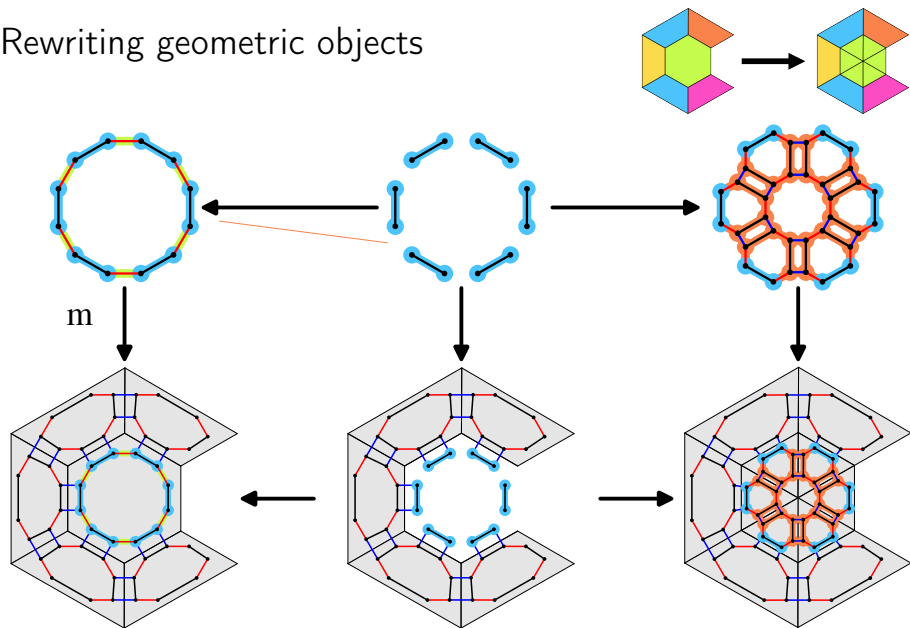
¹Rozenberg 1997; Ehrig et al. 2006; Heckel et al. 2020.

Graph transformation rules¹

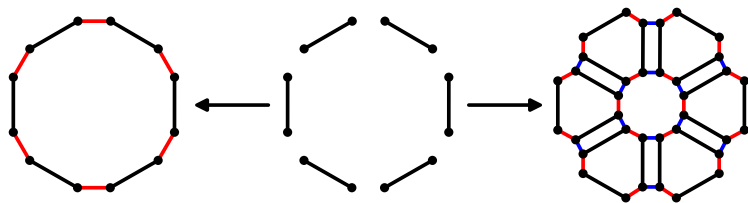
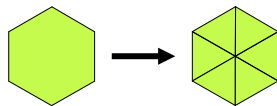


¹Rozenberg 1997; Ehrig et al. 2006; Heckel et al. 2020.

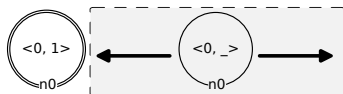
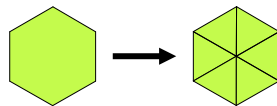
Rewriting geometric objects



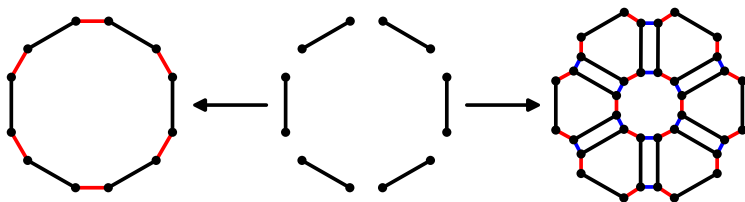
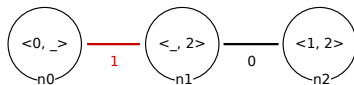
Orbit rewriting



Orbit rewriting

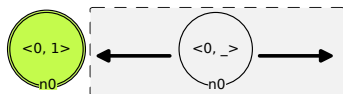
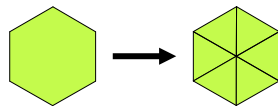


Implicitly
computed



Instantiated
rule

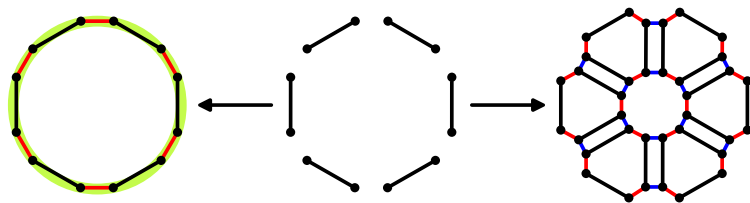
Orbit rewriting



Implicitly
computed

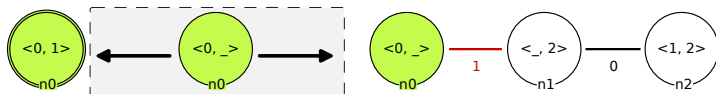
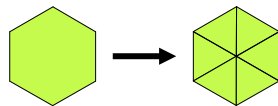


Local



Instantiated
rule

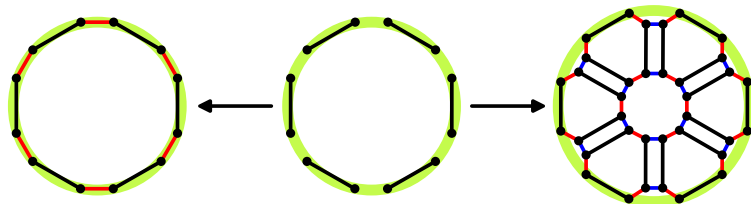
Orbit rewriting



Implicitly
computed

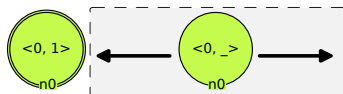
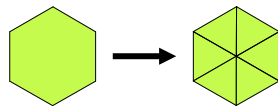


Local

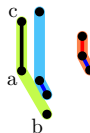


Instantiated
rule

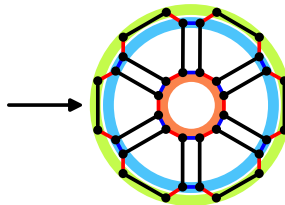
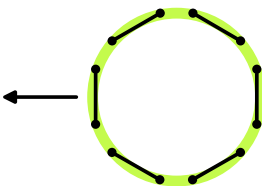
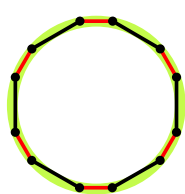
Orbit rewriting



Implicitly
computed

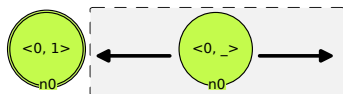
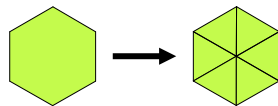


Local

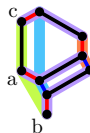
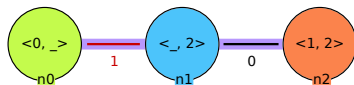


Instantiated
rule

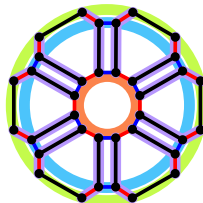
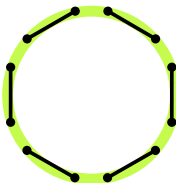
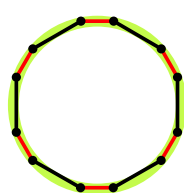
Orbit rewriting



Implicitly
computed

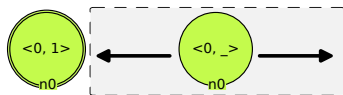
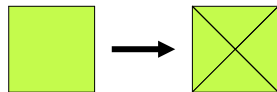


Local

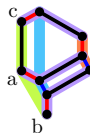
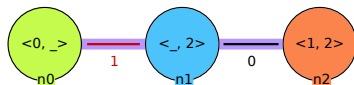


Instantiated
rule

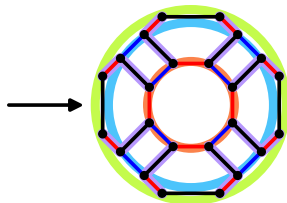
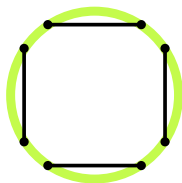
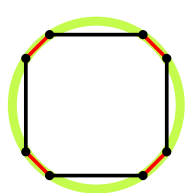
Orbit rewriting



Implicitly
computed

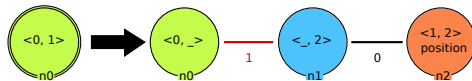


Local



Instantiated
rule

Some results

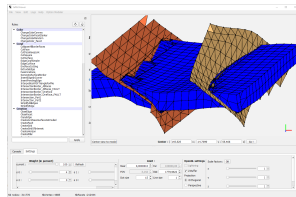
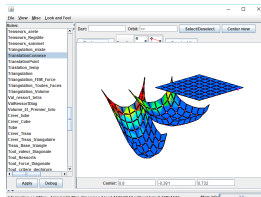
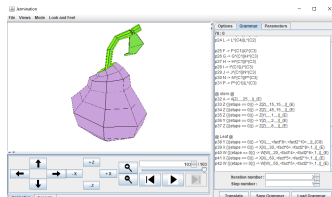


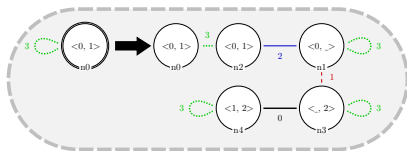
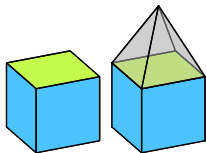
Topology : categorical semantics for operations

Geometry : structure-based algebraic formalisation

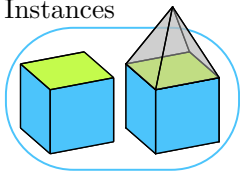
Static analysis : preservation of the model's constraints

Formalizing the DSL of  *Jerkboa*

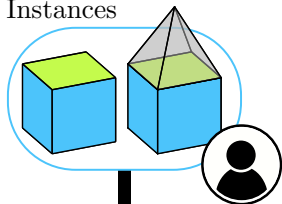




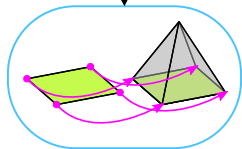
Instances



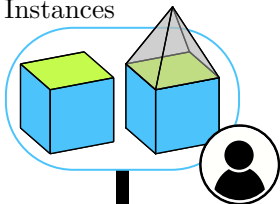
Instances



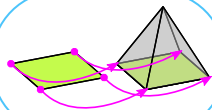
Mapping



Instances

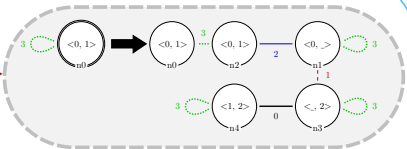


Mapping

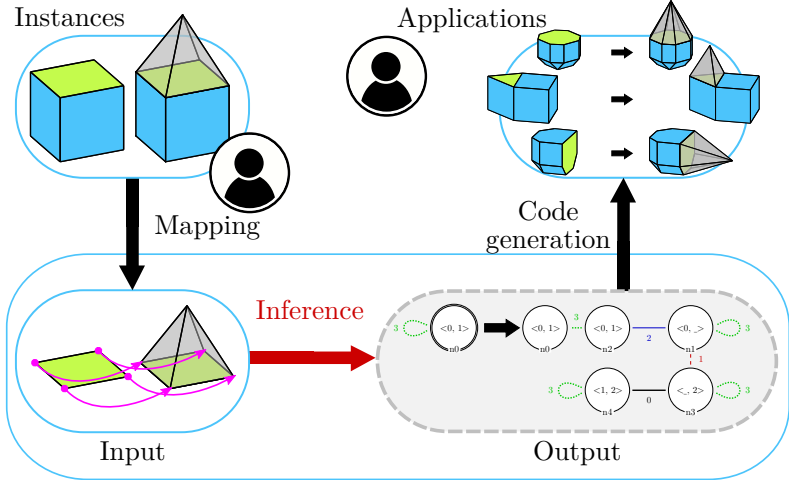


Input

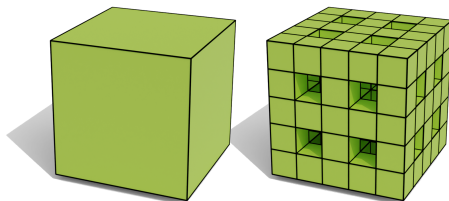
Inference



Output

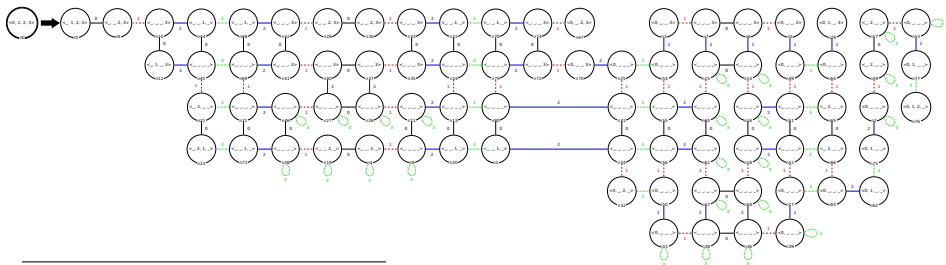
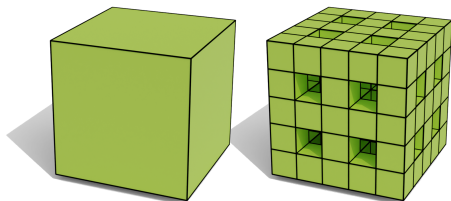


$(2, 2, 2)$ -Menger polycube¹



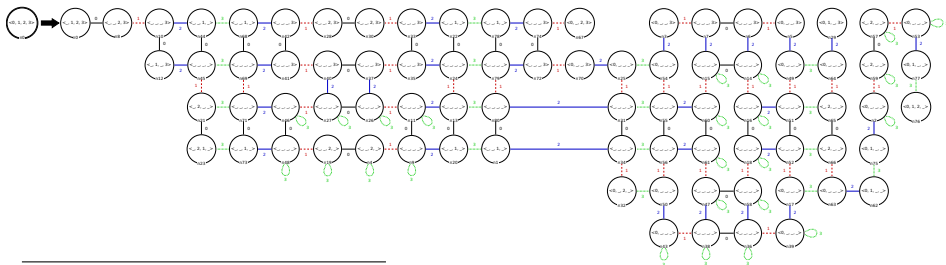
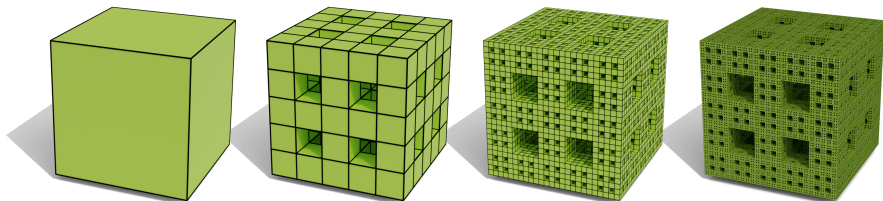
¹Richaume et al. 2019.

$(2, 2, 2)$ -Menger polycube¹



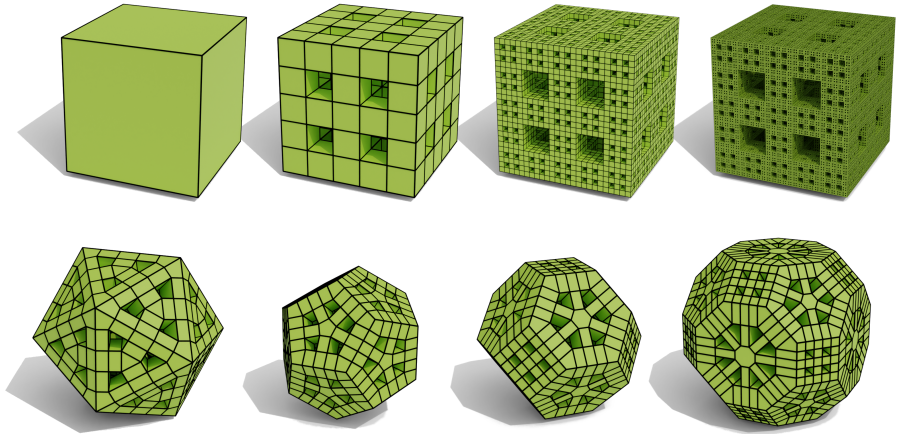
¹Richaume et al. 2019.

$(2, 2, 2)$ -Menger polycube¹



¹Richaume et al. 2019.

$(2, 2, 2)$ -Menger polycube¹



¹Richaume et al. 2019.

Who's who?

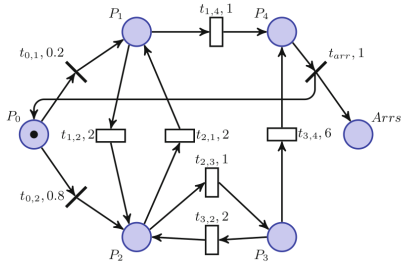


Fig. 12. eGSPN encoding of PH arrival model with PH specified in (1) in Sect. 3.1

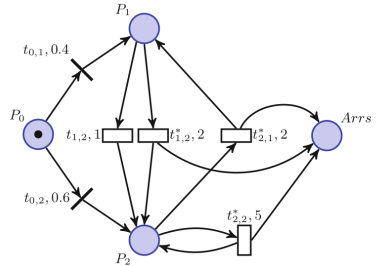
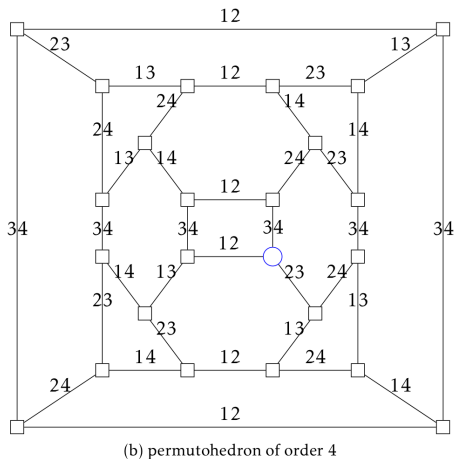
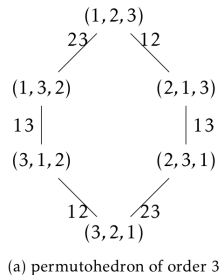
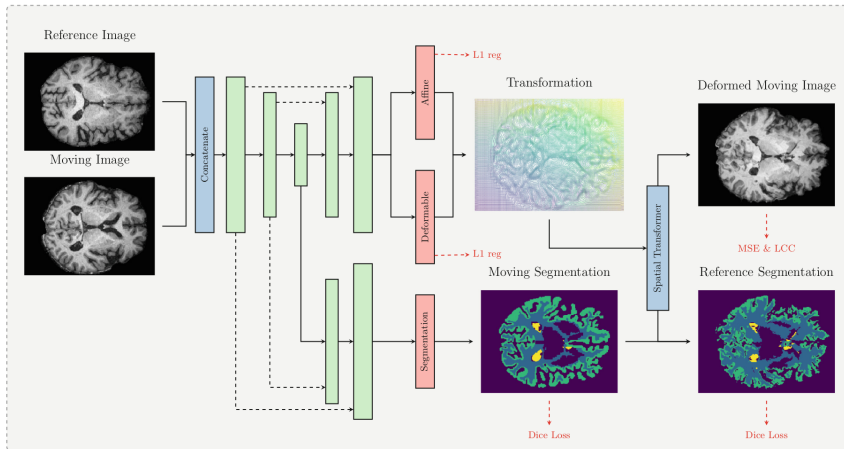


Fig. 13. eGSPN encoding of MAP arrival model with MAP given in (2) in Sect. 3.2

Who's who?



Who's who?



Who's who?

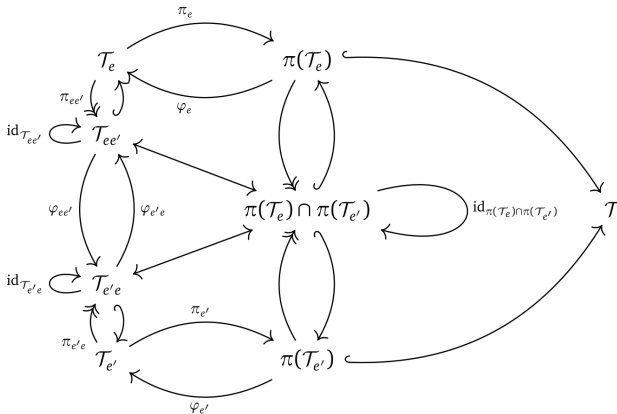
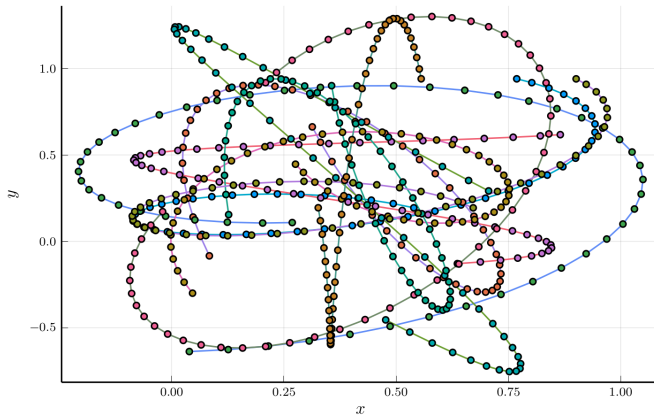


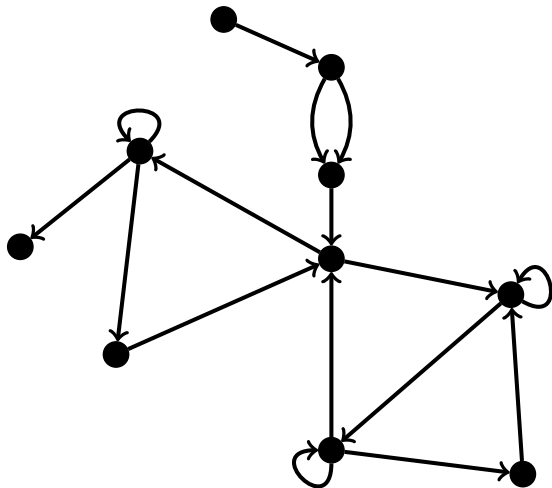
Figure 1.3: Diagram illustrating the parallel gluing from Proposition 1.2.26.

Who's who?



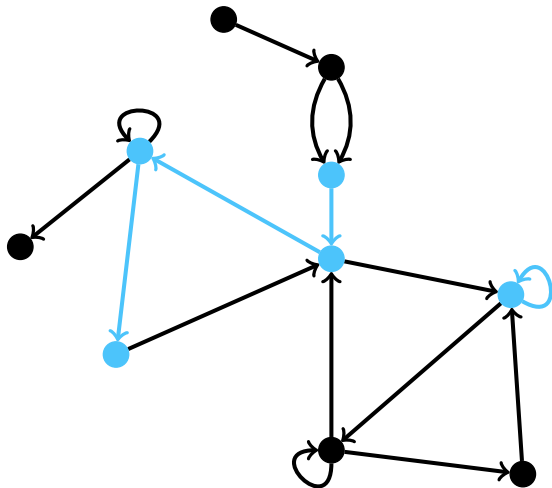
(a) $\sigma = 0$

The subgraph logic is intuitionistic



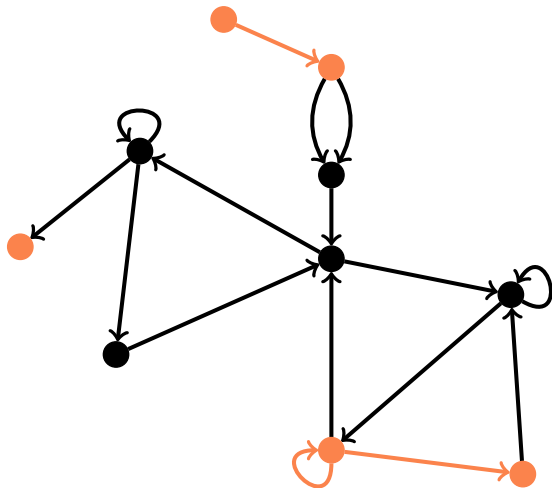
G

The subgraph logic is intuitionistic



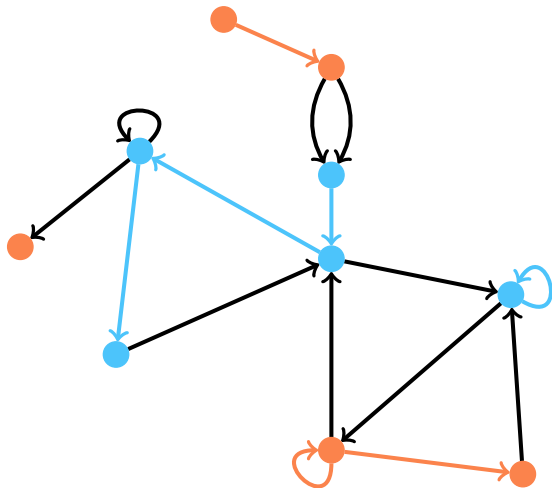
$(A)_G$

The subgraph logic is intuitionistic



$$(\neg A)_G$$

The subgraph logic is intuitionistic



$$(\neg A \vee A)_G$$

References I



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.



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.

References II



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.