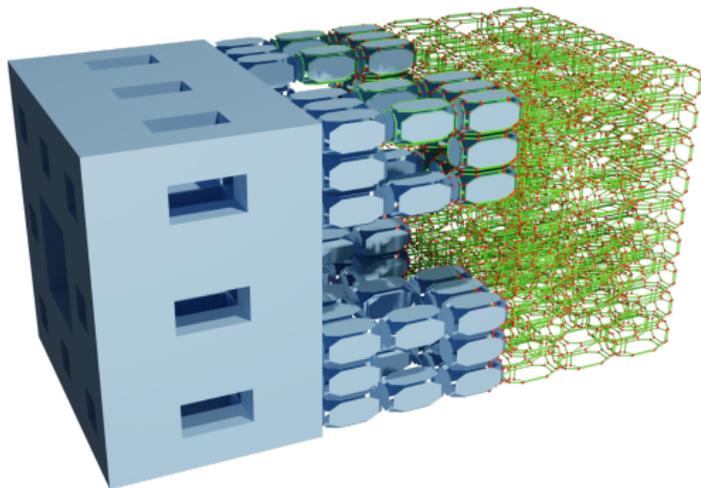


Déduction topologique et géométrique pour l'inférence d'opérations de modélisation



Romain Pascual, Hakim
Belhaouari, Agnès Arnould,
Pascale Le Gall

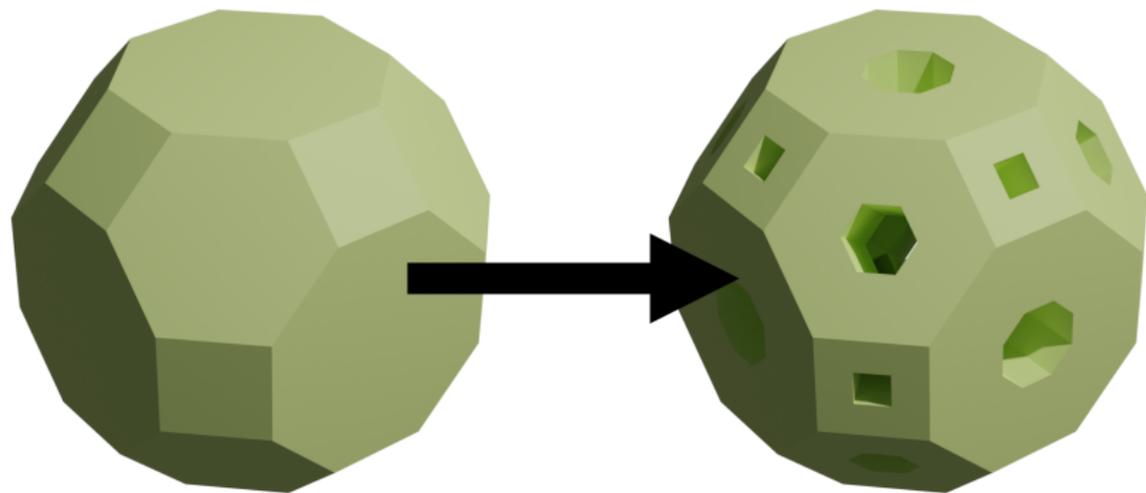
16-17 mars 2022

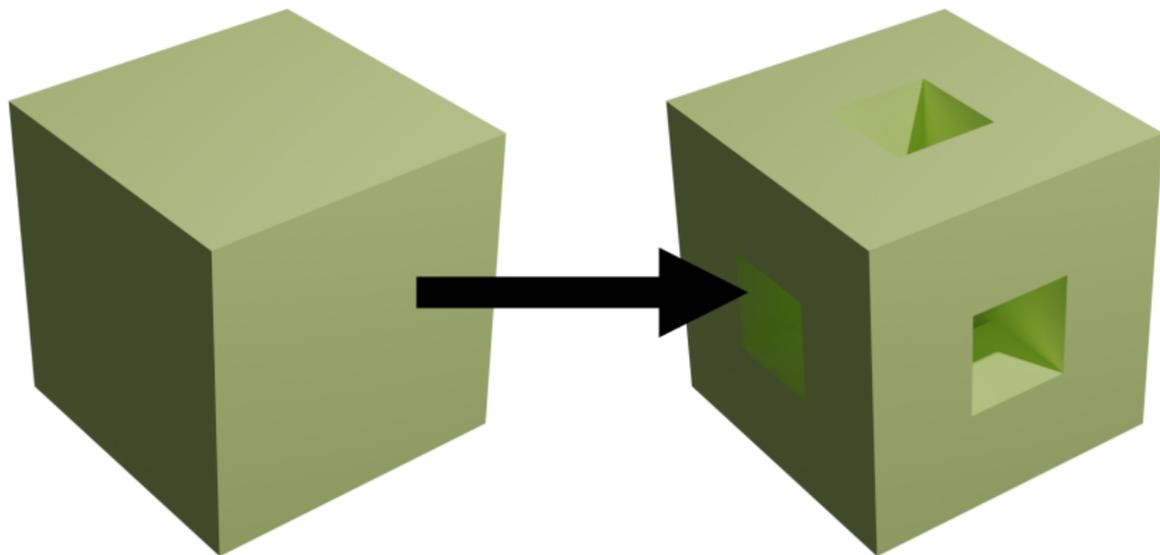


université
PARIS-SACLAY

xl
lim

UNIVERSITÉ
de Poitiers





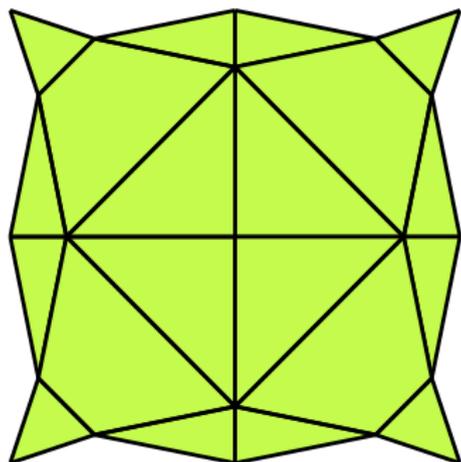
- 2 Inférence topologique
 - Cartes généralisées
 - Réécriture de graphes
 - Algorithme de repliement topologique

- 3 Inférence géométrique
 - Modèle plongé
 - Expressions de plongements
 - Résolution par programmation par contraintes

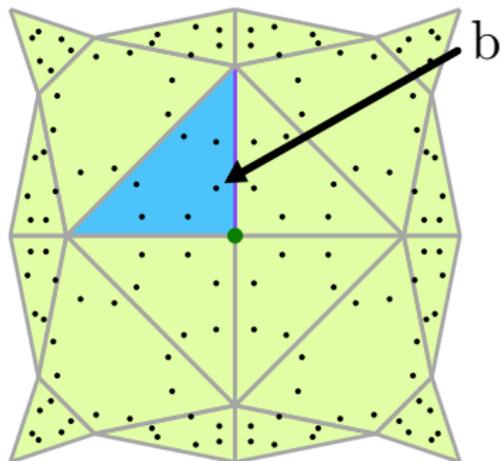
Inférence topologique

- ▶ Algorithme de parcours de graphe pour l'inférence de règles appliquées sur des cartes généralisées.

Cartes généralisées [Damiand et Lienhardt 2004]



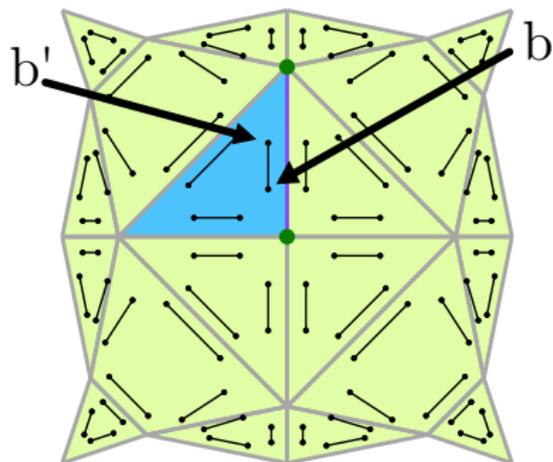
Cartes généralisées [Damiand et Lienhardt 2004]



Construction sous forme de graphe des G-cartes.

b identifie le sommet vert, l'arête violette et la face bleue.

Cartes généralisées [Damiand et Lienhardt 2004]

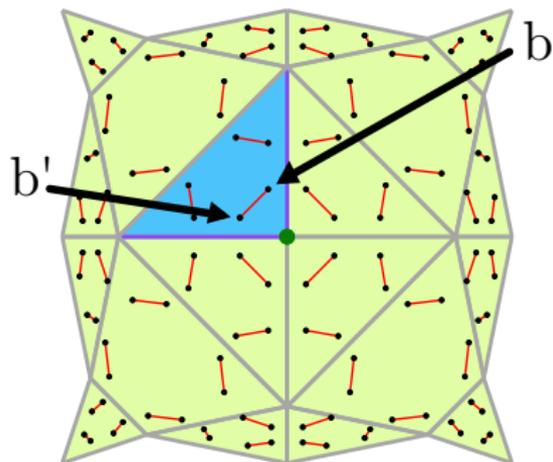


Construction sous forme de graphe des G-cartes.

0-liens :

- différentes sommets.
- mêmes arêtes et faces.

Cartes généralisées [Damiand et Lienhardt 2004]

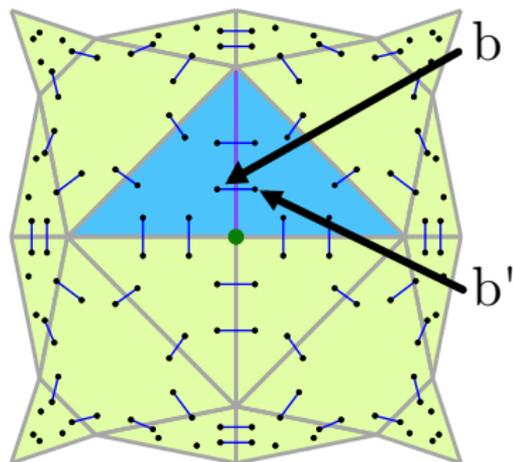


Construction sous forme de graphe des G-cartes.

1-liens :

- différentes arêtes.
- mêmes sommets et faces.

Cartes généralisées [Damiand et Lienhardt 2004]

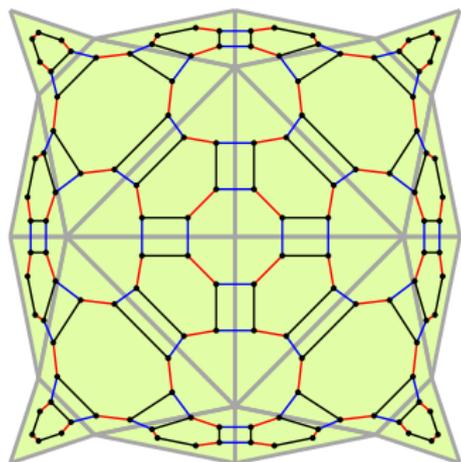


Construction sous forme de graphe des G-cartes.

2-liens :

- différents faces.
- mêmes sommets et arêtes.

Cartes généralisées [Damiand et Lienhardt 2004]

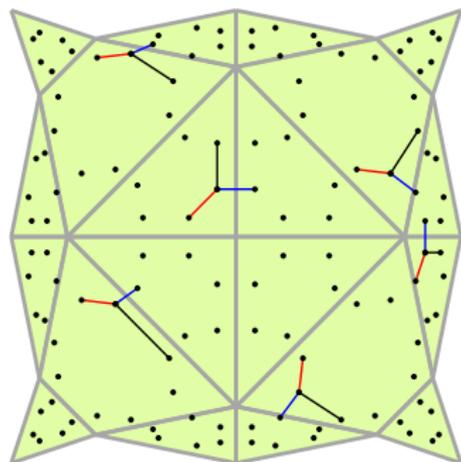


Construction sous forme de
graphe des G-cartes.

n-G-carte (union des graphes)

Graphe non-orienté étiqueté sur
les arcs sur $\llbracket 0, n \rrbracket$ tel que :

Cartes généralisées [Damiand et Lienhardt 2004]



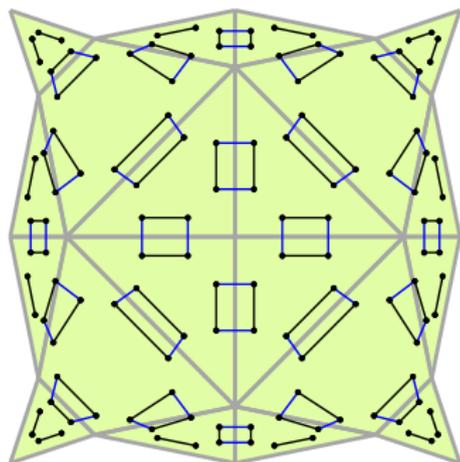
Construction sous forme de graphe des G-cartes.

n-G-carte (union des graphes)

Graphe non-orienté étiqueté sur les arcs sur $\llbracket 0, n \rrbracket$ tel que :

- **Incidence** : chaque brin est la source d'un unique arc par dimension

Cartes généralisées [Damiand et Lienhardt 2004]



Construction sous forme de graphe des G-cartes.

n-G-carte (union des graphes)

Graphe non-orienté étiqueté sur les arcs sur $\llbracket 0, n \rrbracket$ tel que :

- **Incidence** : chaque brin est la source d'un unique arc par dimension
- **Cycle** : tout $ijij$ -chemin est un cycle si $i + 2 \leq j$

Réécriture

$$n = 4$$

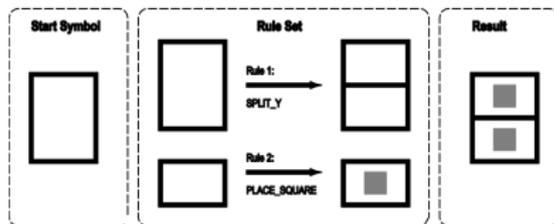
$$\alpha = 25^\circ$$

$$V = \{S, F\}$$

$$T = \{+, -, [,]\}$$

$$P = \{ F \rightarrow F [+ F] F [- F] \}$$

$$S = F$$



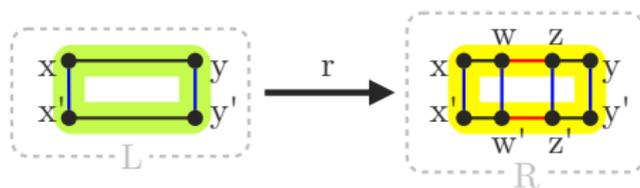
L-systèmes

([Santos et Coelho 2009])

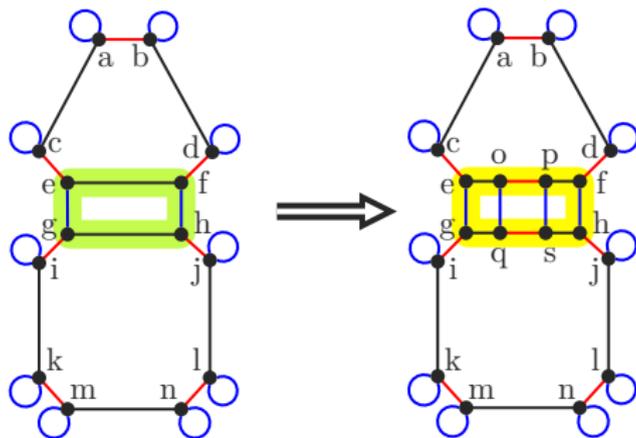
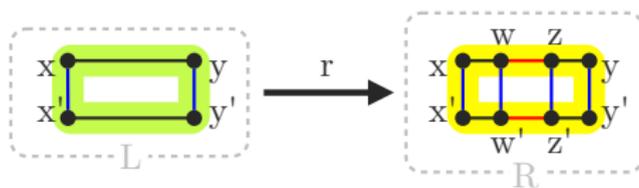
Grammaires de formes

([Di Angelo et al. 2012])

Réécriture de G-carte : insertion d'un sommet



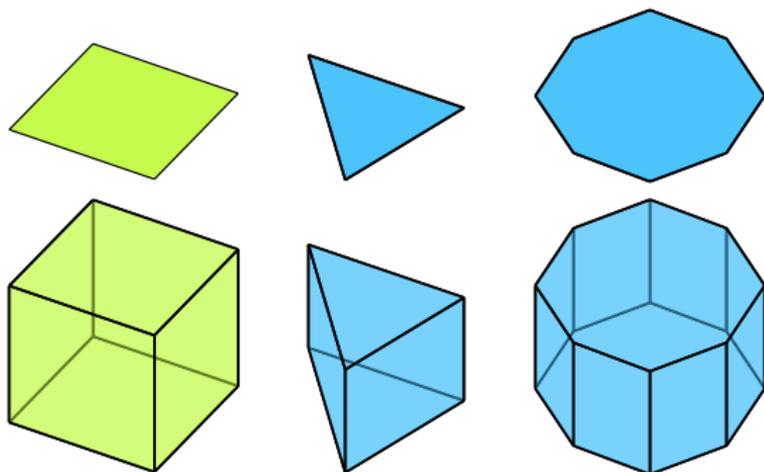
Réécriture de G-carte : insertion d'un sommet



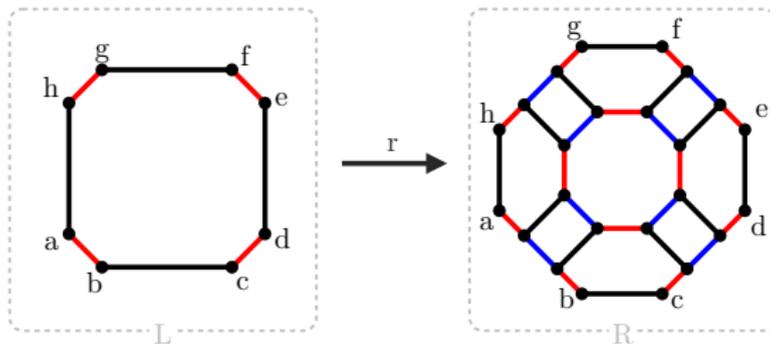
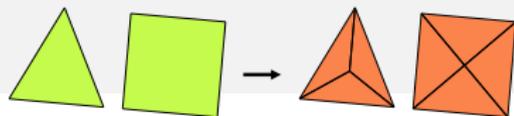
Enjeu de généricité

Exploiter la structure des G-cartes.

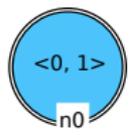
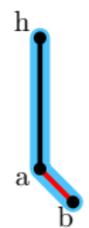
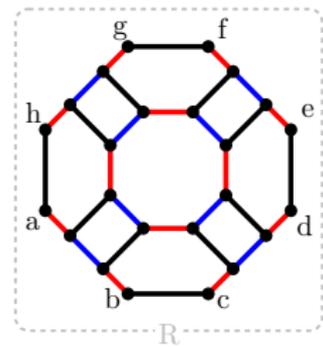
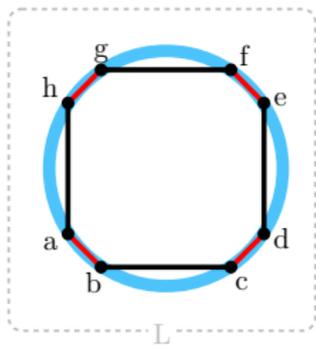
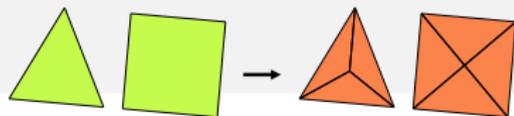
- ▶ Les opérations de modélisation géométrique sont paramétrées par un type de cellules topologiques.



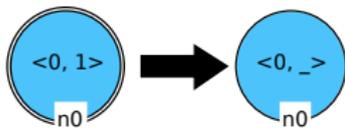
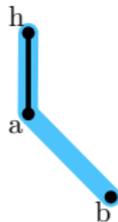
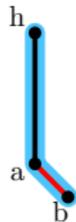
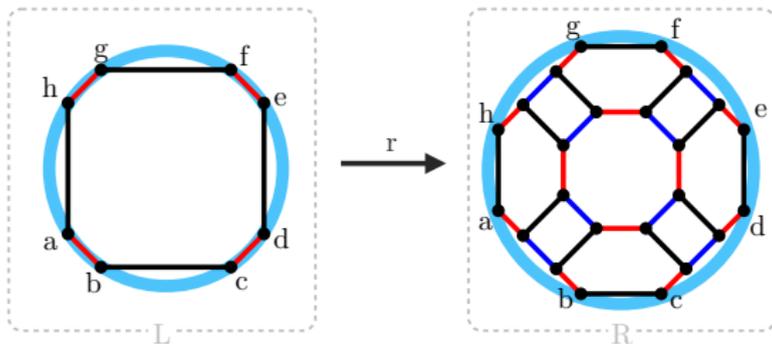
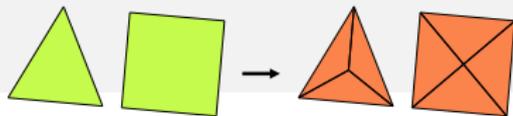
Analyse des symétries



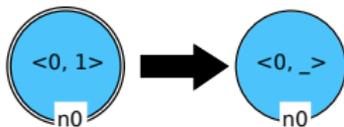
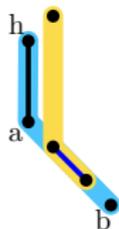
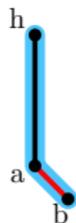
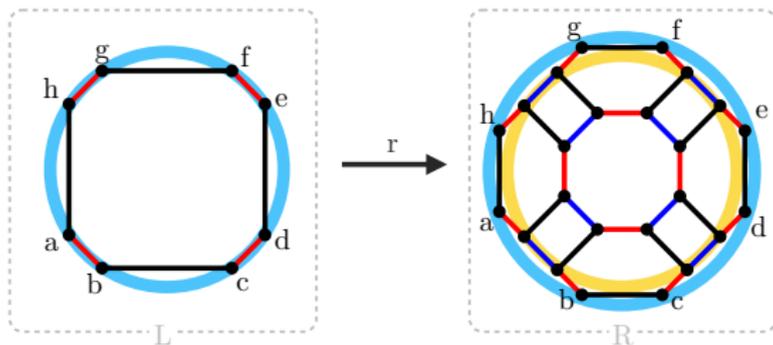
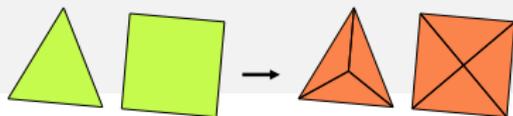
Analyse des symétries



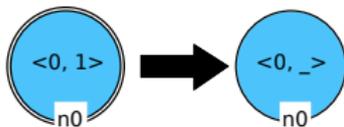
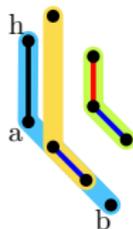
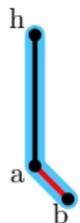
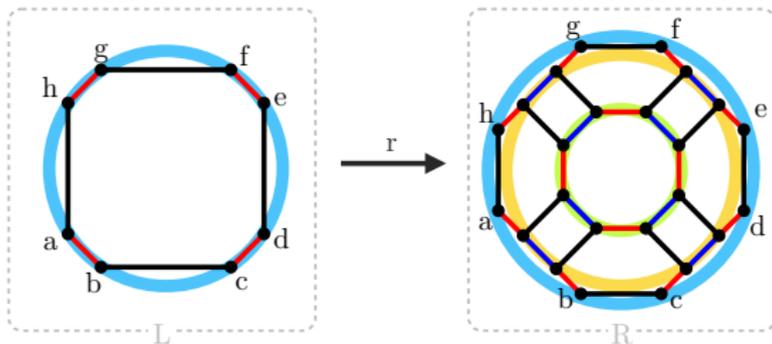
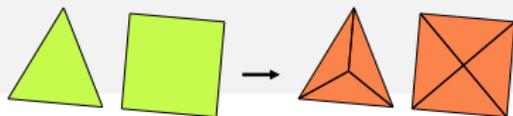
Analyse des symétries



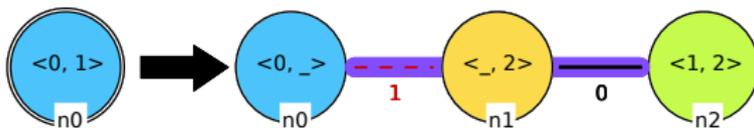
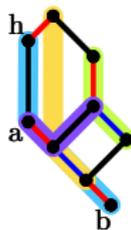
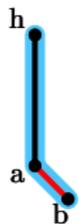
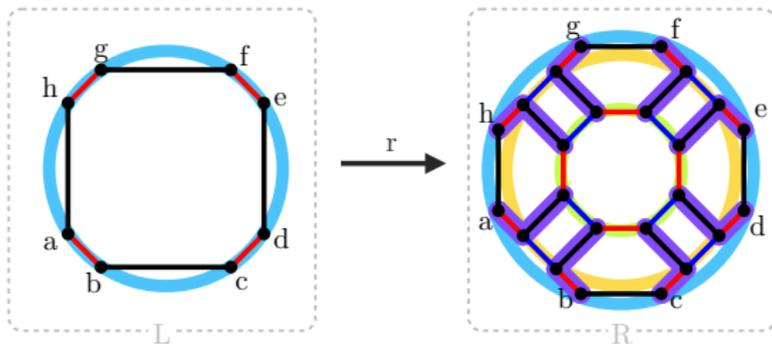
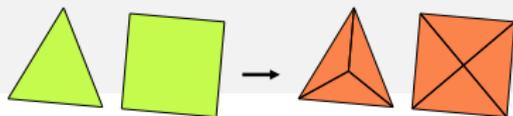
Analyse des symétries



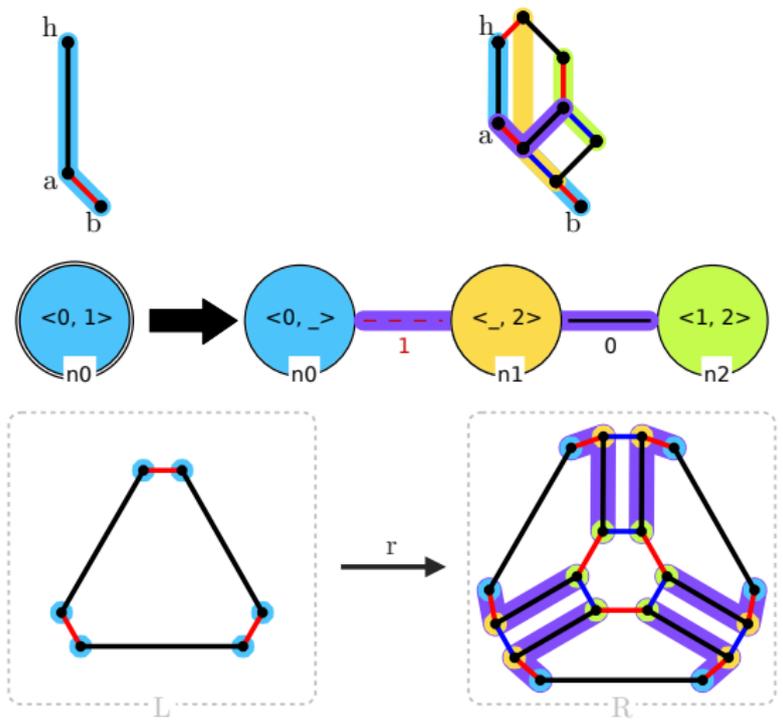
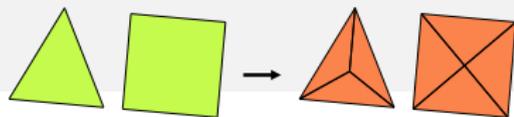
Analyse des symétries



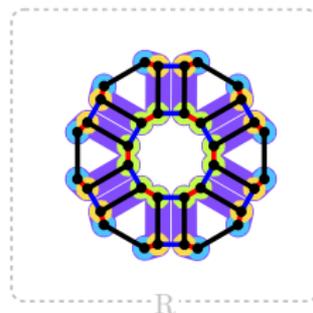
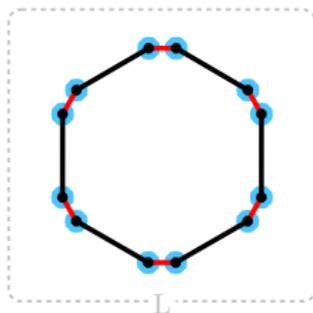
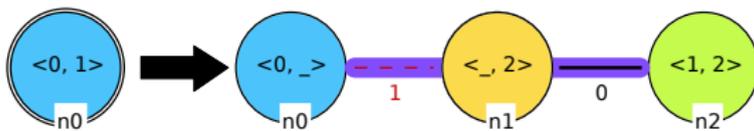
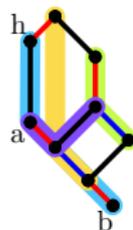
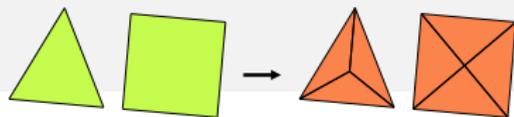
Analyse des symétries



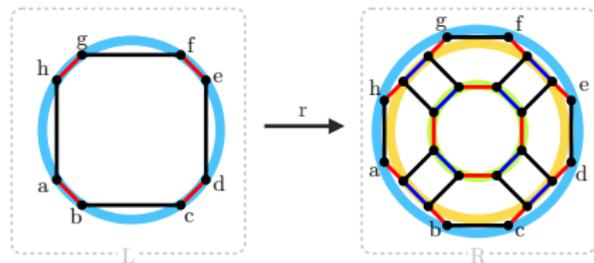
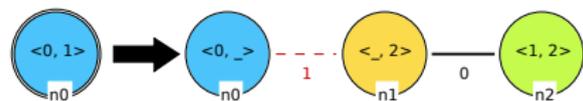
Analyse des symétries



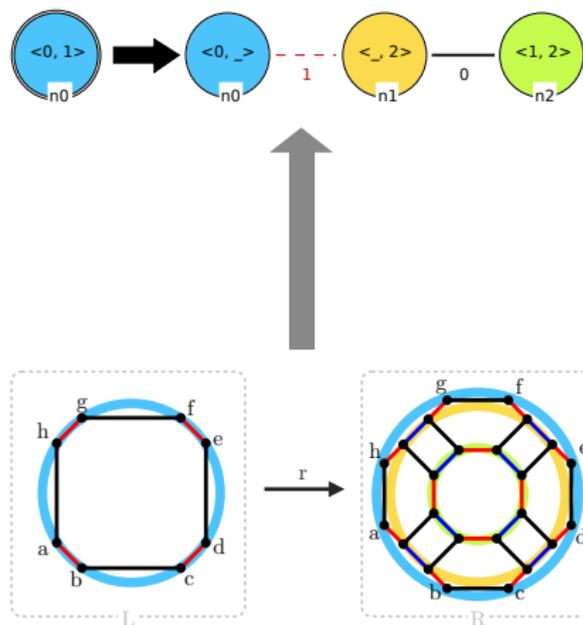
Analyse des symétries



Instanciation et inférence (GTMG 2021)



Instanciation et inférence (GTMG 2021)



Algorithm : Topological folding algorithm

Input: A graph G encoding the preservation relation between two partial G-maps, an orbit type $\langle o \rangle$, and a dart a of G .

Output: A graph S that encodes the Jerboa rule with $\langle o \rangle$ as variable, given that the operation is applied at the dart a .

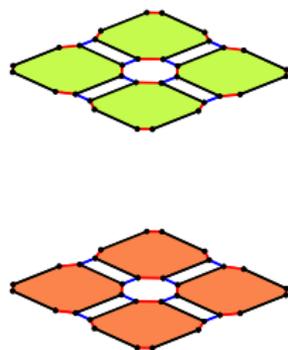
```

1  $Q \leftarrow \emptyset, S \leftarrow \emptyset$  // empty queue and empty 'rule' graph
2  $h \leftarrow \text{Node}(G, \langle o \rangle, a)$  // build the hook node
3  $\text{add\_node}(S, h)$  // add h to the 'rule' graph
4  $\text{enqueue}(Q, h)$ 
5 while  $Q \neq \emptyset$  do
6    $m \leftarrow \text{dequeue}(Q)$ 
7   foreach  $d \in [0, n] \setminus \text{label}(m)$  do
8      $v \leftarrow \text{arc\_expansion}(G, m, d)$  // extend arcs
9      $\text{buildLabel}(G, v)$  // deduce the relabeling function
10     $\text{add\_node}(S, v)$ 
11     $\text{enqueue}(Q, v)$ 
12 return  $S$ 

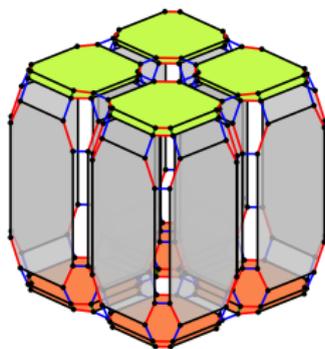
```

Exemple en géologie (purement topologique)

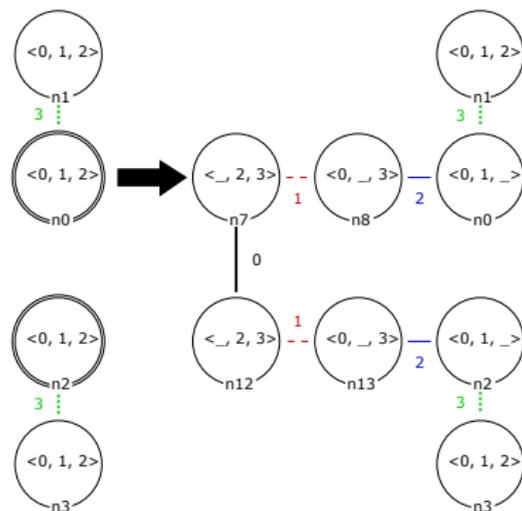
Avant



Après



Operation

Temps d'inférence : ~ 3 ms

Exemple avec Jerboa (purement topologique)

Avant



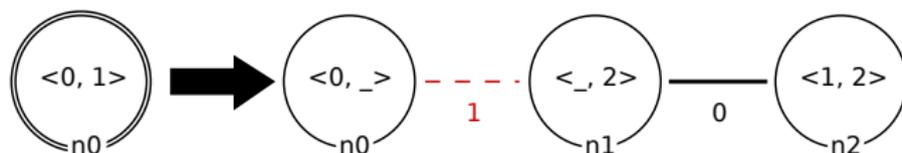
Exemple avec Jerboa (purement topologique)

Après



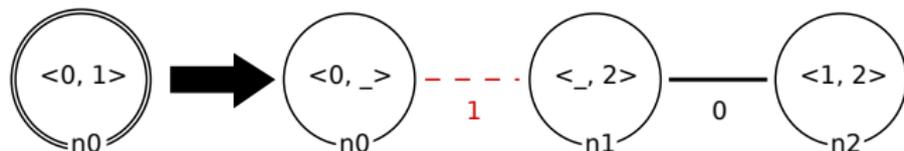
Points clefs

Les opérations de modélisation géométriques sont décrites par des schémas de règles.



Points clefs

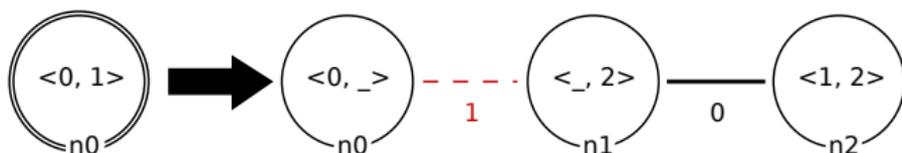
Les opérations de modélisation géométriques sont décrites par des schémas de règles.



- Description des opérations via un langage dédié.

Points clefs

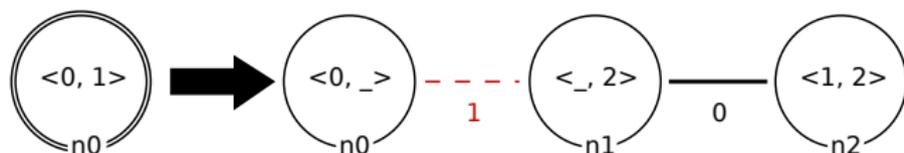
Les opérations de modélisation géométriques sont décrites par des schémas de règles.



- Description des opérations via un langage dédié.
- Motifs gauche et droit de la règle sont des graphes.

Points clefs

Les opérations de modélisation géométriques sont décrites par des schémas de règles.



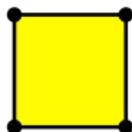
- Description des opérations via un langage dédié.
- Motifs gauche et droit de la règle sont des graphes.
- Chaque noeud du graphe représente un ensemble de brins dans la carte généralisée.

Inférence géométrique

- ▶ Programmation par contraintes pour l'inférence de calculs de plongements.

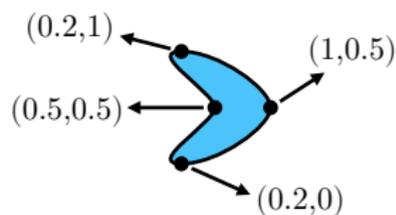
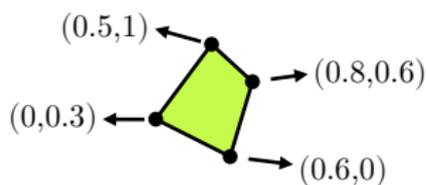
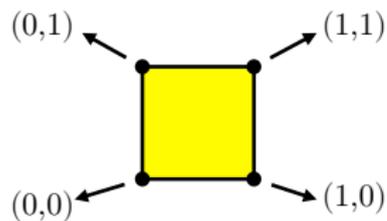
Comment afficher les objets ?

- ▶ Différentes géométries pour une même topologie.



Comment afficher les objets ?

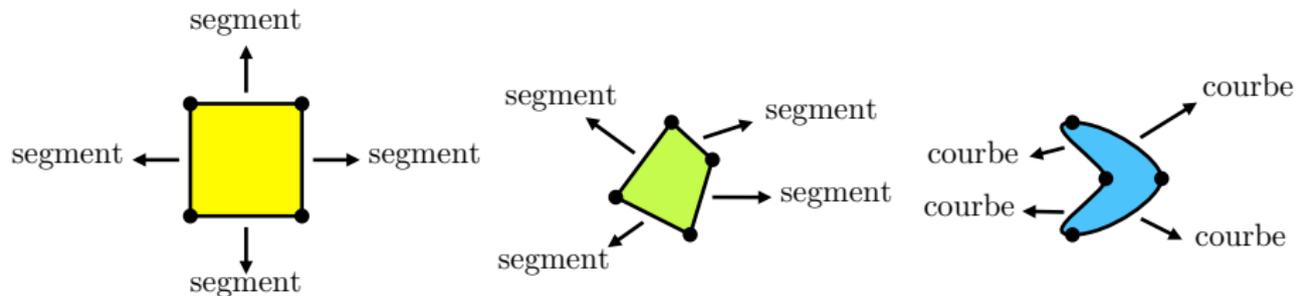
- Différentes géométries pour une même topologie.



Plongement sur les cellules topologiques : position.

Comment afficher les objets ?

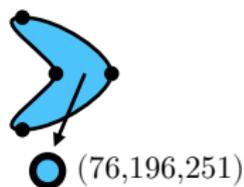
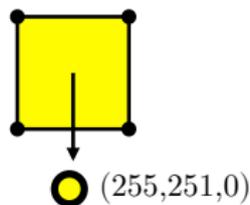
- Différentes géométries pour une même topologie.



Plongement sur les cellules topologiques : courbure.

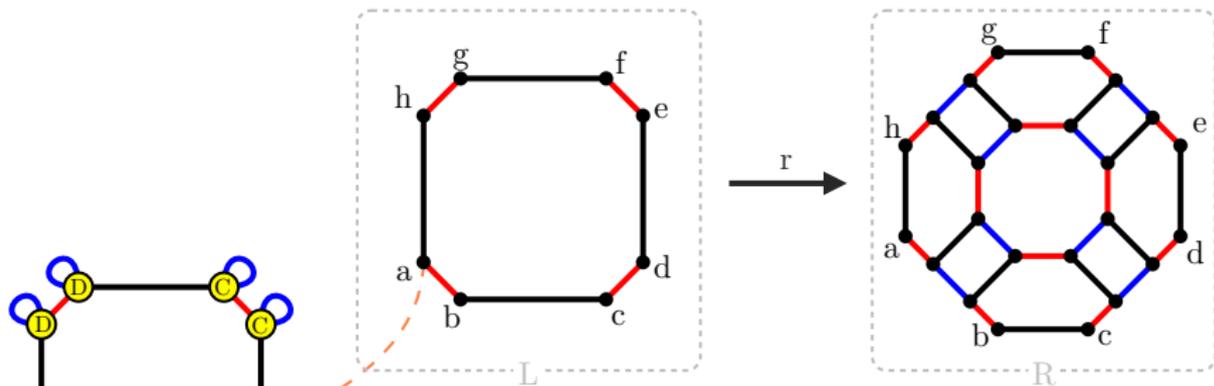
Comment afficher les objets ?

- ▶ Différentes géométries pour une même topologie.



Plongement sur les cellules topologiques : couleur.

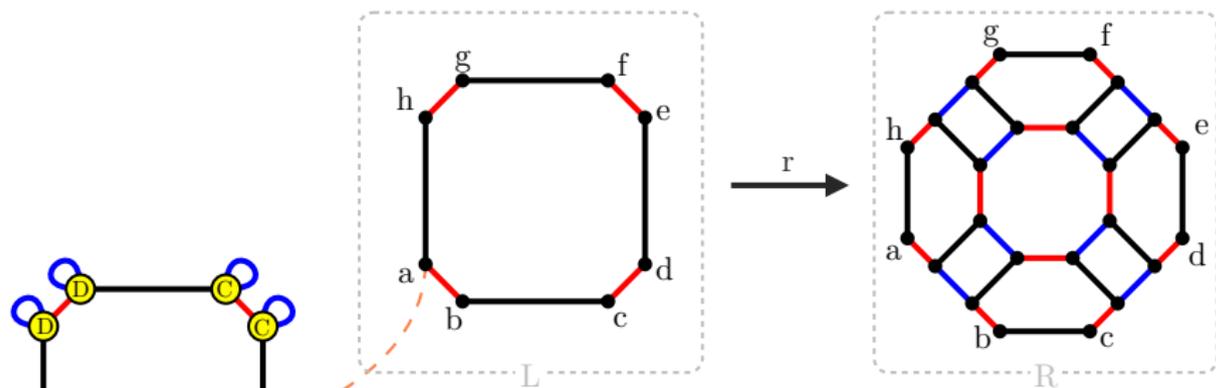
Expressions de plongement



Trois familles d'expressions :

- Accès aux plongements
- ▶ a.couleur = ● ▶ a.position = A

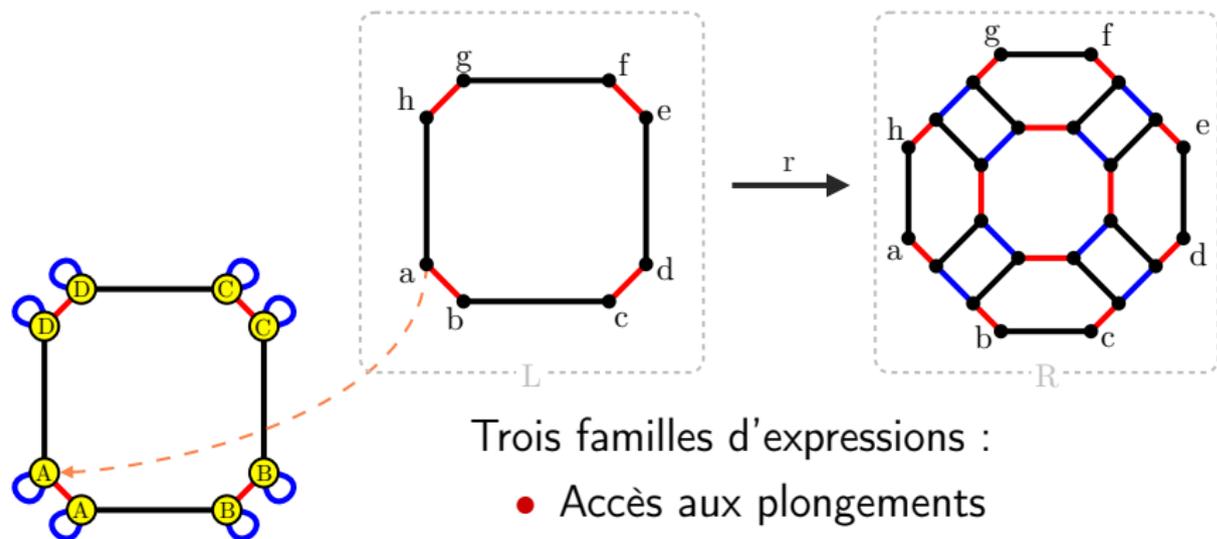
Expressions de plongement



Trois familles d'expressions :

- Accès aux plongements
- Calculs sur les types de plongement
- ▶ ● + ● = ● (RGB)

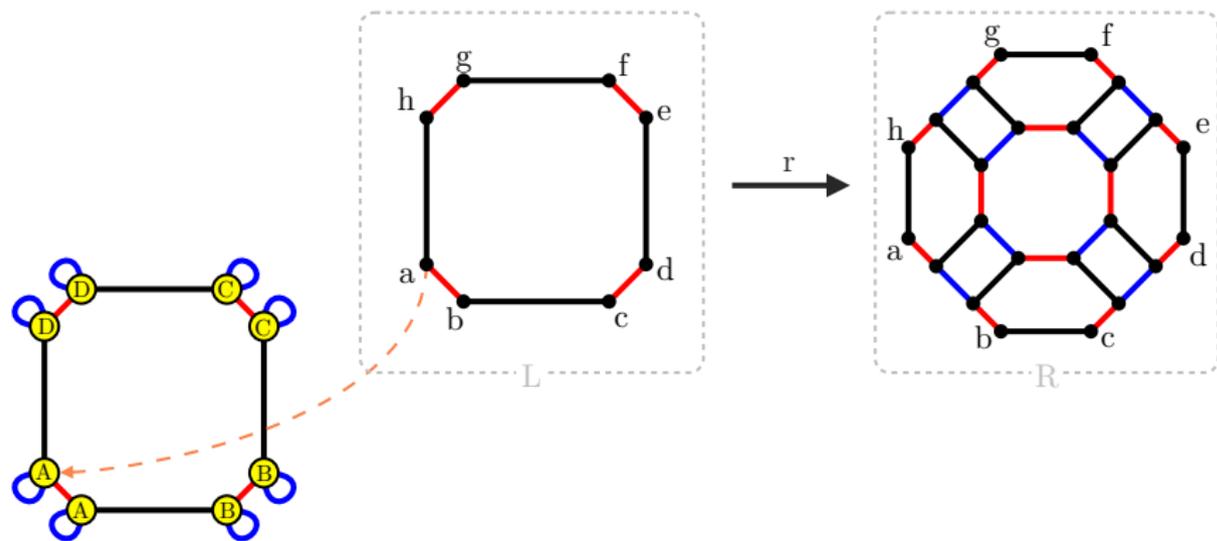
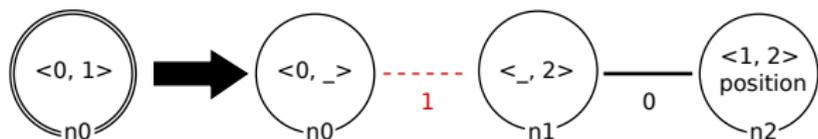
Expressions de plongement



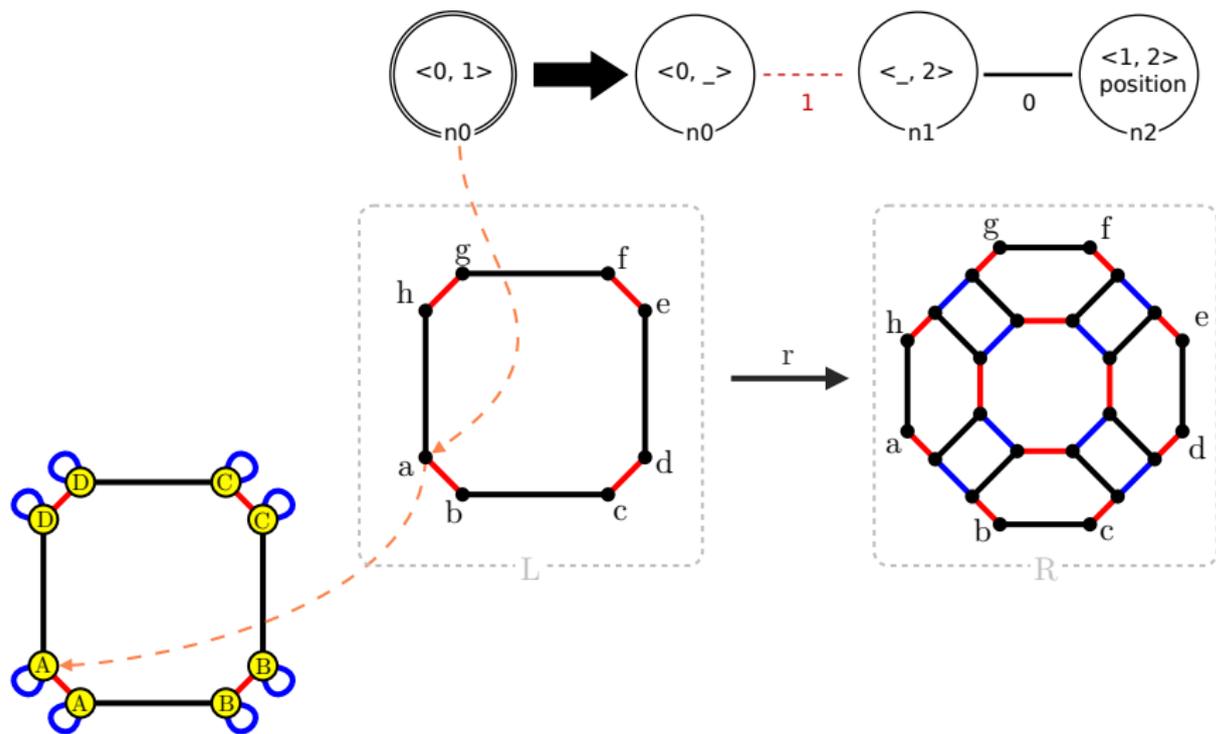
Trois familles d'expressions :

- Accès aux plongements
- Calculs sur les types de plongement
- Parcours sur la G-carte
 - ▶ $a@0.position = D$
 - ▶ $position_{\langle 0,1 \rangle}(a) = \{A, B, C, D\}$

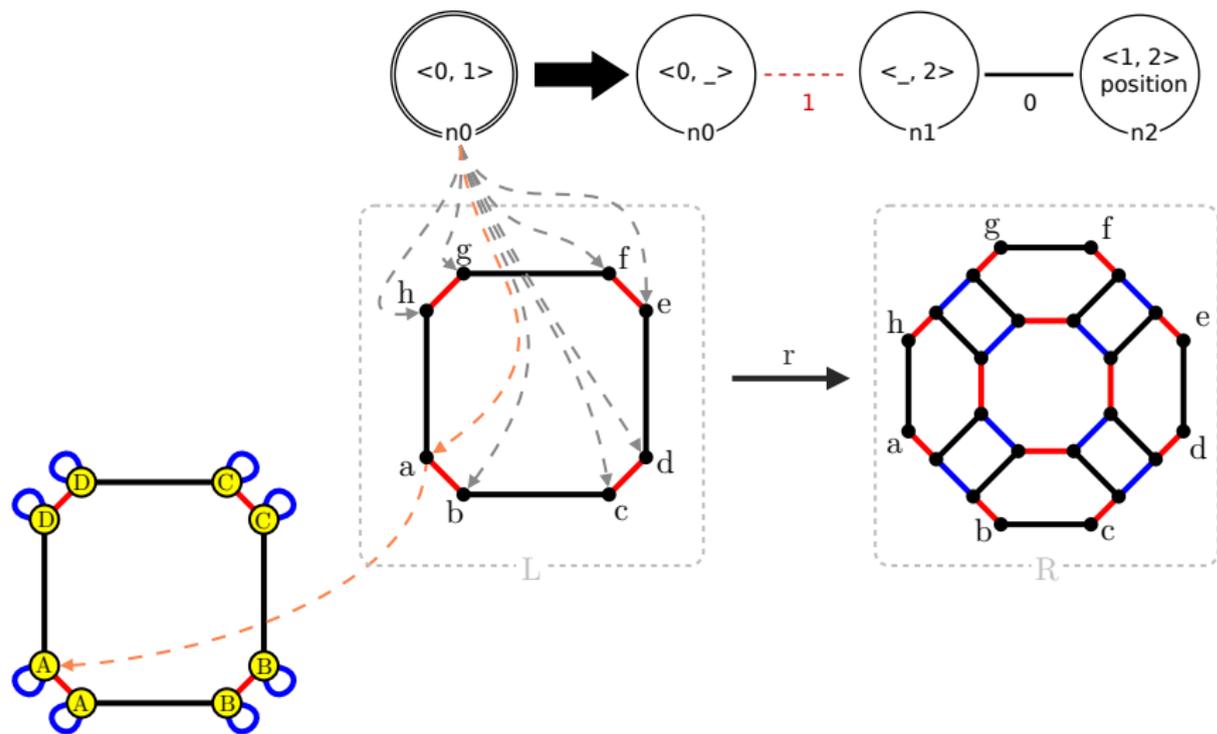
Extension aux schémas de règles



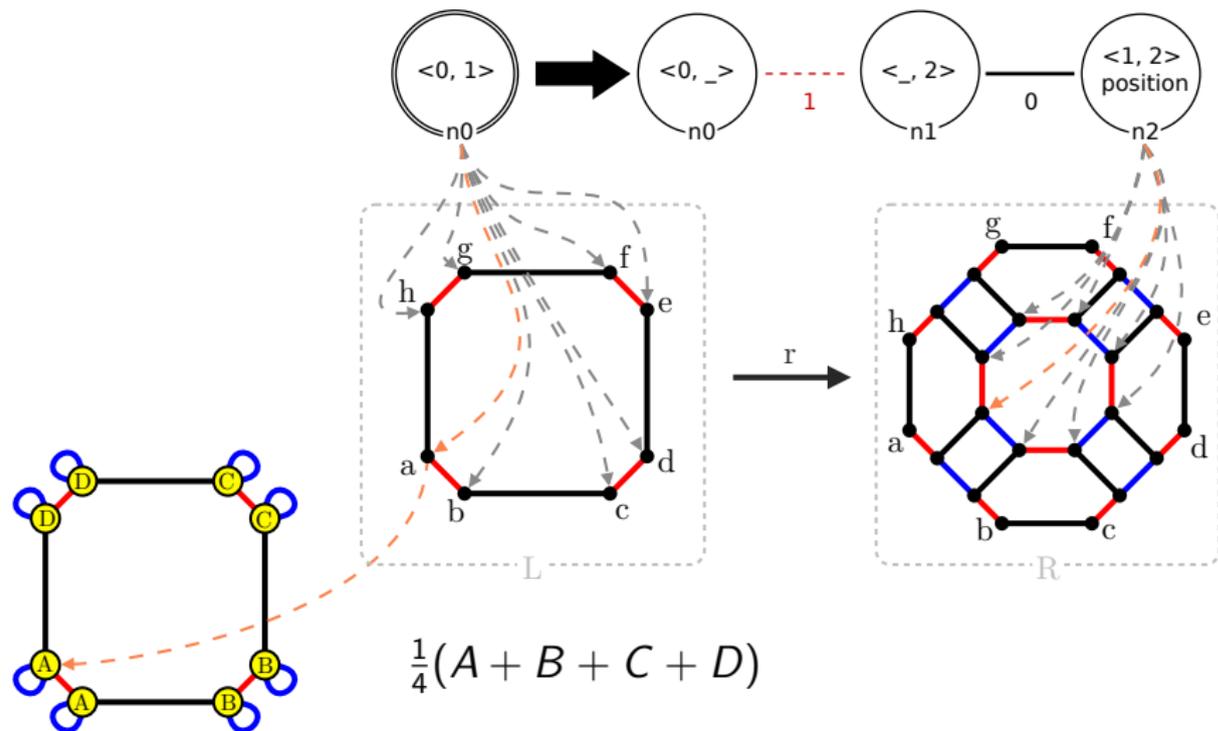
Extension aux schémas de règles



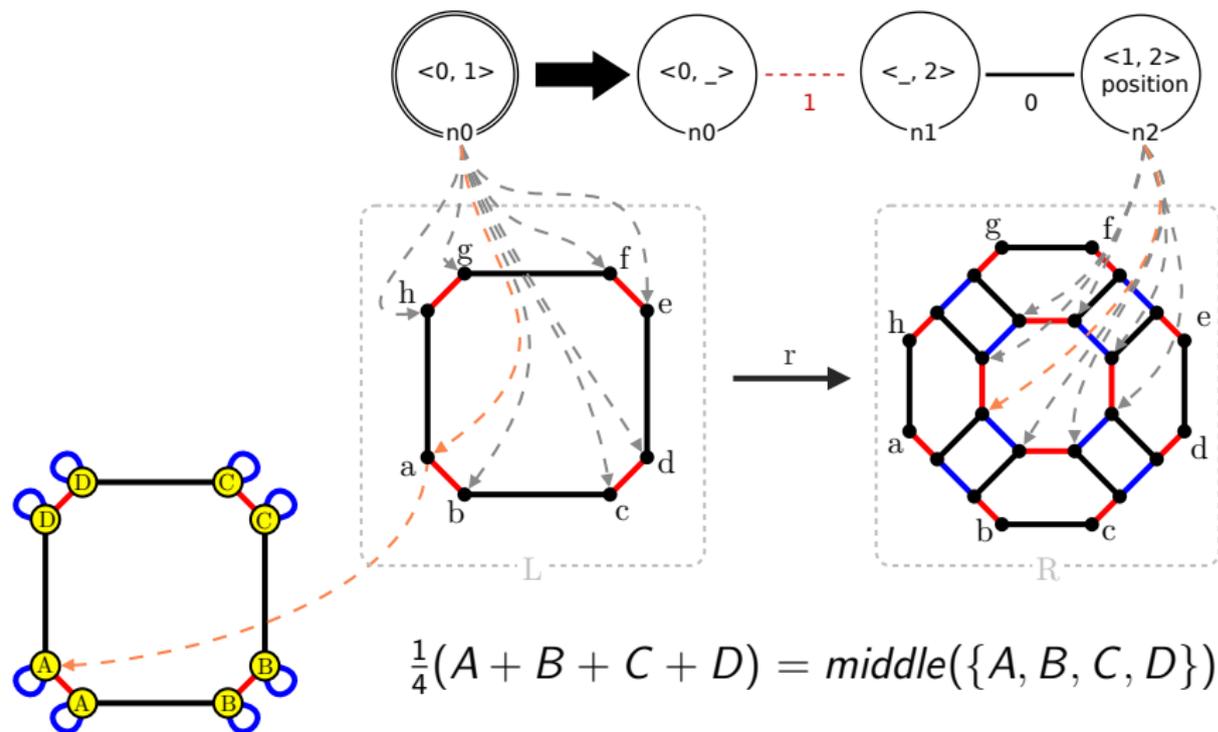
Extension aux schémas de règles



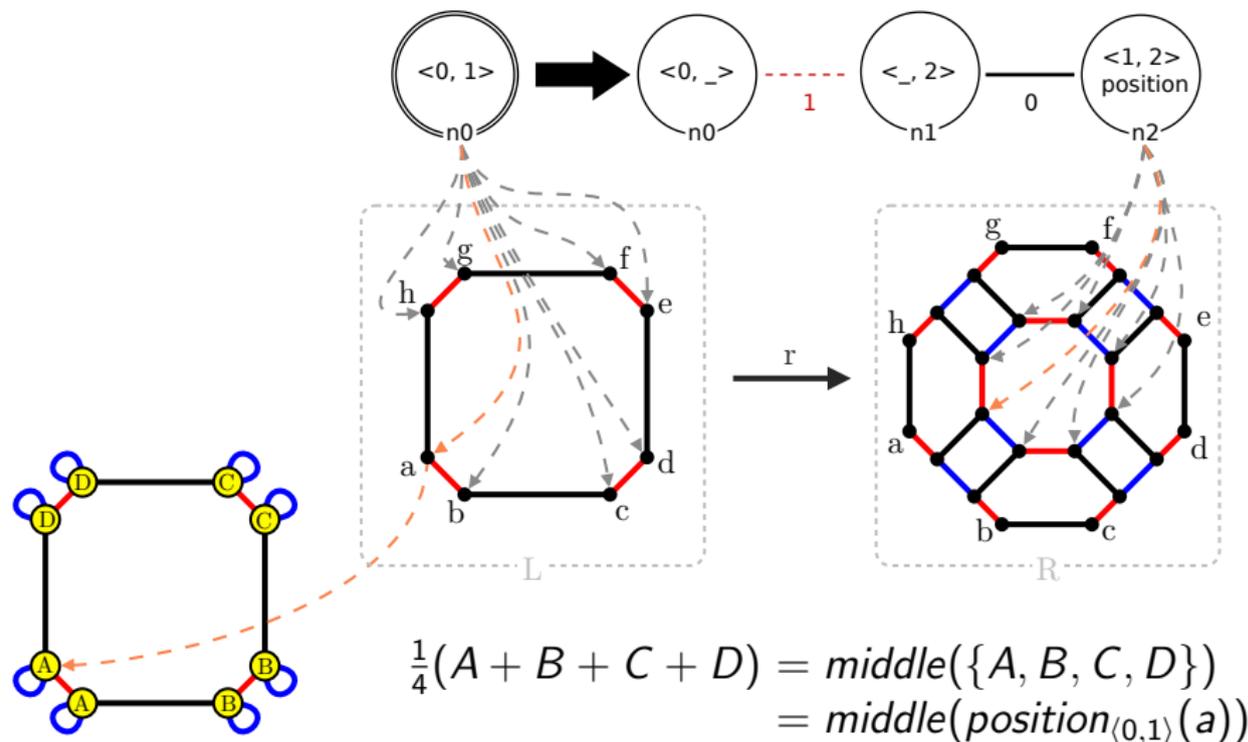
Extension aux schémas de règles



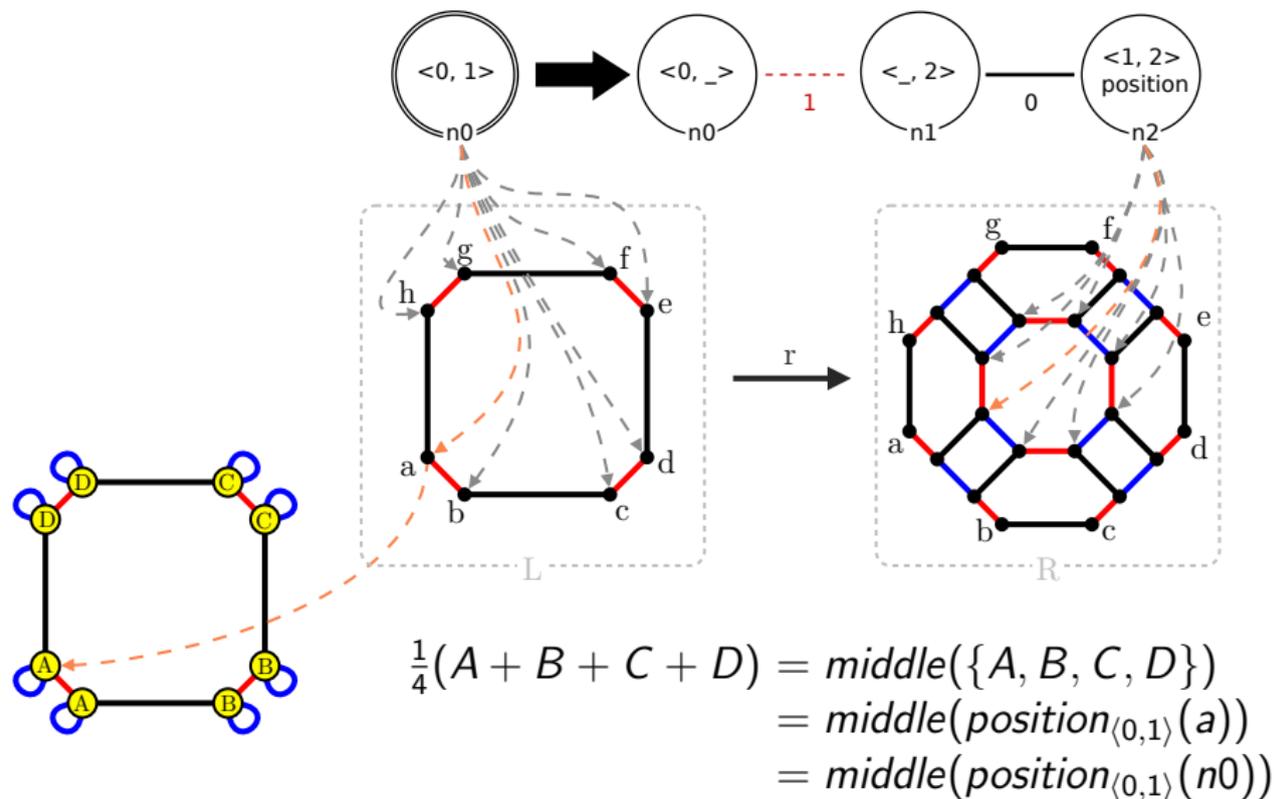
Extension aux schémas de règles



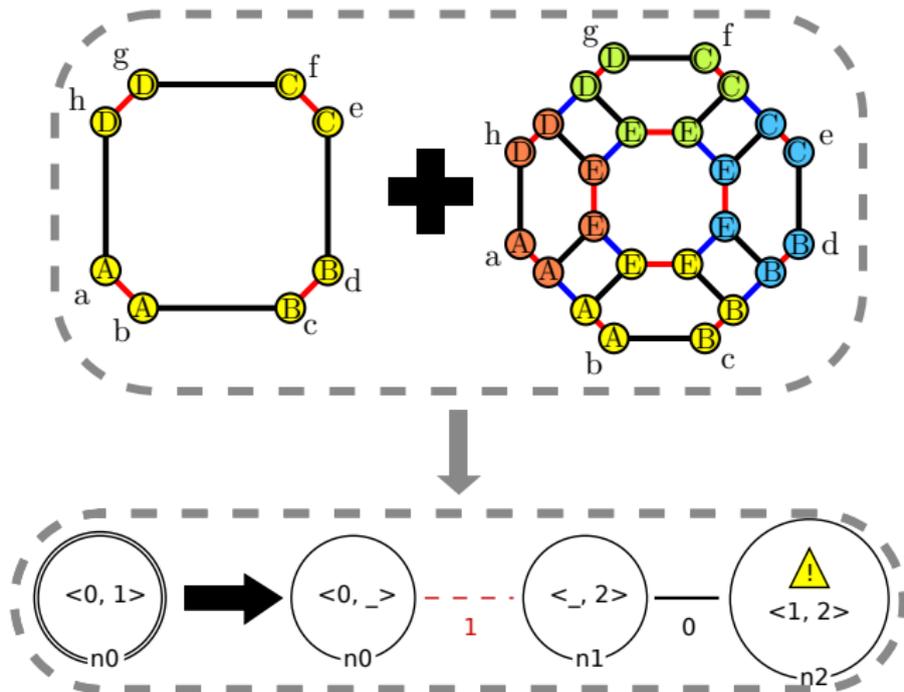
Extension aux schémas de règles



Extension aux schémas de règles



Inférence géométrique



Il manque les calculs des plongements

Méthode (inférence des positions)

- **Hypothèse** : Positions des points de l'objet cible C calculés par combinaison affine des points de l'objet d'origine O .

Méthode (inférence des positions)

► **Hypothèse** : Positions des points de l'objet cible C calculés par combinaison affine des points de l'objet d'origine O .

Pour chaque position p d'un point de C , on cherche une expression :

$$p = \sum_{i=0}^k a_i p_i + t$$

avec :

- p : position du point cible (connu)

Méthode (inférence des positions)

► **Hypothèse** : Positions des points de l'objet cible C calculés par combinaison affine des points de l'objet d'origine O .

Pour chaque position p d'un point de C , on cherche une expression :

$$p = \sum_{i=0}^k a_i p_i + t$$

avec :

- p : position du point cible (connu)
- p_i : position d'un point d'origine (connu)

Méthode (inférence des positions)

► **Hypothèse** : Positions des points de l'objet cible C calculés par combinaison affine des points de l'objet d'origine O .

Pour chaque position p d'un point de C , on cherche une expression :

$$p = \sum_{i=0}^k a_i p_i + t$$

avec :

- p : position du point cible (connu)
- p_i : position d'un point d'origine (connu)
- a_i : coefficient (à déterminer)

Méthode (inférence des positions)

► **Hypothèse** : Positions des points de l'objet cible C calculés par combinaison affine des points de l'objet d'origine O .

Pour chaque position p d'un point de C , on cherche une expression :

$$p = \sum_{i=0}^k a_i p_i + t$$

avec :

- p : position du point cible (connu)
- p_i : position d'un point d'origine (connu)
- a_i : coefficient (à déterminer)
- t : translation intrinsèque (à déterminer, optionnel)

Schémas de règle : besoin d'abstraction

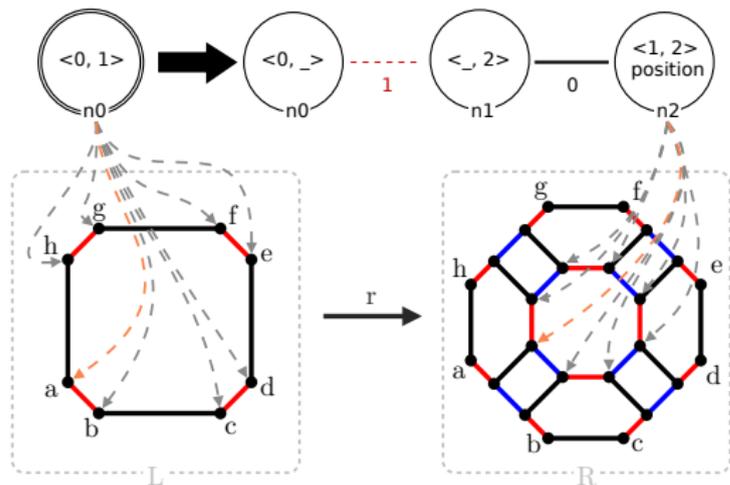
On cherche $(a_i)_i$ tels que :

$$p = \sum_{i=0}^k a_i p_i + t$$

Schémas de règle : besoin d'abstraction

On cherche $(a_i)_i$ tels que :

$$p = \sum_{i=0}^k a_i p_i + t$$



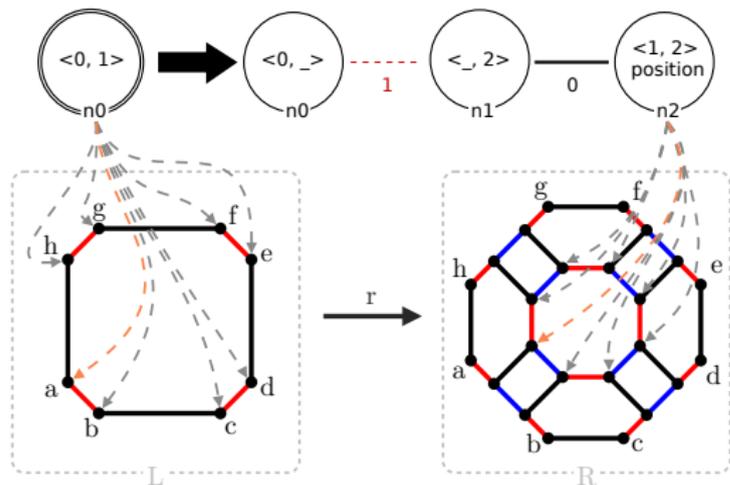
Problème : il faut la même expression pour tous les brins.

► Un schéma de règle crée des abstractions des cellules topologiques.

Schémas de règle : besoin d'abstraction

On cherche $(a_i)_i$ tels que :

$$p = \sum_{i=0}^k a_i p_i + t$$



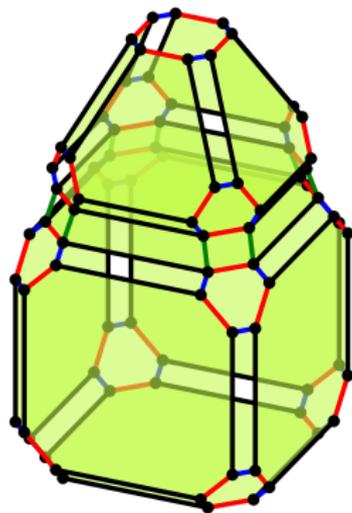
Problème : il faut la même expression pour tous les brins.

► Un schéma de règle crée des abstractions des cellules topologiques.

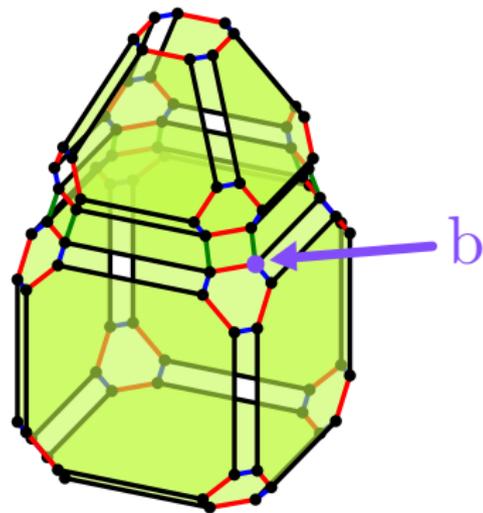
Solution : Exploiter la topologie.

► Positions remarquables, exprimées de la même manière sur tous les brins.

Positions remarquables



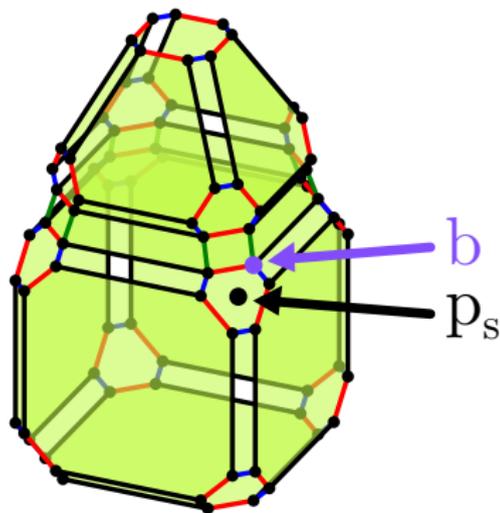
Positions remarquables



Positions remarquables

avec

- p_s : position du sommet

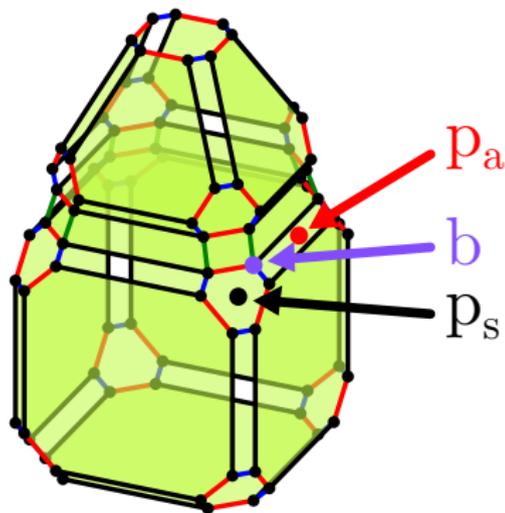


$$p_s = \text{middle}(\text{position}_{\langle 1,2,3 \rangle}(b))$$

Positions remarquables

avec

- p_s : position du sommet
- p_a : milieu de l'arête

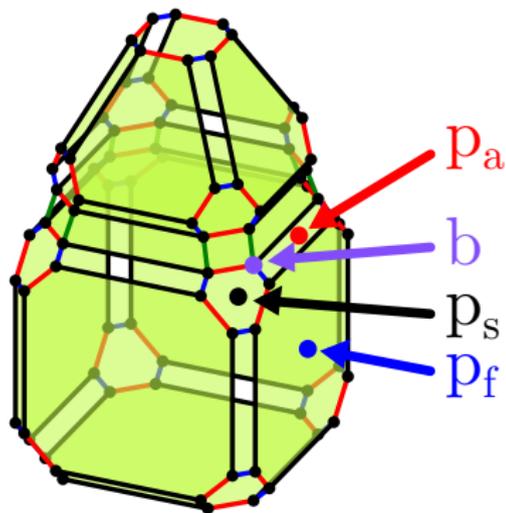


$$p_a = \text{middle}(\text{position}_{\langle 0,2,3 \rangle}(b))$$

Positions remarquables

avec

- p_s : position du sommet
- p_a : milieu de l'arête
- p_f : milieu de la face

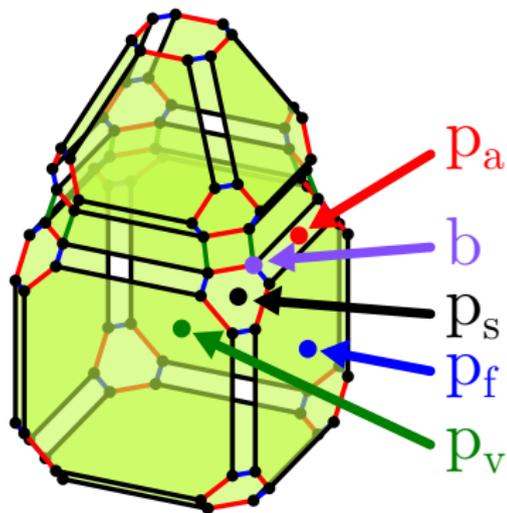


$$p_f = \text{middle}(\text{position}_{\langle 0,1,3 \rangle}(b))$$

Positions remarquables

avec

- p_s : position du sommet
- p_a : milieu de l'arête
- p_f : milieu de la face
- p_v : milieu du volume

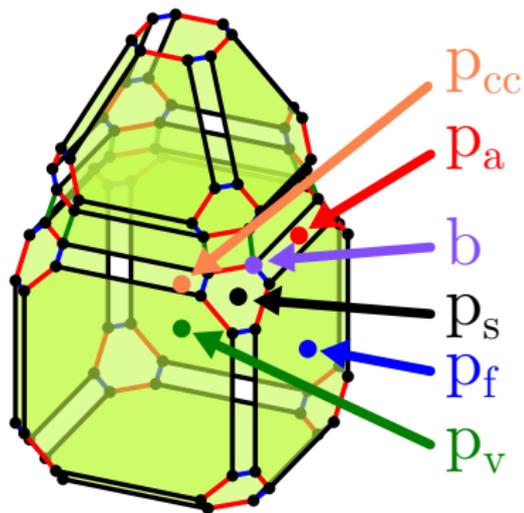


$$p_v = \text{middle}(\text{position}_{\langle 0,1,2 \rangle}(b))$$

Positions remarquables

avec

- p_s : position du sommet
- p_a : milieu de l'arête
- p_f : milieu de la face
- p_v : milieu du volume
- p_{cc} : milieu de la composante connexe

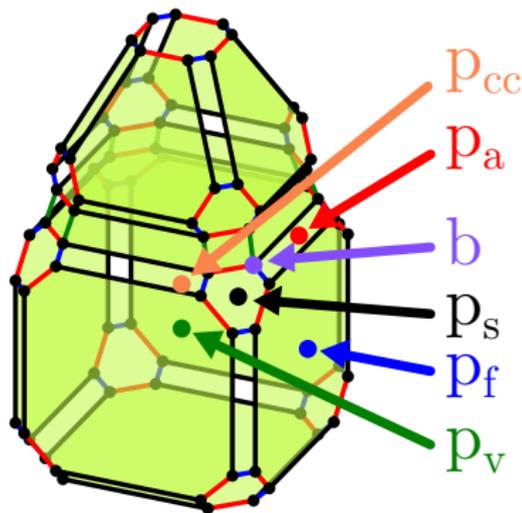


$$p_{cc} = \text{middle}(\text{position}_{\langle 0,1,2,3 \rangle}(b))$$

Positions remarquables

avec

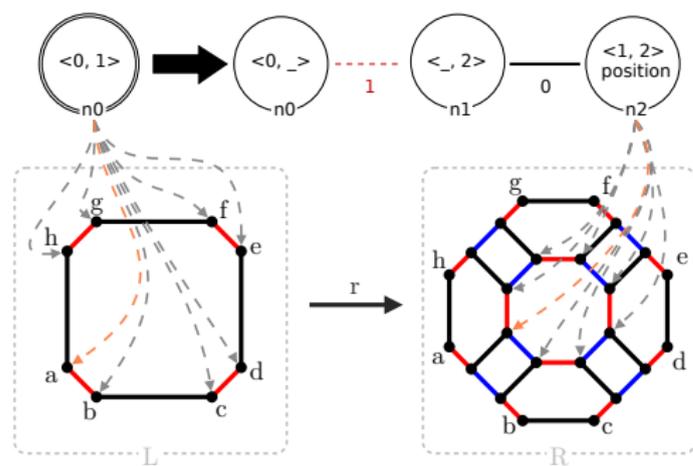
- p_s : position du sommet
- p_a : milieu de l'arête
- p_f : milieu de la face
- p_v : milieu du volume
- p_{cc} : milieu de la composante connexe



Via les positions remarquables, le système se réécrit :

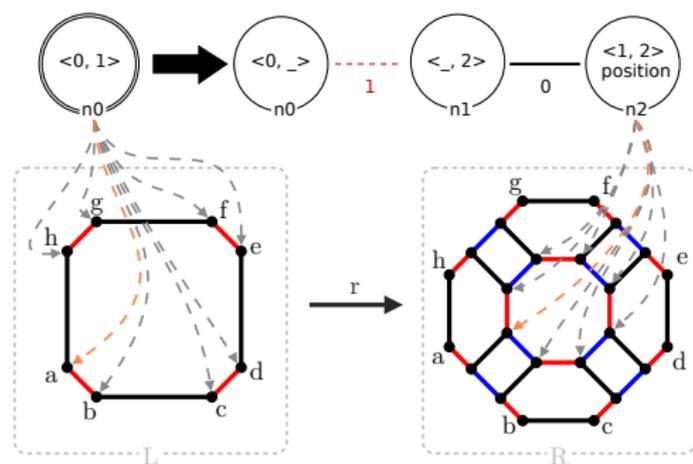
$$p = a_s p_s + a_a p_a + a_f p_f + a_v p_v + a_{cc} p_{cc} + t$$

Exemple de résolution



Position de n_2 dépend uniquement de n_0 .

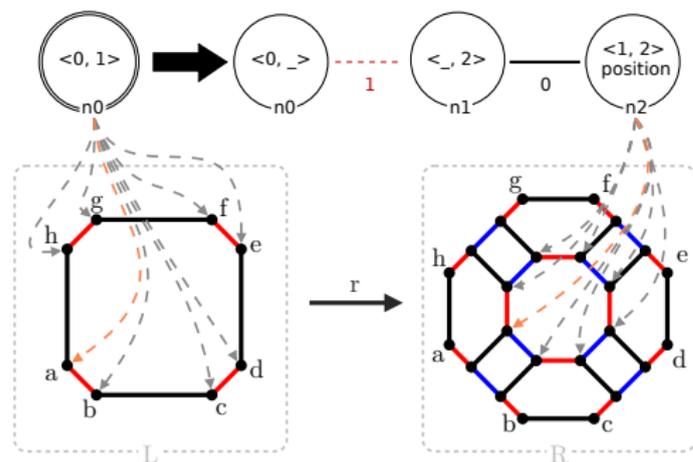
Exemple de résolution



Position de n_2 dépend uniquement de n_0 .

$$n_2.position = a_s n_0.p_s + a_a n_0.p_a + a_f n_0.p_f + a_v n_0.p_v + a_{cc} n_0.p_{cc} + t$$

Exemple de résolution

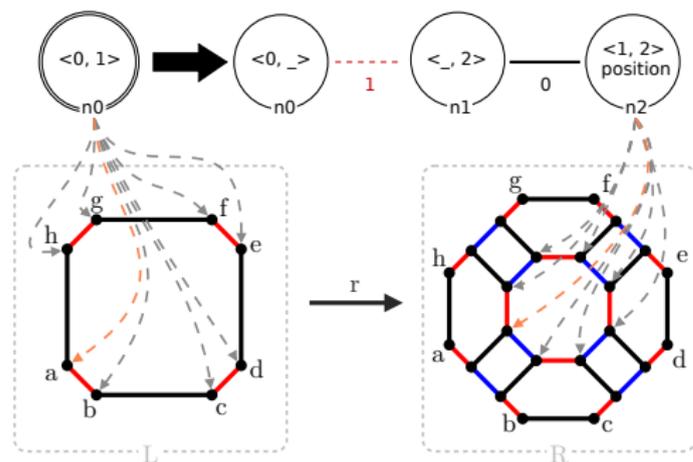


Position de n_2 dépend uniquement de n_0 .

- Une équation par brin de n_0 (8 brins)

$$n_2.position = a_s n_0.p_s + a_a n_0.p_a + a_f n_0.p_f + a_v n_0.p_v + a_{cc} n_0.p_{cc} + t$$

Exemple de résolution

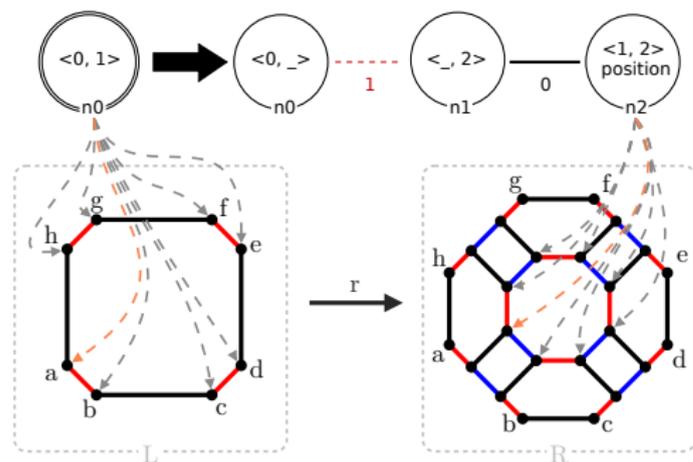


Position de n_2 dépend uniquement de n_0 .

- Une équation par brin de n_0 (8 brins)
- Séparation sur x, y, z

$$n_2.position = a_s n_0.p_s + a_a n_0.p_a + a_f n_0.p_f + a_v n_0.p_v + a_{cc} n_0.p_{cc} + t$$

Exemple de résolution

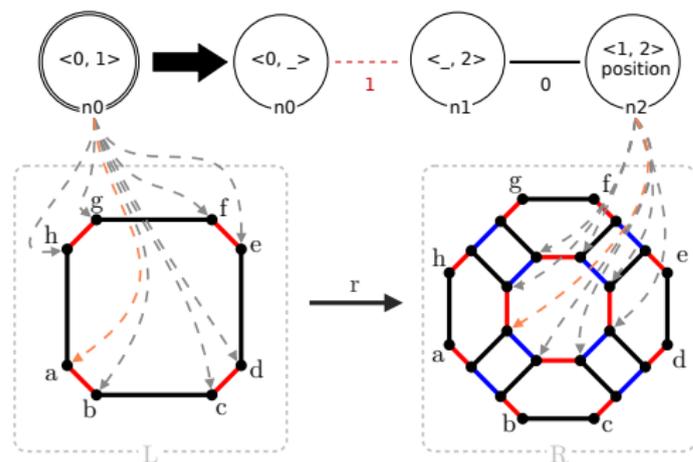


Position de n_2 dépend uniquement de n_0 .

- Une équation par brin de n_0 (8 brins)
- Séparation sur x, y, z
- Système de 24 équations avec 8 variables

$$n_2.position = a_s n_0.p_s + a_a n_0.p_a + a_f n_0.p_f + a_v n_0.p_v + a_{cc} n_0.p_{cc} + t$$

Exemple de résolution



Position de $n2$ dépend uniquement de $n0$.

- Une équation par brin de $n0$ (8 brins)
- Séparation sur x, y, z
- Système de 24 équations avec 8 variables

$$n2.position = a_s n0.p_s + a_a n0.p_a + a_f n0.p_f + a_v n0.p_v + a_{cc} n0.p_{cc} + t$$

► Résolution par CSP (OR-Tools, Z3)

Exemple : triangulation

► Equation globale :

$$n2.position = a_s n0.p_s + a_a n0.p_a + a_f n0.p_f + a_v n0.p_v + a_{cc} n0.p_{cc} + t$$

Exemple : triangulation

- ▶ Equation globale :

$$n2.position = a_s n0.p_s + a_a n0.p_a + a_f n0.p_f + a_v n0.p_v + a_{cc} n0.p_{cc} + t$$

- ▶ Système généré (sans la composante z)

$$\left\{ \begin{array}{l} (0.5; 0.5) = a_s * (0; 0) + a_a * (0.5; 0) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 0) + a_a * (0.5; 0) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 0) + a_a * (1; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 1) + a_a * (1; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 1) + a_a * (0.5; 1) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (0; 1) + a_a * (0.5; 1) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (0; 1) + a_a * (0; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (0; 0) + a_a * (0; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \end{array} \right.$$

Exemple : triangulation

► Equation globale :

$$n2.position = a_s n0.p_s + a_a n0.p_a + a_f n0.p_f + a_v n0.p_v + a_{cc} n0.p_{cc} + t$$

► Système généré (sans la composante z)

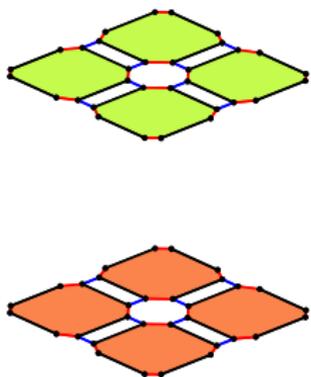
$$\left\{ \begin{array}{l} (0.5; 0.5) = a_s * (0; 0) + a_a * (0.5; 0) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 0) + a_a * (0.5; 0) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 0) + a_a * (1; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 1) + a_a * (1; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (1; 1) + a_a * (0.5; 1) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (0; 1) + a_a * (0.5; 1) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (0; 1) + a_a * (0; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \\ (0.5; 0.5) = a_s * (0; 0) + a_a * (0; 0.5) + a_f * (0.5; 0.5) + a_v * (0.5; 0.5) + a_{cc} * (0.5; 0.5) + (tx; ty) \end{array} \right.$$

► Solution trouvée :

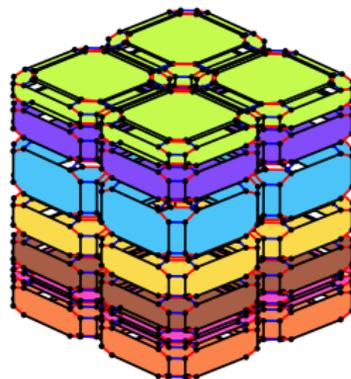
- $a_s = -6.601425600620388E - 17$
- $a_a = 0.0$
- $a_f = 1.0$
- $a_v = 0.0$
- $a_{cc} = 0.0$
- $t = (0.0, 0.0)$

Exemple en géologie (avec géométrie)

Avant

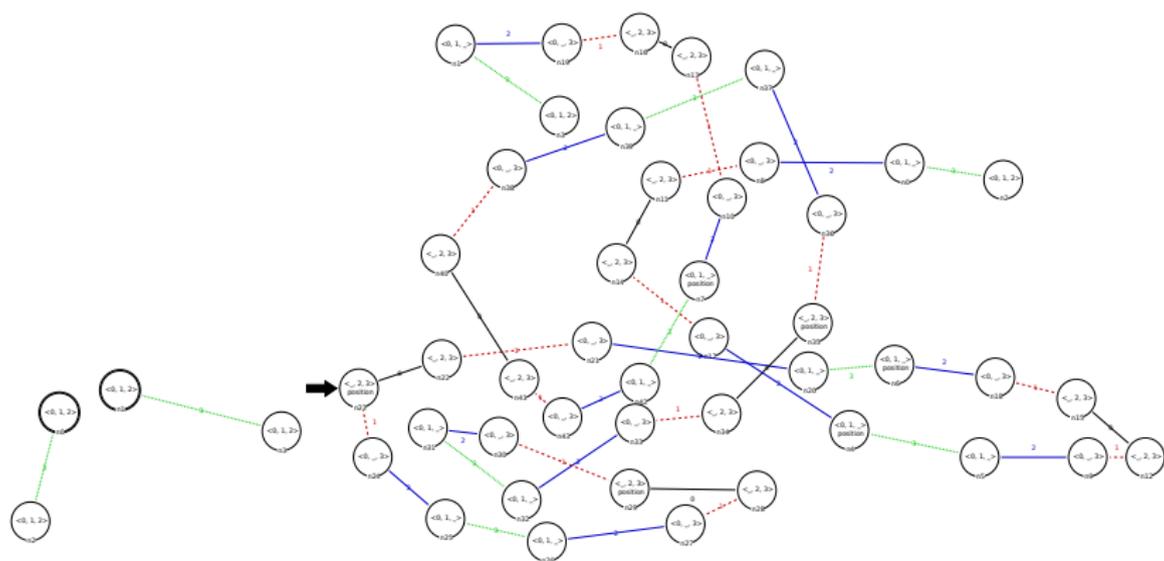


Après



Exemple en géologie (avec géométrie)

Operation



Temps d'inférence : ~ 26 ms pour la topologie,
 ~ 549 ms pour la géométrie

Exemple en géologie (avec géométrie)

Avant



Exemple en géologie (avec géométrie)

Après



Exemple : tapis de Menger

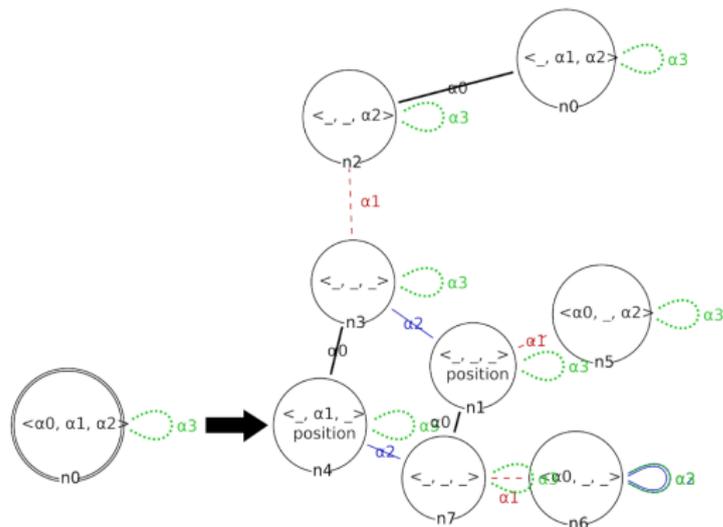
The screenshot displays the Inference Studio interface for creating a Menger carpet. The central 3D view shows a blue carpet structure. The right sidebar lists the inference rules used, including 'category48', 'color', 'great', 'duplication', 'exhibit', 'extrusion', 'geologie', 'hexadominant', 'infer', 'infertool', 'modélisation', 'normal', 'remain_test', 'rotation', 'sew', 'subdivision', 'suppression', 'tordu', 'transform', and 'triangulation'. The bottom console shows the command 'Inference Target orbit (put "." in the orbit): <a5, a1, a2>' and a 'Set' button.

Exemple : tapis de Menger

The screenshot displays the Inference Studio interface for constructing a Menger carpet. The main window is divided into several panels:

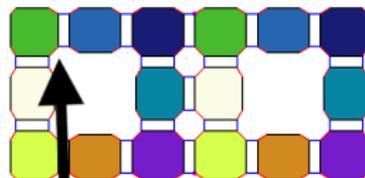
- Left Panel (Rules):** A list of inference rules such as 'Inference category08', 'color', 'create', 'duplication', 'exhibit', 'extrusion', 'geologie', 'hexadominant', 'infer', 'inferool', 'modélisation', 'normal', 'remain_test', 'rotation', 'sew', 'subdivision', 'suppression', 'tordu', 'transform', and 'triangulation'. The 'category48' rule is currently selected.
- Center View:** A 3D perspective view of the carpet, showing a grid of colored squares (green, blue, yellow, orange, purple) arranged in a pattern. A red line indicates the current position of the carpet.
- Right Panel (Parameters):** A list of parameters for the selected rule, including 'a0', 'a1', 'a2', 'a3', 'a4', 'a5', 'a6', 'a7', 'a8', 'a9', 'a10', 'a11', 'a12', 'a13', 'a14', 'a15', 'a16', 'a17', 'a18', 'a19', 'a20', 'a21', 'a22', 'a23', 'a24', 'a25', 'a26', 'a27', 'a28', 'a29', 'a30', 'a31', 'a32', 'a33', 'a34', 'a35', 'a36', 'a37', 'a38', 'a39', 'a40', 'a41', 'a42', 'a43', 'a44', 'a45', 'a46', 'a47', 'a48', 'a49', 'a50', 'a51', 'a52', 'a53', 'a54', 'a55', 'a56', 'a57', 'a58', 'a59', 'a60', 'a61', 'a62', 'a63', 'a64', 'a65', 'a66', 'a67', 'a68', 'a69', 'a70', 'a71', 'a72', 'a73', 'a74', 'a75', 'a76', 'a77', 'a78', 'a79', 'a80', 'a81', 'a82', 'a83', 'a84', 'a85', 'a86', 'a87', 'a88', 'a89', 'a90', 'a91', 'a92', 'a93', 'a94', 'a95', 'a96', 'a97', 'a98', 'a99', 'a100'. The 'Move' parameter is set to 'a0: 47', 'a1: 45', 'a2: 85', 'a3: 46'.
- Bottom Panel (Console):** A console window showing the progress of the inference process, including the number of rules applied and the current state of the carpet.

Exemple : tapis de Menger



```

Expression (n1#position)
1 Point3 res = new Point3(0.0,0.0,0.0);
2 Point3 p0 = Point3::middle(<1, 2, 3>_position(n0));
3 p0.scale(0.333333);
4 res.add(p0);
5 Point3 p1 = Point3::middle(<0, 2, 3>_position(n0));
6 p1.scale(0.666667);
7 res.add(p1);
8 return res;
    
```



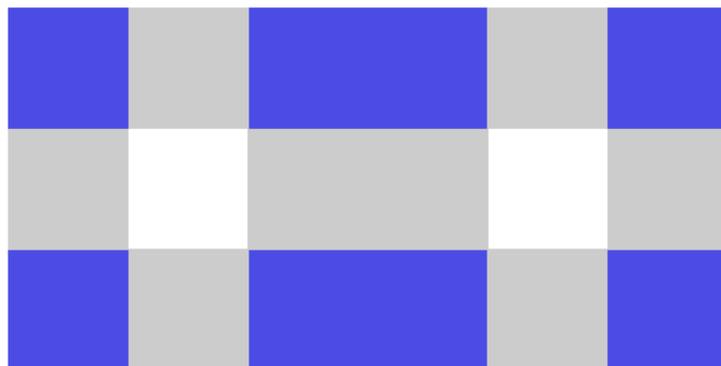
```

Expression (n4#position)
1 Point3 res = new Point3(0.0,0.0,0.0);
2 Point3 p0 = Point3::middle(<1, 2, 3>_position(n0));
3 p0.scale(0.333333);
4 res.add(p0);
5 Point3 p2 = Point3::middle(<0, 1, 3>_position(n0));
6 p2.scale(0.6666699999999999);
7 res.add(p2);
8 return res;
    
```

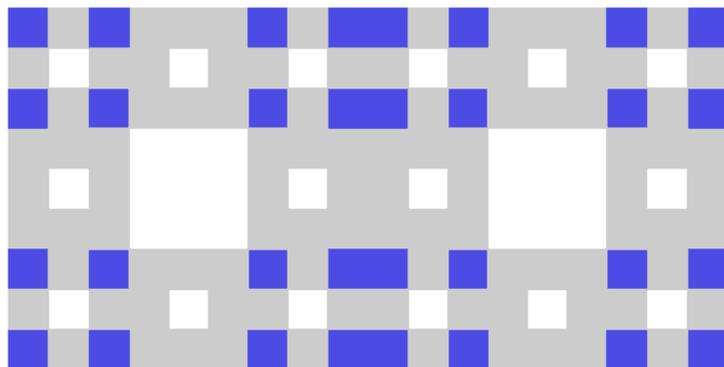
Exemple : tapis de Menger



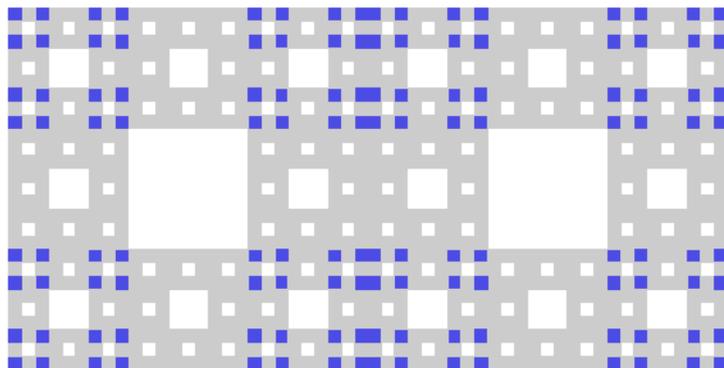
Exemple : tapis de Menger



Exemple : tapis de Menger



Exemple : tapis de Menger

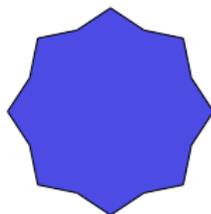
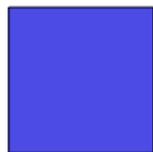


Limites

- 1. On ne sait pas résoudre le système.

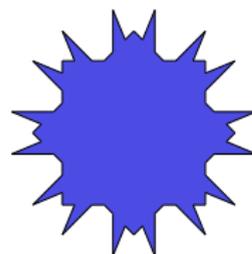
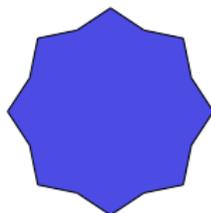
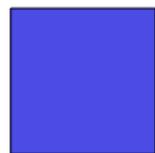
Limites

- 1. On ne sait pas résoudre le système.
- 2. On ne trouve pas la "bonne" solution.



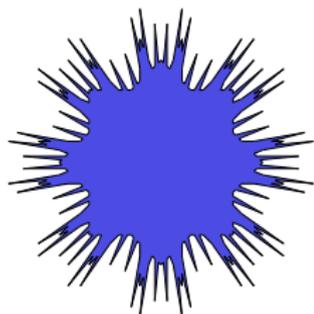
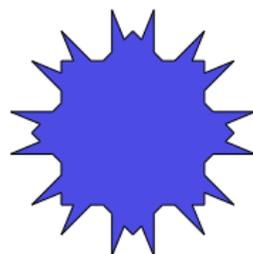
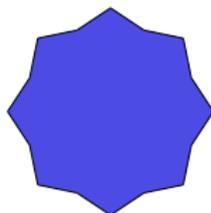
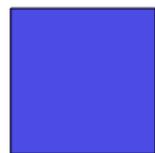
Limites

- 1. On ne sait pas résoudre le système.
- 2. On ne trouve pas la "bonne" solution.



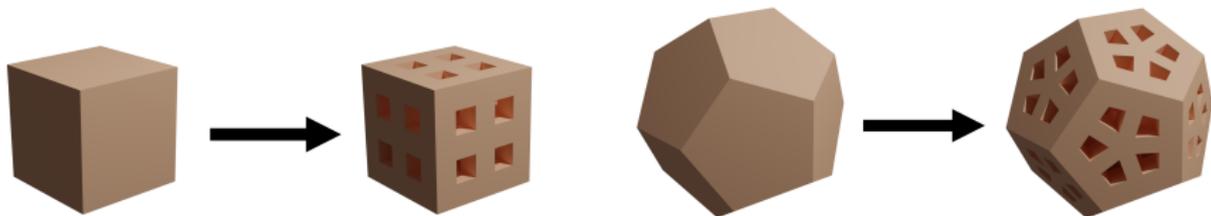
Limites

- 1. On ne sait pas résoudre le système.
- 2. On ne trouve pas la "bonne" solution.



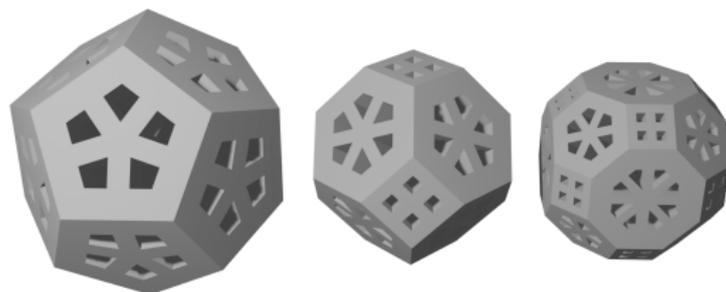
Conclusion

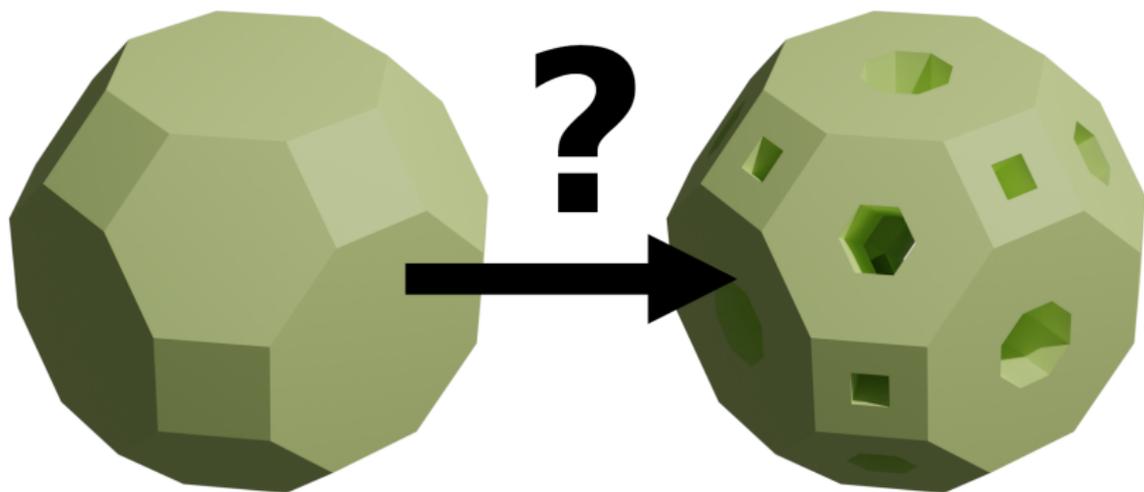
- ▶ On infère une opération de modélisation géométrique à partir de deux **instances** d'un objet **avant** et **après** modification.
- ▶ **Topologie** : Algorithme de parcours de graphe.
- ▶ **Géométrie** : Combinaison affine de points remarquables.

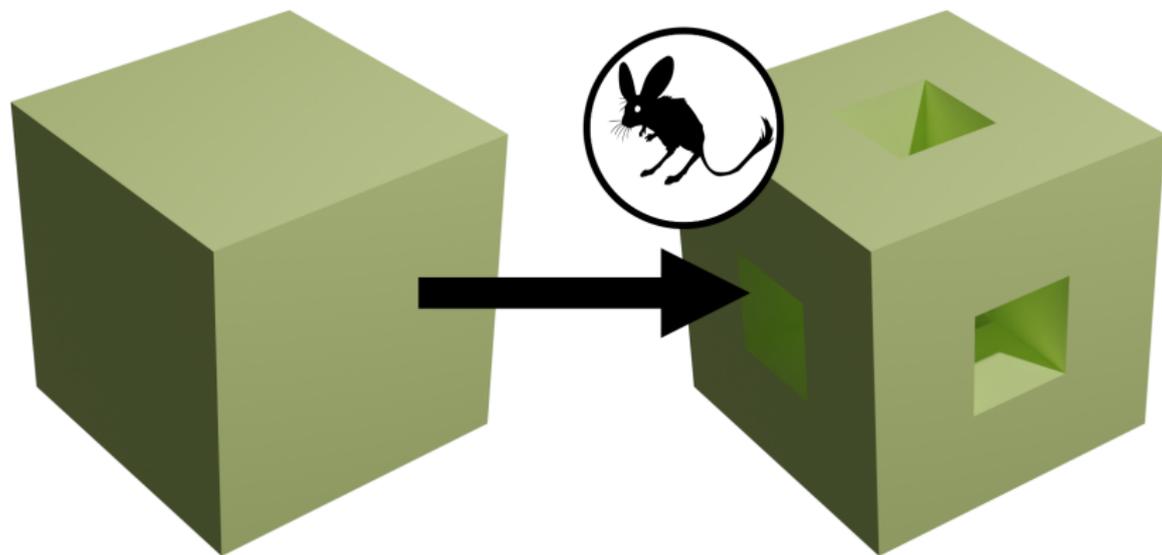


Perspectives

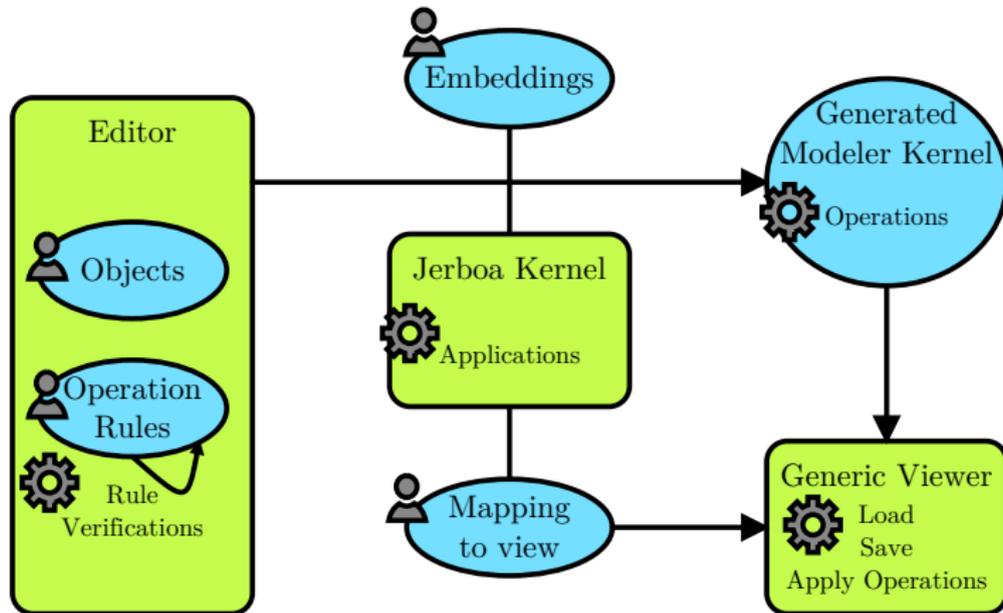
- Autres positions remarquables (expressions de parcours)
- Autres familles de fonctions (plutôt que les combinaisons affines)
- Autres plongements



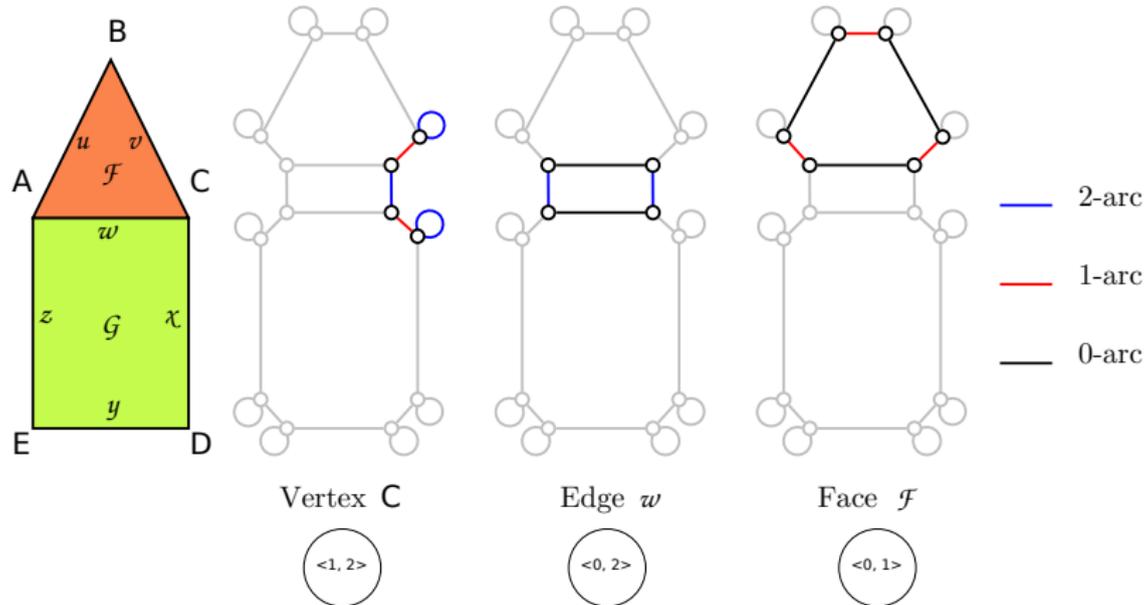




Jerboa's architecture



Orbits



► Defined as subgraph induced by a subset of dimensions.

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.

Réécriture

Réécriture de mots

- un alphabet Σ
- un ensemble de règles de réécritures $u \rightarrow v$ (u et v sont des mots de Σ^*)

Réécriture

Réécriture de mots

- un alphabet Σ
- un ensemble de règles de réécritures $u \rightarrow v$ (u et v sont des mots de Σ^*)

$aux_{\sqcup}JIG \rightarrow au_{\sqcup}GTMG$

Réécriture

Réécriture de mots

- un alphabet Σ
- un ensemble de règles de réécritures $u \rightarrow v$ (u et v sont des mots de Σ^*)

$\text{aux}_{\sqcup} \text{JIG} \rightarrow \text{au}_{\sqcup} \text{GTMG}$

$\text{Vous}_{\sqcup} \text{assistez}_{\sqcup} \text{bien}_{\sqcup} \text{aux}_{\sqcup} \text{JIG}_{\sqcup}!$

Réécriture

Réécriture de mots

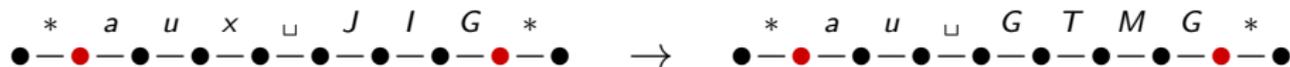
- un alphabet Σ
- un ensemble de règles de réécritures $u \rightarrow v$ (u et v sont des mots de Σ^*)

$\text{aux}_\perp \text{JIG} \rightarrow \text{au}_\perp \text{GTMG}$

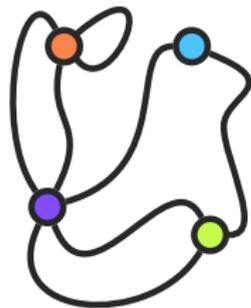
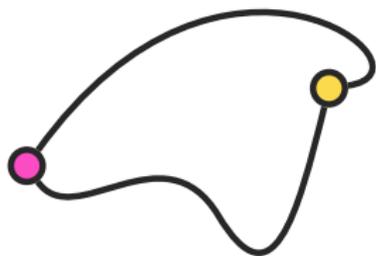
$\text{Vous}_\perp \text{assistez}_\perp \text{bien}_\perp \text{aux}_\perp \text{JIG}_\perp! \rightarrow \text{Vous}_\perp \text{assistez}_\perp \text{bien}_\perp \text{au}_\perp \text{GTMG}_\perp!$

Comment réécrire un graphe ?

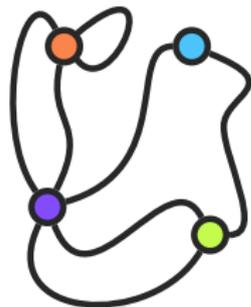
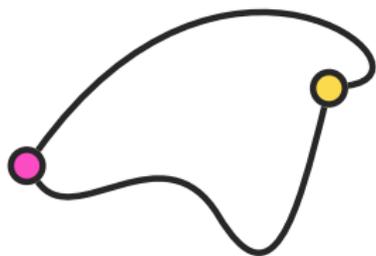
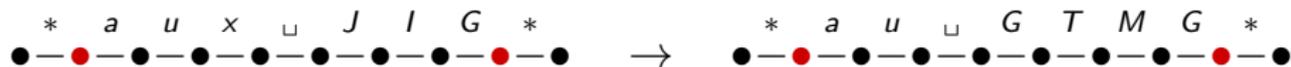
Comment réécrire un graphe ? [Ehrig 1979]



Comment réécrire un graphe? [Ehrig 1979]



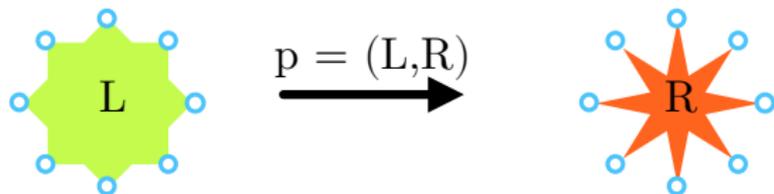
Comment réécrire un graphe? [Ehrig 1979]



- ▶ Il n'y a pas de début ni de fin dans un graphe.
- On identifie des éléments de recollement.

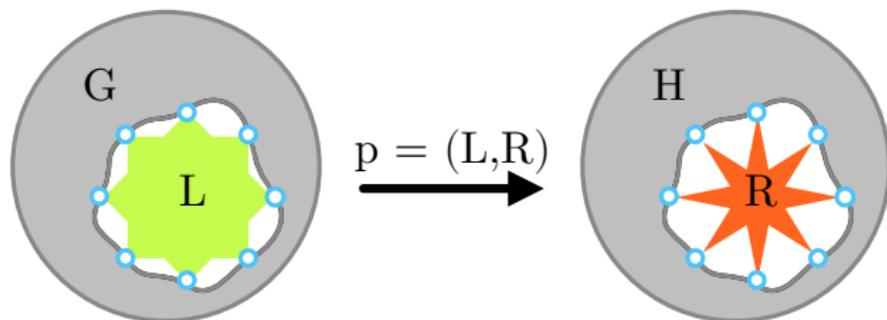
Règles de transformation de graphes

- **But** : Généraliser les techniques de réécriture de termes aux graphes.



Règles de transformation de graphes

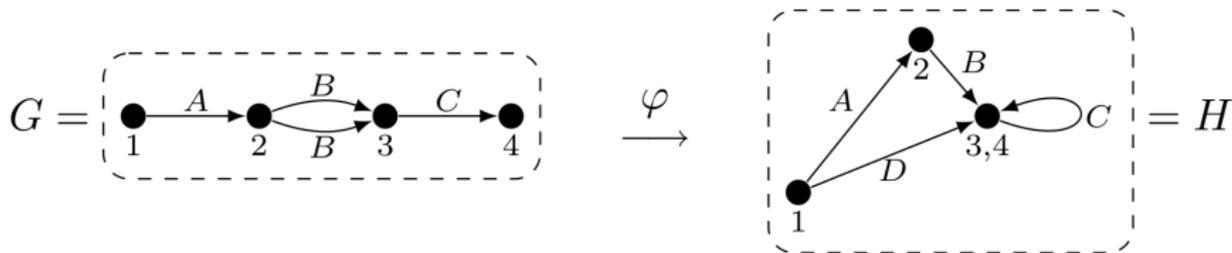
► **But** : Généraliser les techniques de réécriture de termes aux graphes.



- Trouver un motif similaire à L dans G ,
- Supprimer le motif de G ,
- Reconnecter R avec le contexte plus global du graphe cible.

How to map graphs? (from [König 18])

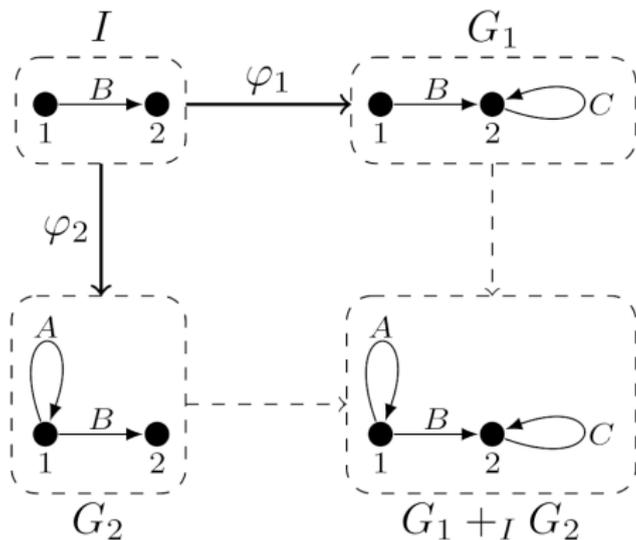
Graph **morphism** :



- Functions on nodes and arcs that preserve structure.

How to glue graphs? (from [König 18])

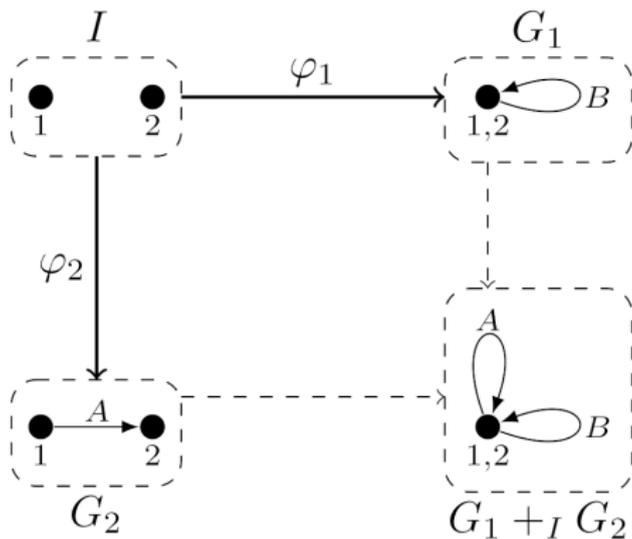
Graph **gluing** :



► \sim Quotiented disjoint union.

How to glue graphs? (from [König 18])

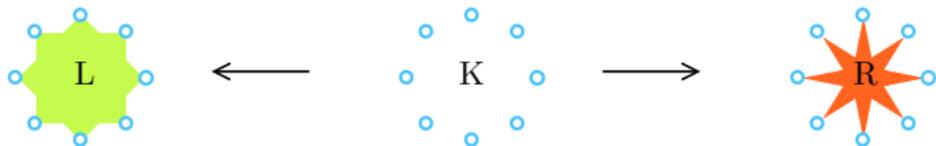
Graph **gluing** :



► \sim Quotiented disjoint union.

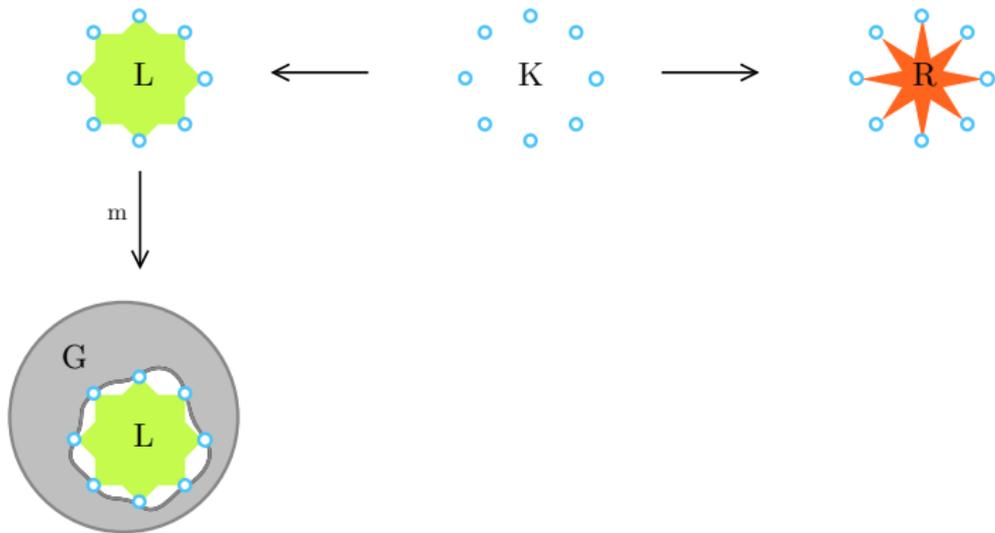
Back to graph transformation rules

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



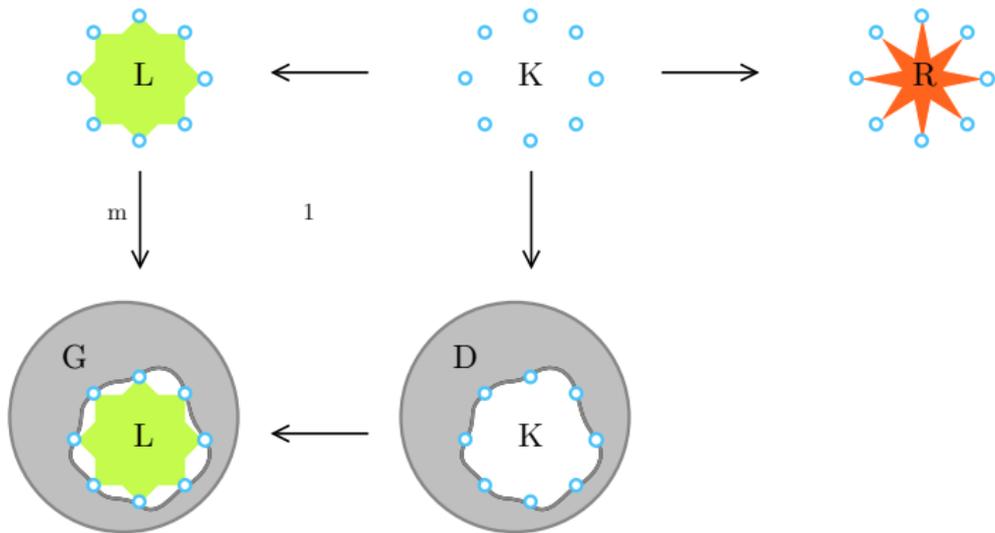
Back to graph transformation rules

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



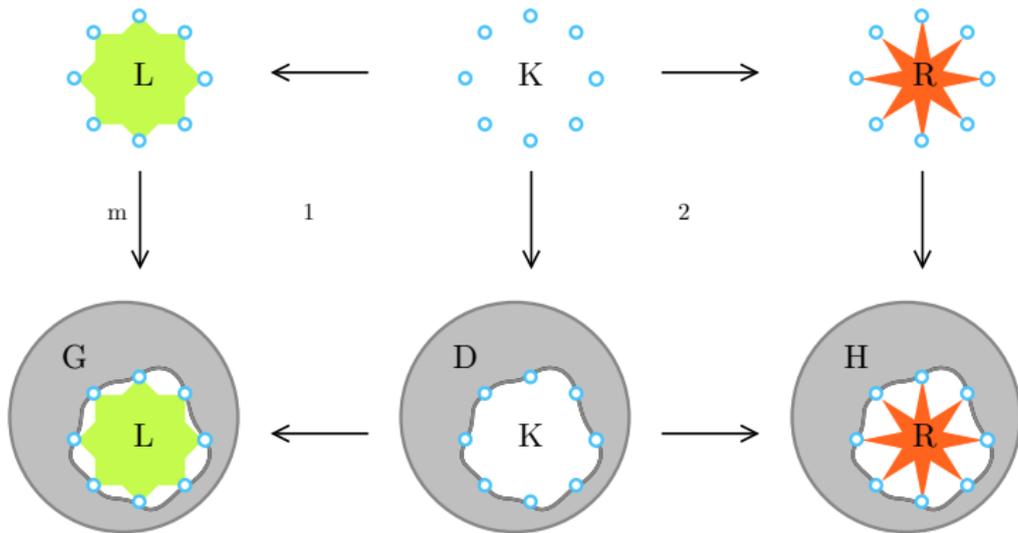
Back to graph transformation rules

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

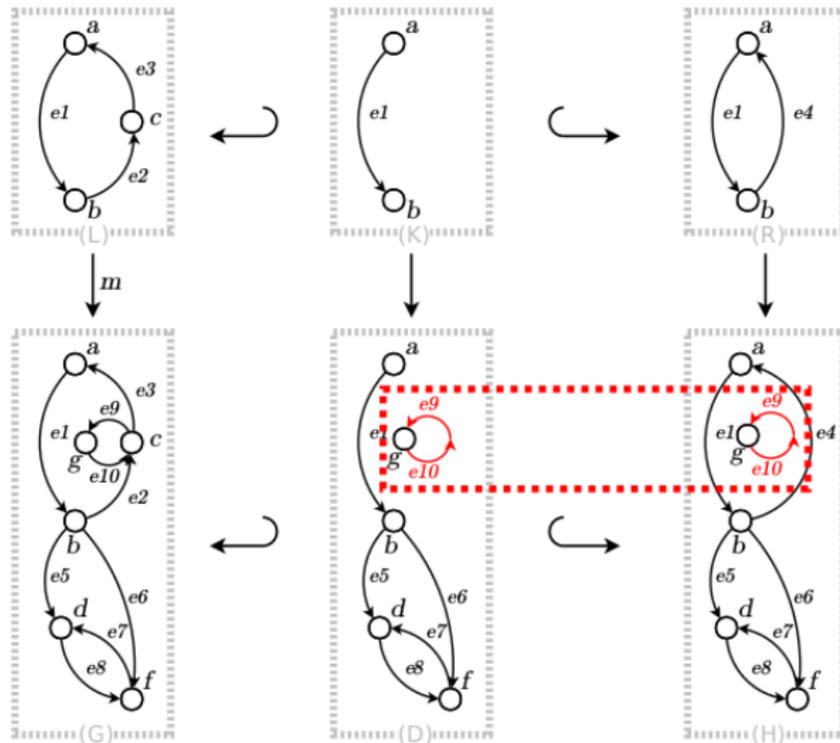


Back to graph transformation rules

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

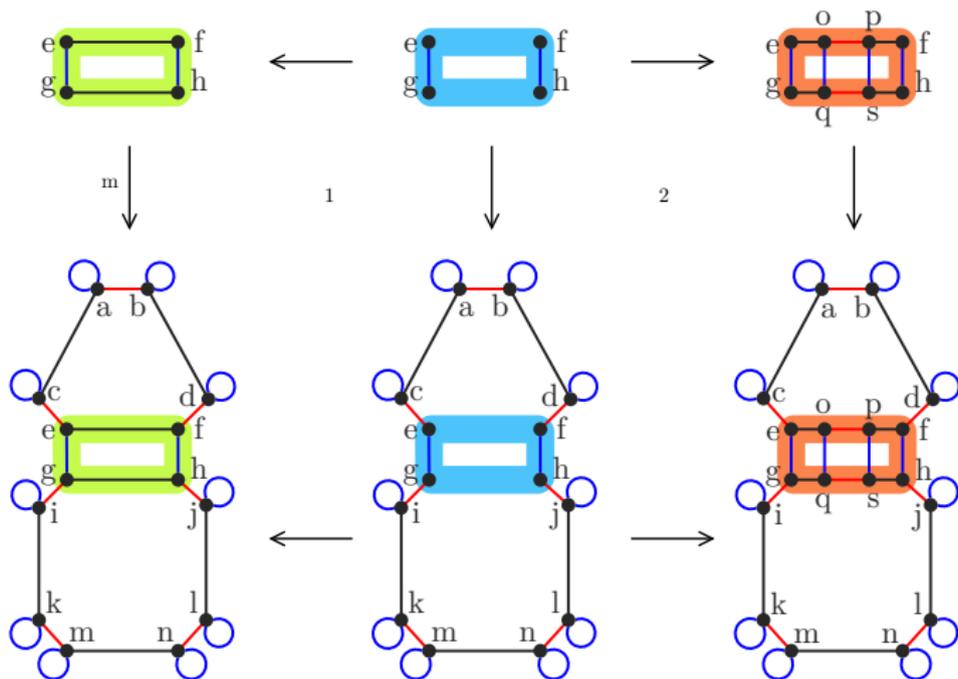


Why does it has to be so complicated?



Rewriting G-maps

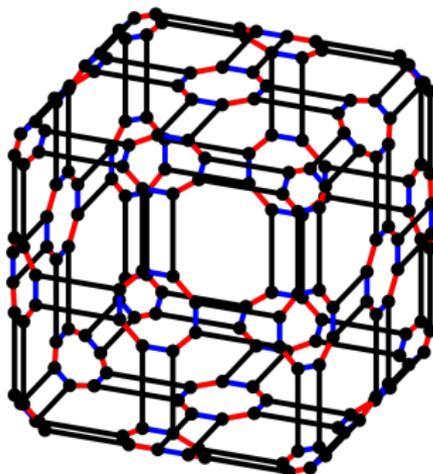
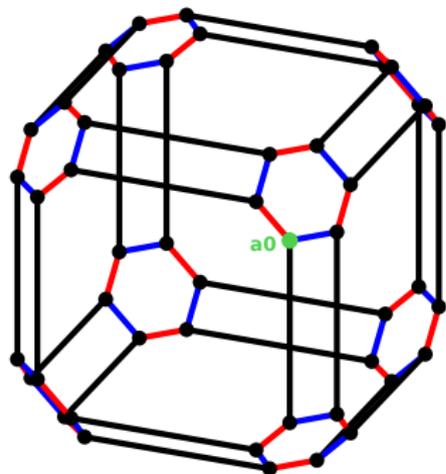
- Most conservative framework (all morphisms are injectives).



Spécification de l'algorithme

Algorithme :

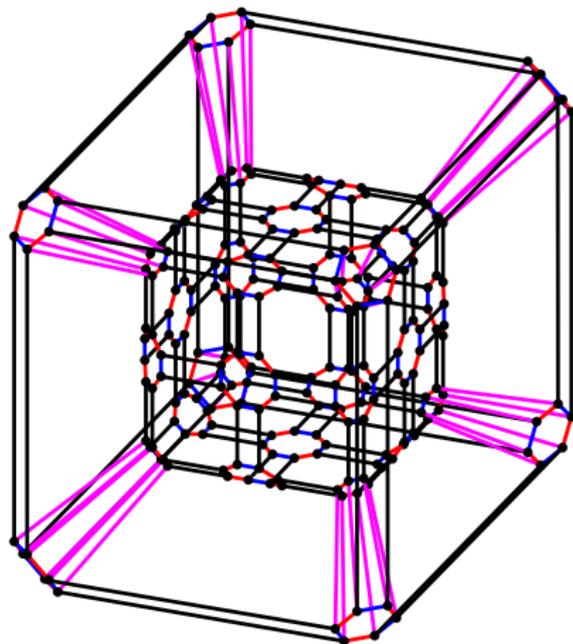
- ▶ **Entrée** : G-carte **avant**, G-carte **après**, **brin** de la G-carte avant et type d'orbite.
- ▶ **Sortie** : **Schéma de règles** pour l'opération correspondante.



- Brin a_0
- Type d'orbite $\langle 1, 2 \rangle$

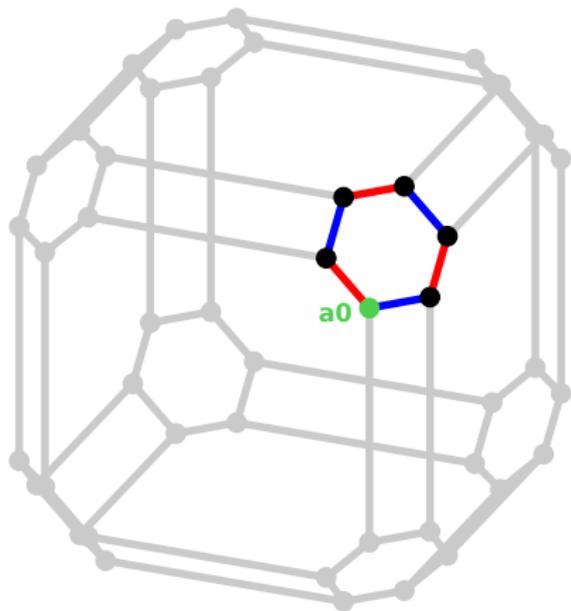
Construction de l'interface (éléments préservés)

- Union de l'instance "avant" et de l'instance "après" à l'aide d'arcs reliant les brins préservés.



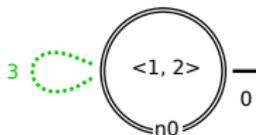
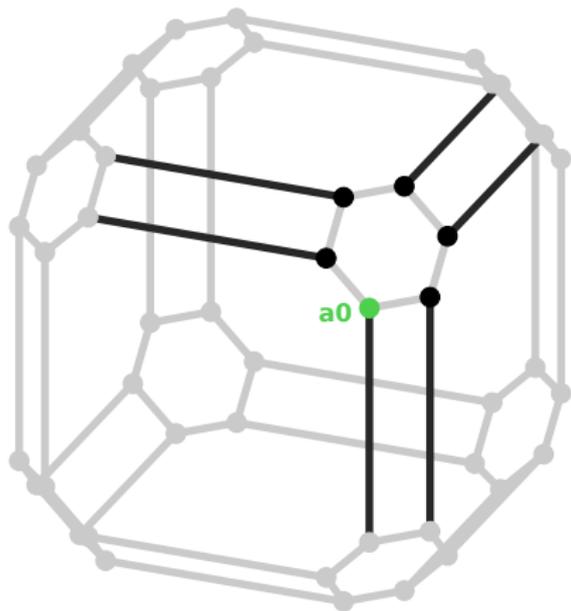
Détails de l'algorithme

Construction d'un nœud, cas de l'ancre avec l'orbite $\langle 1, 2 \rangle$.



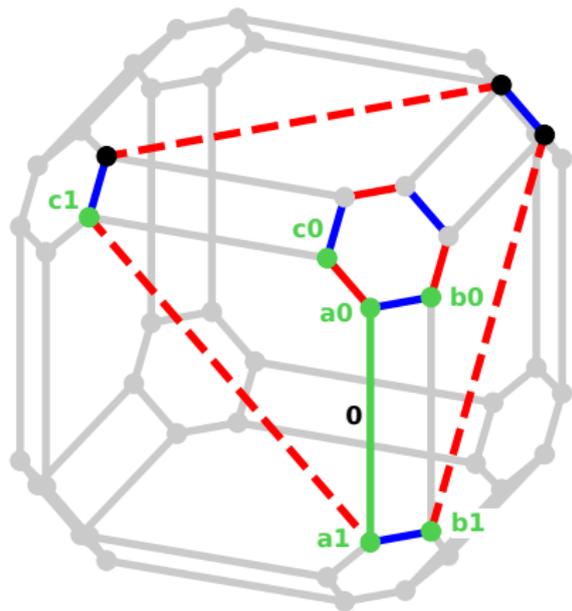
Détails de l'algorithme

Extension des arcs.



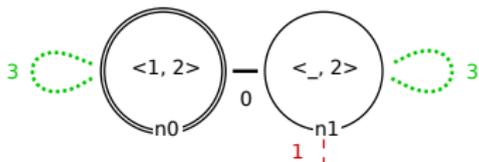
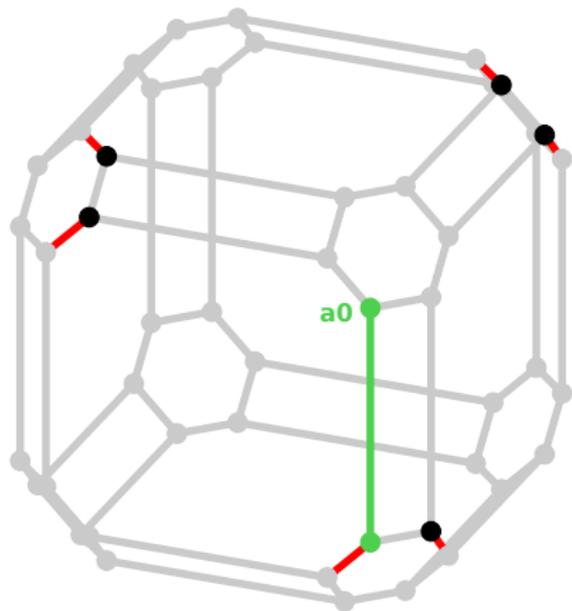
Détails de l'algorithme

Construction d'un nœud.



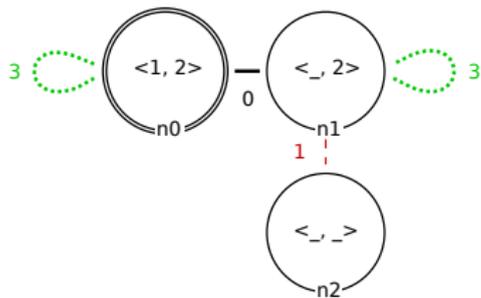
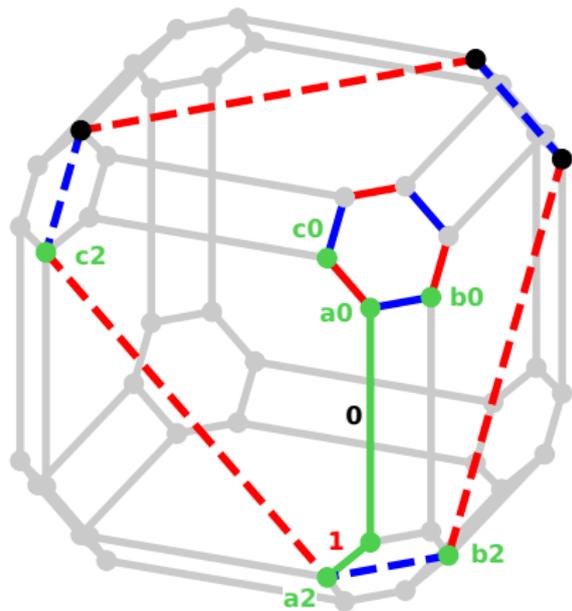
Détails de l'algorithme

Extension des arcs.



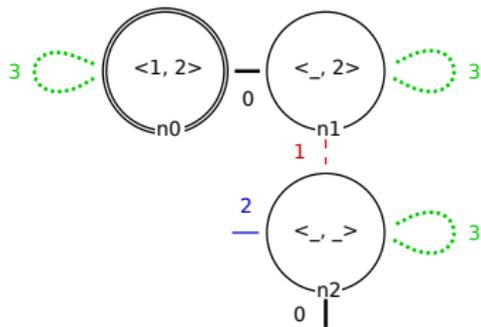
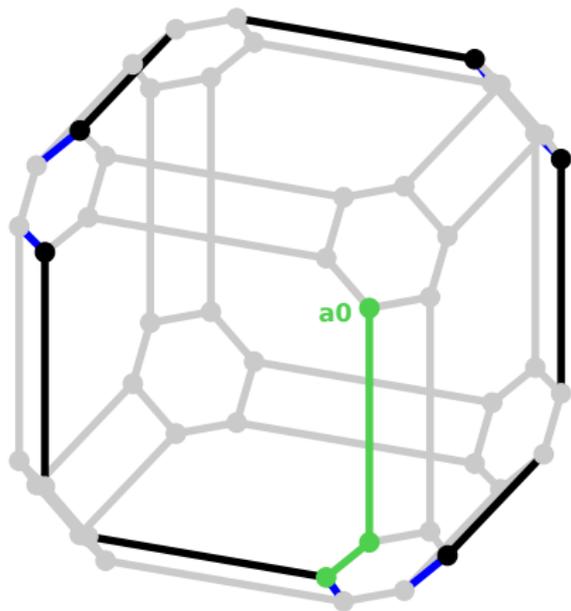
Détails de l'algorithme

Construction d'un nœud.



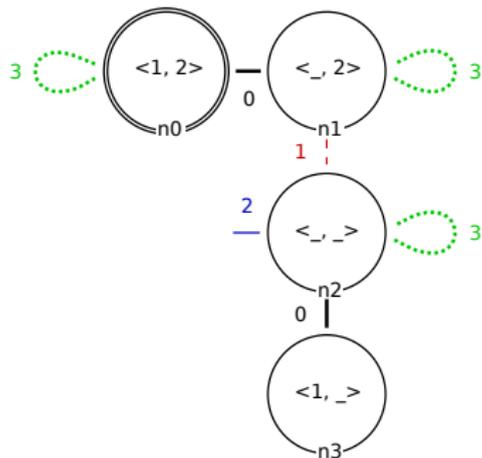
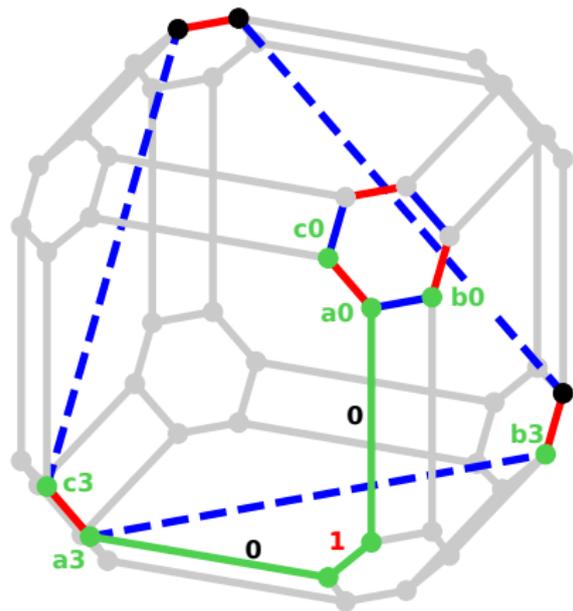
Détails de l'algorithme

Extension des arcs.



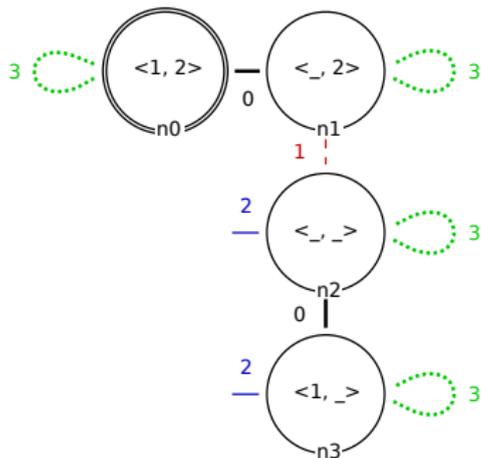
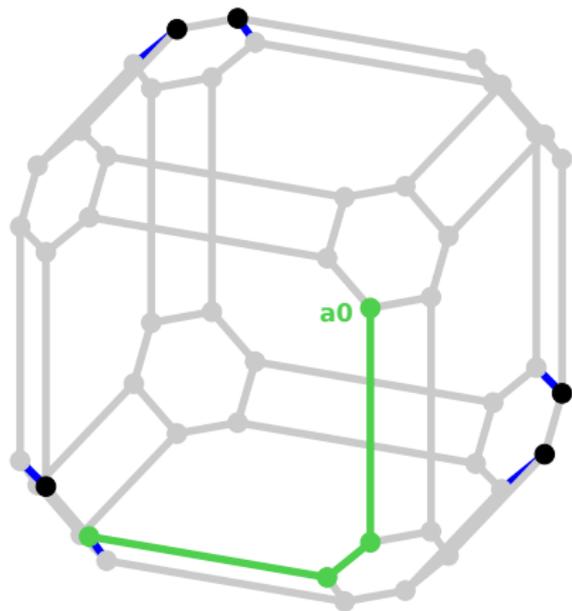
Détails de l'algorithme

Construction d'un nœud.



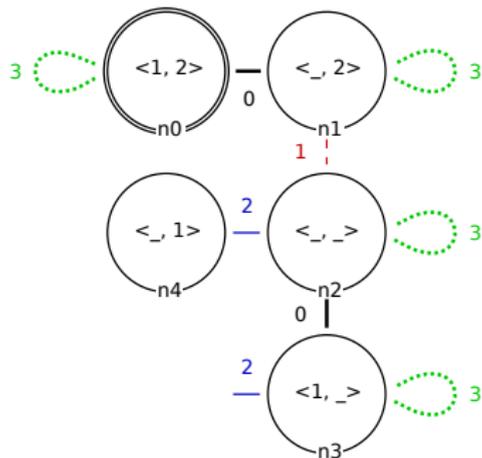
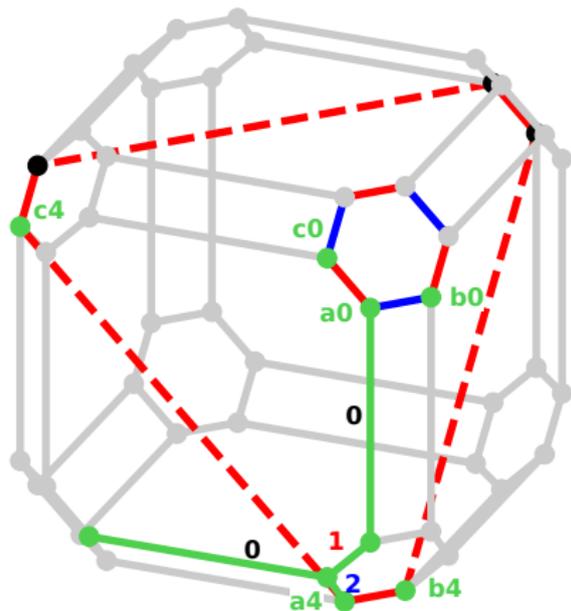
Détails de l'algorithme

Extension des arcs.



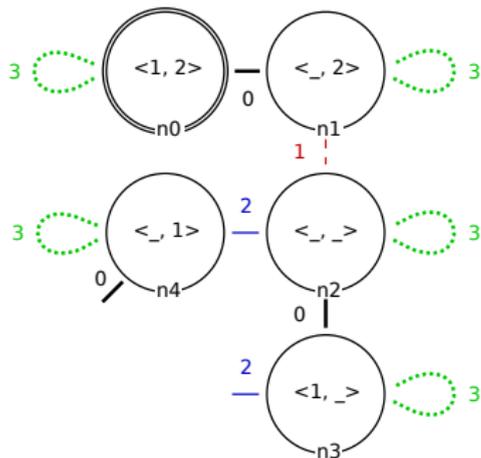
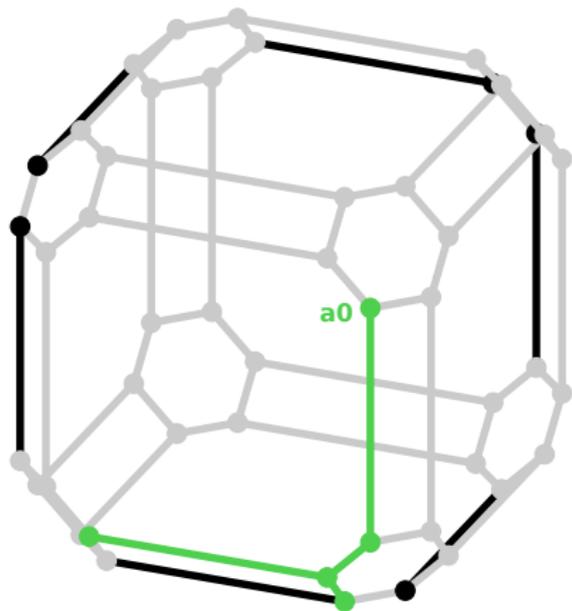
Détails de l'algorithme

Construction d'un nœud.



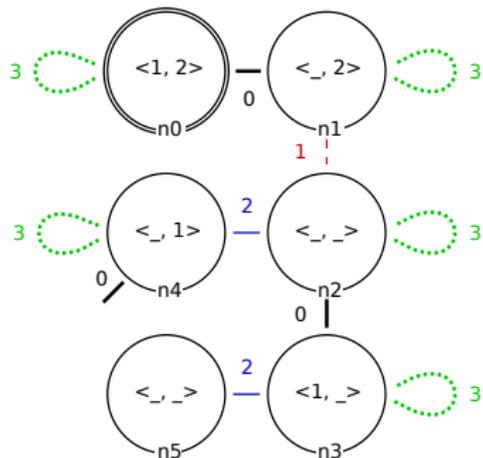
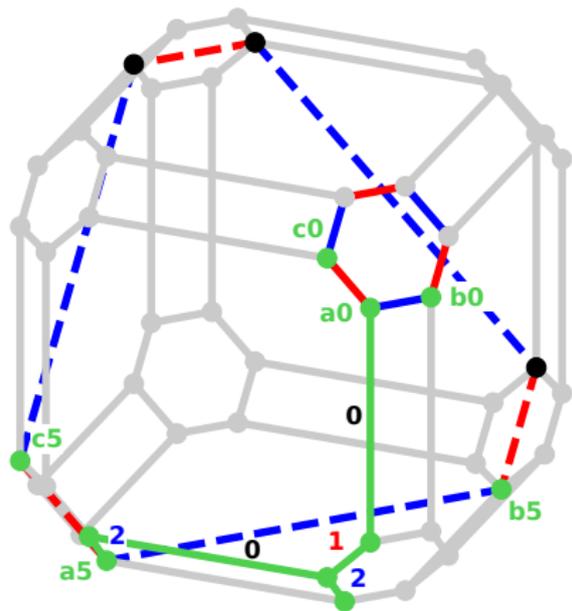
Détails de l'algorithme

Extension des arcs.



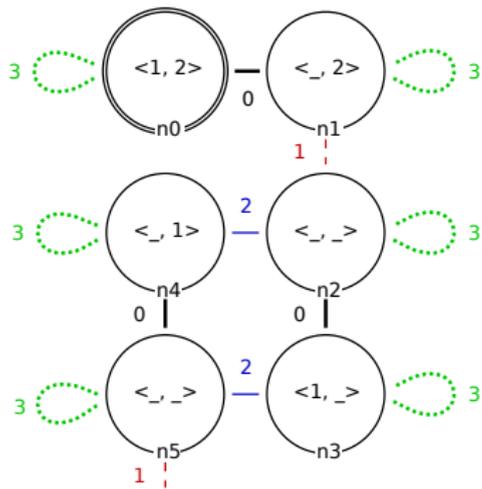
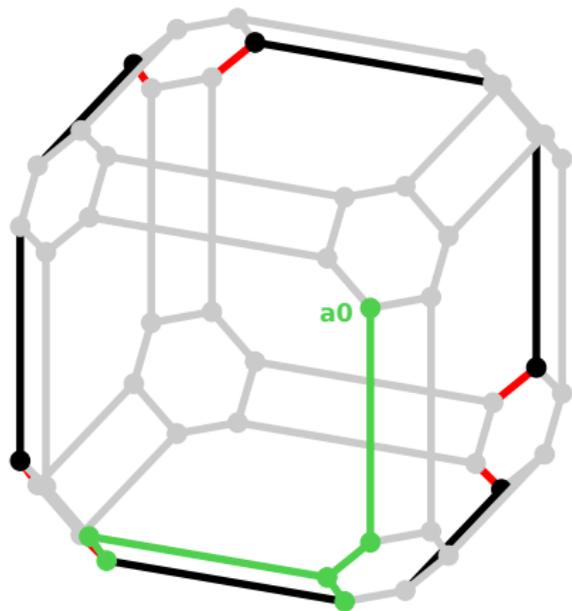
Détails de l'algorithme

Construction d'un nœud.



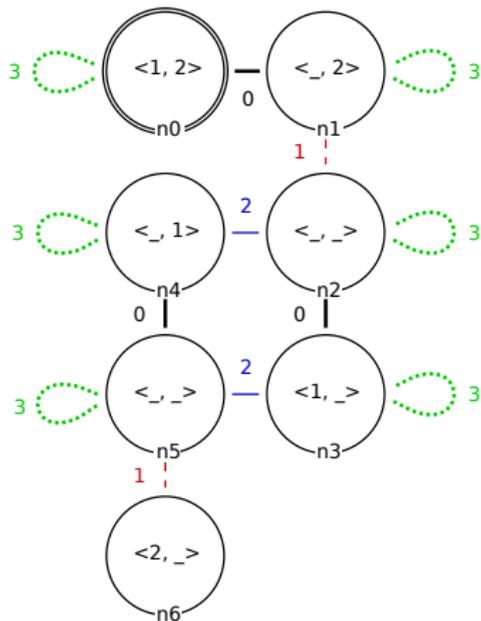
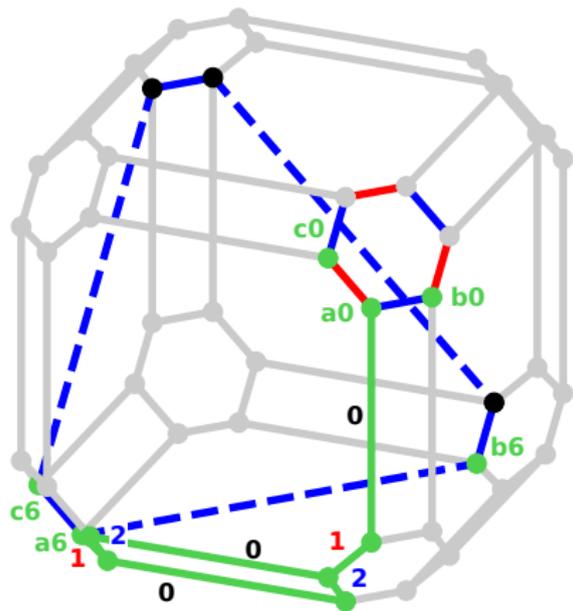
Détails de l'algorithme

Extension des arcs.



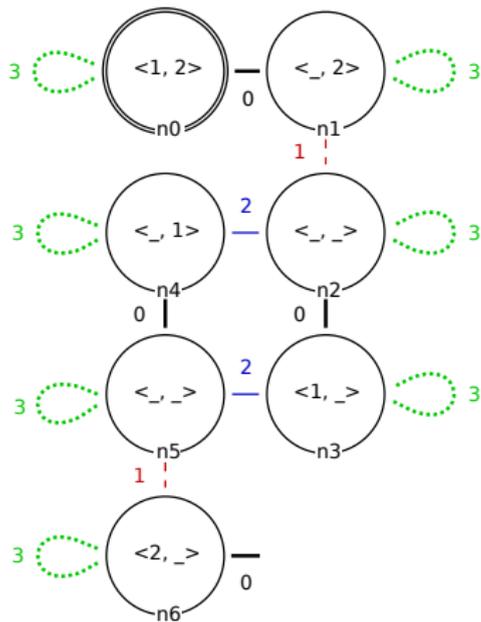
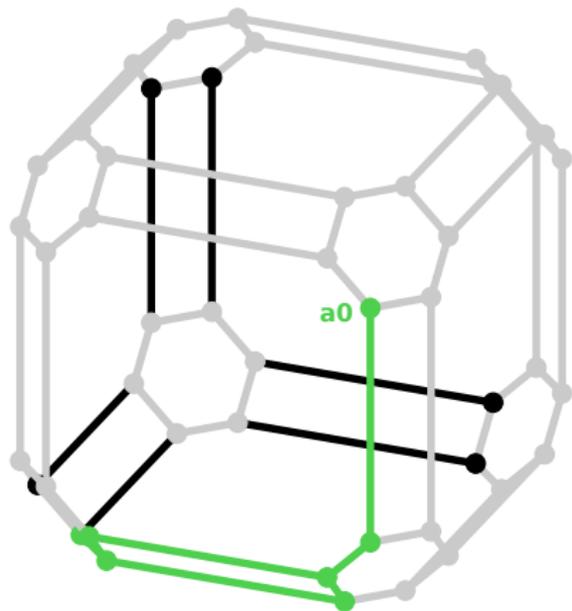
Détails de l'algorithme

Construction d'un nœud.



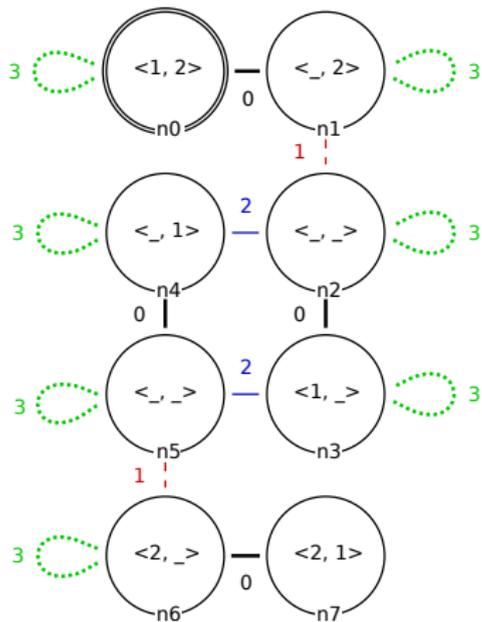
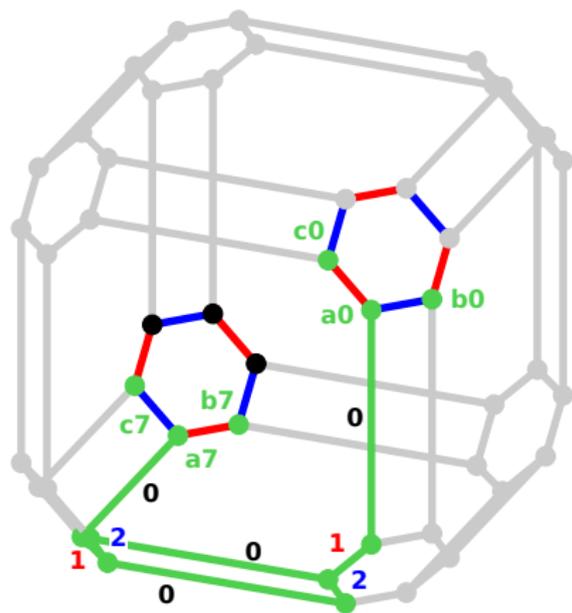
Détails de l'algorithme

Extension des arcs.



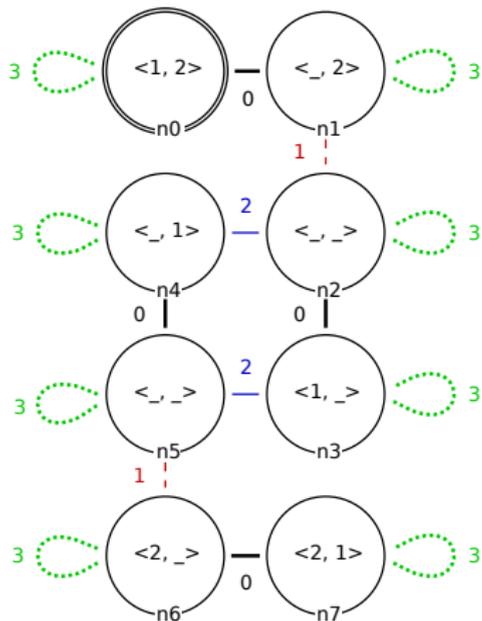
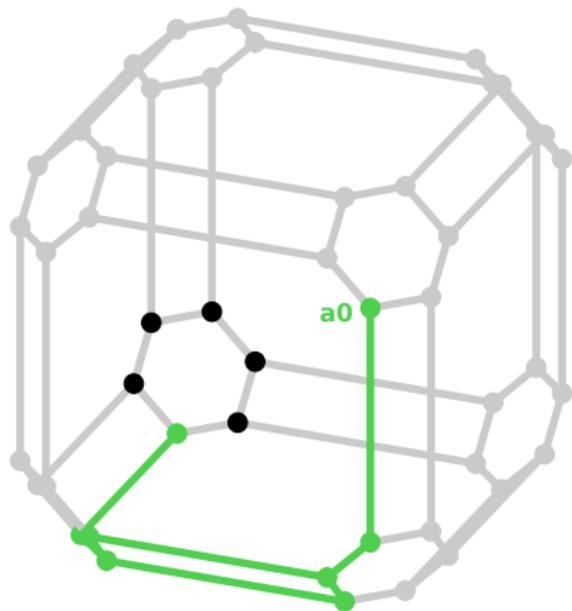
Détails de l'algorithme

Construction d'un nœud.



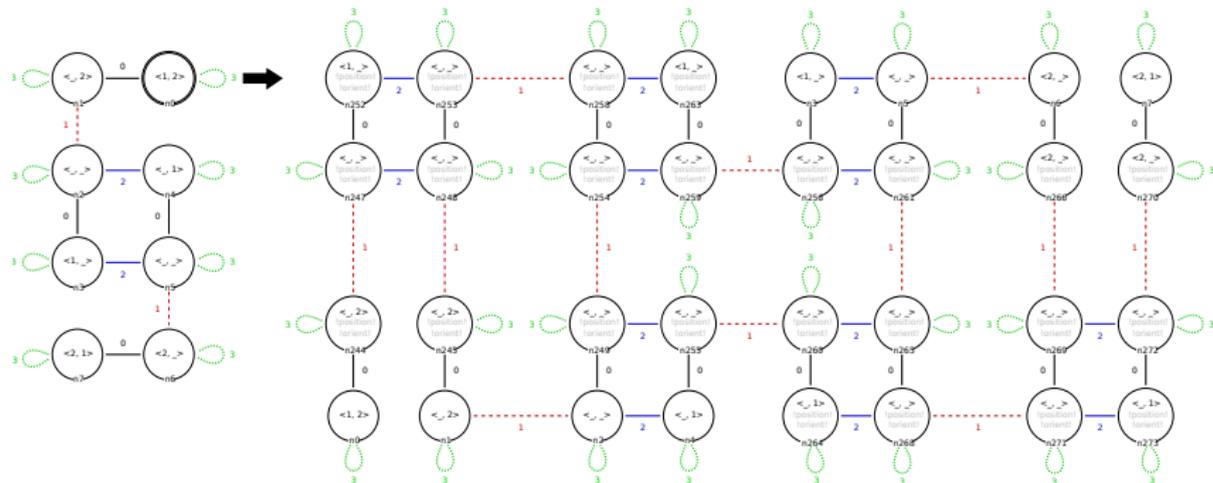
Détails de l'algorithme

Extension des arcs.



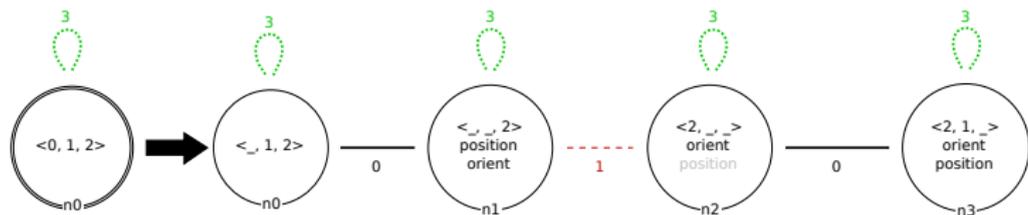
Résultat de l'algorithme

► Schéma de règles à partir d'un sommet du cube :

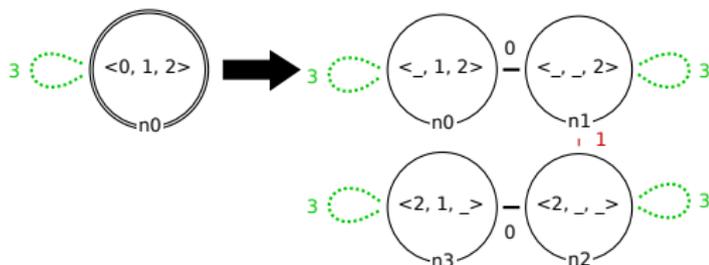


Résultat de l'algorithme

► Schéma de règles utilisé :

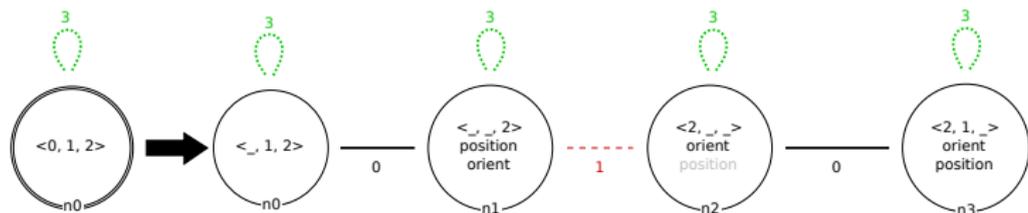


► Parmi les schémas inférés, on retrouve le schéma utilisé :

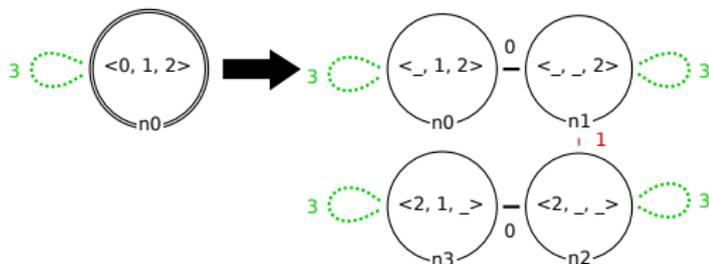


Résultat de l'algorithme

► Schéma de règles utilisé :



► Parmi les schémas inférés, on retrouve le schéma utilisé :



- 768 schémas possibles.
- 14 schémas distincts (symétrie du cube).
- 48 schémas construits.