

Transformations de graphes décorés

Application aux opérations de modélisation géométrique

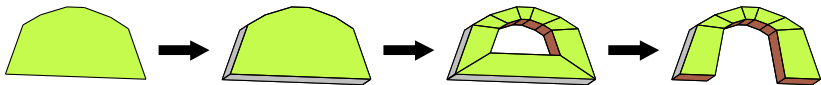
Pascale Le Gall¹ Romain Pascual¹
Hakim Belhaouari² Agnès Arnould²

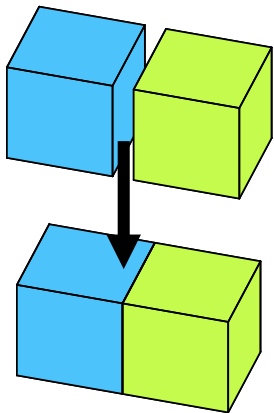
¹ Laboratoire MICS, CentraleSupélec, Université Paris-Saclay

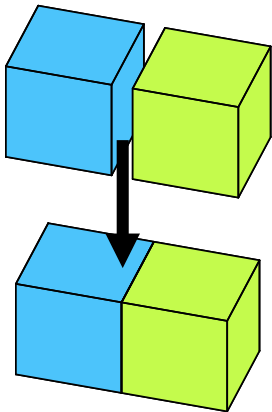
² Laboratoire XLIM, Université de Poitiers

22 juin 2023





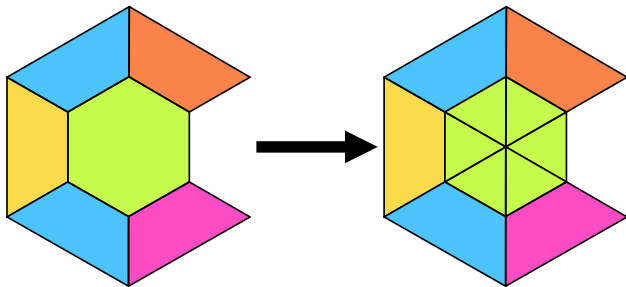




CGAL's sew operation

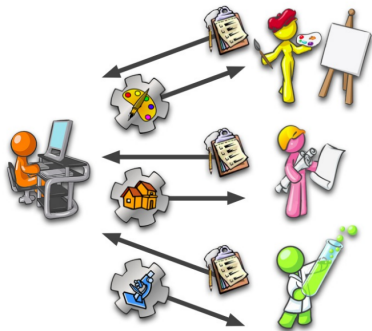
```
template<unsigned int i>
void sew(Dart_descriptor adart1, Dart_descriptor adart2)

CGAL_assertion( i<=dimension );
CGAL_assertion( (is_sevable<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);
}
```

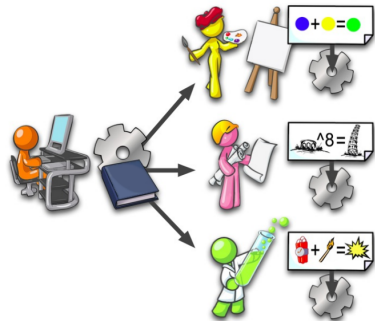


Ambition : définir un *domain-specific language* (DSL) pour la modélisation géométrique

Motivations : abstraction, performance, concision, correction, simplicité

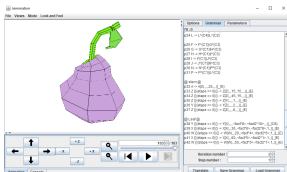


Approche Classique

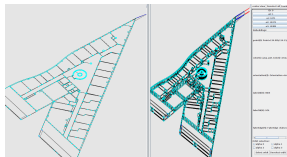


Jerboa¹

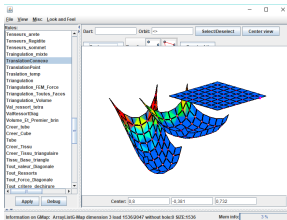
Croissance de plantes



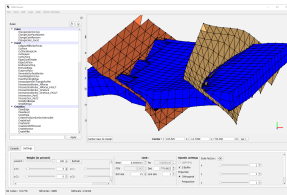
Architecture



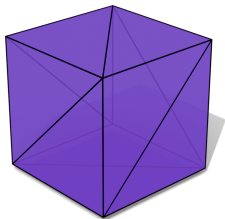
Simulation physique



Géologie

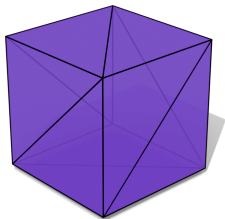


Topologie vs géométrie

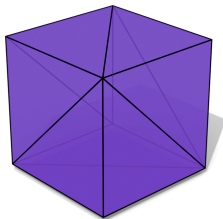


Objet

Topologie vs géométrie

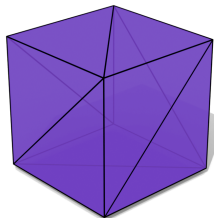


Objet

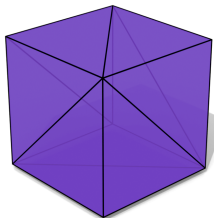


Topologie différente

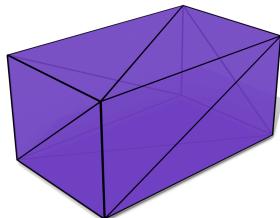
Topologie vs géométrie



Objet



Topologie différente



Géométrie différente

Topologie

Cartes généralisées

Opérations topologiques

Géométrie

Plongements

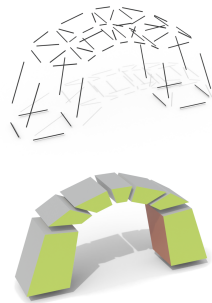
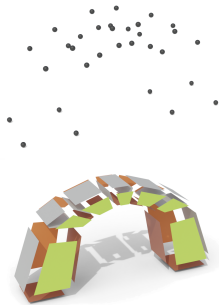
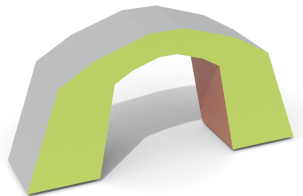
Calculs

Jerboa

Topologie

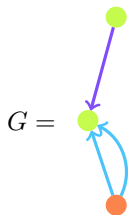
- ▶ Représentation et modification de la structure des objets

Cellules topologiques

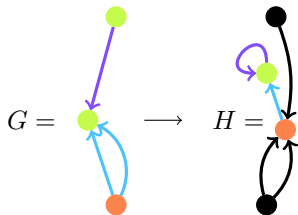


Catégorie(s) des graphes

Graphe

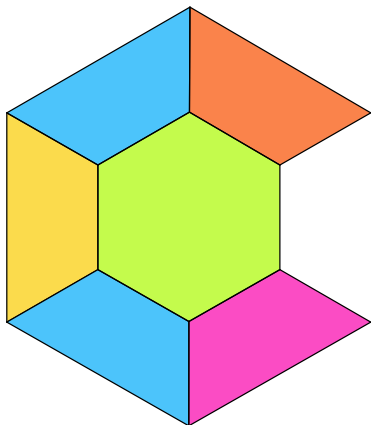


Morphisme de graphes



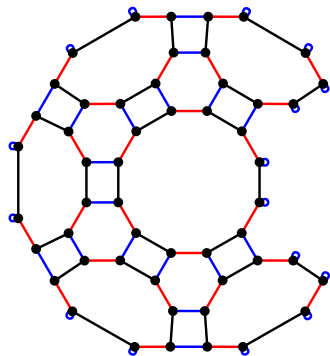
Forme une catégorie *adhésive*

Cartes généralisées ¹



1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées ¹

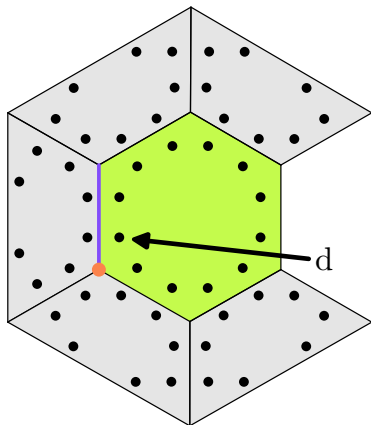


Représentation par graphes

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées ¹



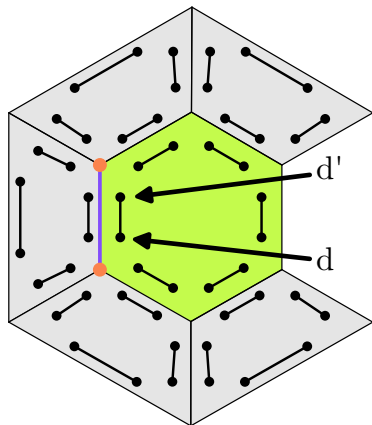
Représentation par graphes

Brin d identifie
sommet orange
arête violette
face verte

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées ¹



Représentation par graphes

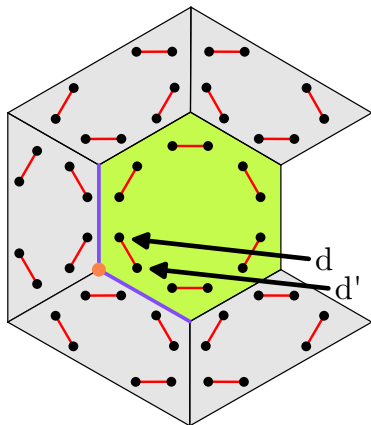
0-arc

- ▶ sommets distincts
- ▶ mêmes arêtes et faces

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées ¹



Représentation par graphes

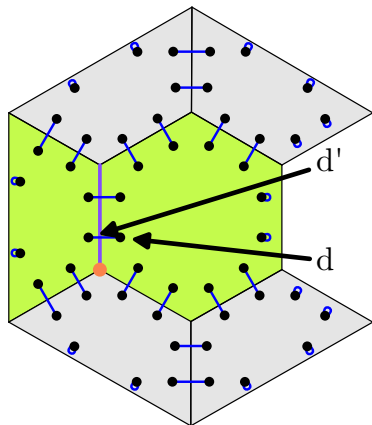
1-arc

- ▶ arêtes distinctes
- ▶ mêmes sommets et faces

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées ¹



Représentation par graphes

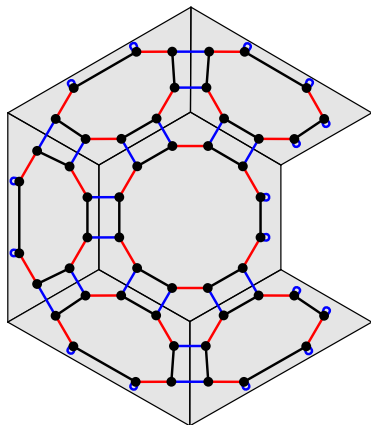
2-arc

- ▶ faces distinctes
- ▶ mêmes sommets et arêtes

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées ¹



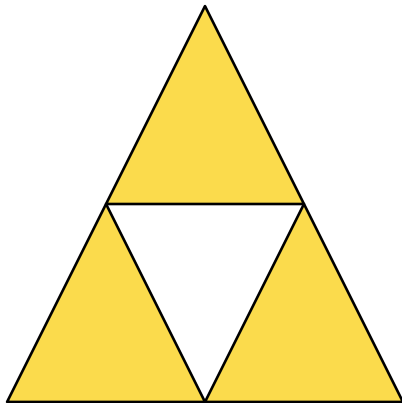
Superposition des graphes

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

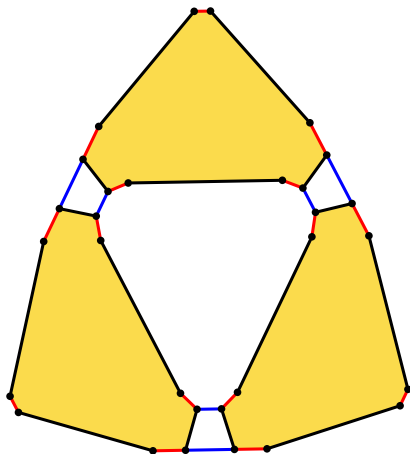
Exercice

Donner la carte généralisée associée à l'objet suivant



Exercice

Donner la carte généralisée associée à l'objet suivant



Question : Comment décrire formellement une carte généralisée (sous forme de graphe) ?

Question : Comment décrire formellement une carte généralisée (sous forme de graphe) ?

Pour rappel, voici la définition combinatoire d'une carte généralisée (cours de Samuel Peltier)

Une n -Gcarte ($n \geq 0$), est un $(n + 2)$ -tuple $(B, \alpha_0, \dots, \alpha_n)$ tel que :

- ▶ B est un ensemble fini de brins
- ▶ $\forall i \in 0..n, \alpha_i$ est une involution sur B
- ▶ $\forall i \in 0..(n - 2), \forall j \in (i + 2)..n, \alpha_i \alpha_j$ est une involution sur B

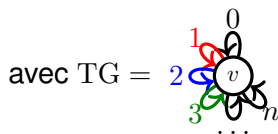
Graphes topologiques

Graphes topologiques : arcs typés par une dimension de $0..n$

Graphes topologiques

Graphes topologiques : arcs typés par une dimension de $0..n$

n -Graph := Graph / TG



Contraintes topologiques

Gcartes : n -Graph avec contraintes topologiques

Contraintes topologiques

Gcartes : n -Graph avec contraintes topologiques

non-orientation chaque arc admet un unique inverse

Contraintes topologiques

Gcartes : n -Graph avec contraintes topologiques

non-orientation chaque arc admet un unique inverse

arcs incidents chaque nœud est incident à un unique arc par
dimension $i \in 0..n$

Contraintes topologiques

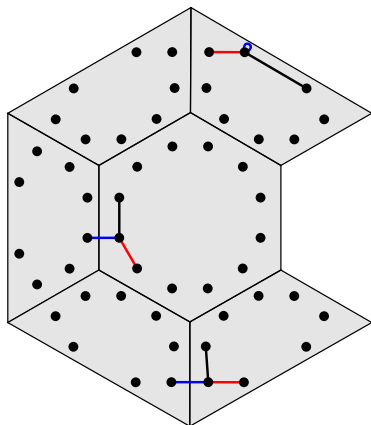
Gcartes : n -Graph avec contraintes topologiques

non-orientation chaque arc admet un unique inverse

arcs incidents chaque nœud est incident à un unique arc par dimension $i \in 0..n$

cycles pour i, j tels que $i + 2 \leq j$, tout chemin $ijij$ est un cycle

Contrainte d'arcs incidents



chaque nœud est incident à un
unique arc par dimension

$$i \in 0..n$$

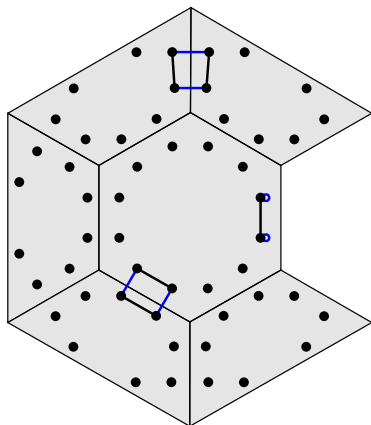
Traduction de α_i **permutation**

→ cours de Samuel Peltier

involution car graphe
non-orienté

Légende : 0, 1, 2

Contrainte de cycle



pour i, j tels que $i + 2 \leq j$, tout chemin $ijij$ est un cycle

Traduction de $\alpha_i \circ \alpha_j$

permutation

→ cours de Samuel Peltier

involution car graphe
non-orienté

Légende : 0, 1, 2

Digression sur les contraintes

Contraintes exprimées dans le ‘langage courant’

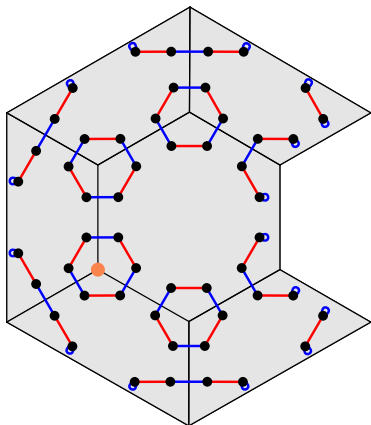
Reformulation possible en logique du premier ordre, à la Courcelle¹

Représentation graphique des contraintes par ‘nested conditions’²

1. COURCELLE 1997

2. HABEL et al. 2009 ; EHRIG et al. 2014 ; BEHR et al. 2021

Orbites et cellules topologiques

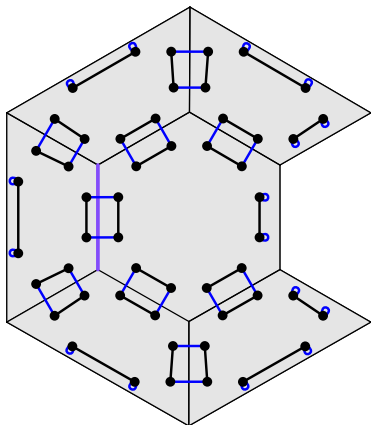


Orbite : Sous-graphe induit par un sous-ensemble $\langle o \rangle$ de dimensions

► Sommets : orbites $\langle 1, 2 \rangle$

Légende : 0, 1, 2

Orbites et cellules topologiques

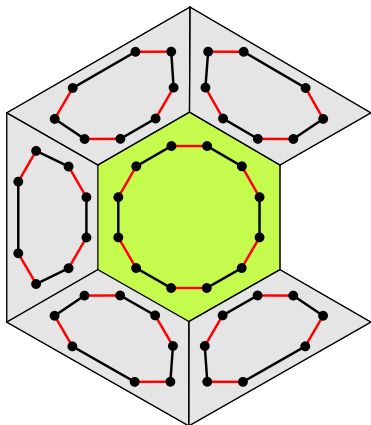


Orbite : Sous-graphe induit par un sous-ensemble $\langle o \rangle$ de dimensions

- ▶ Sommets : orbites $\langle 1, 2 \rangle$
- ▶ Arêtes : orbites $\langle 0, 2 \rangle$

Légende : 0, 1, 2

Orbites et cellules topologiques

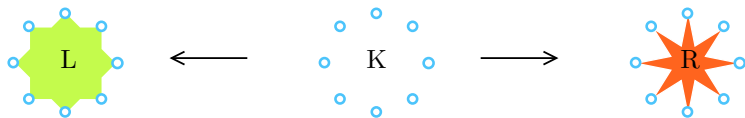


Orbite : Sous-graphe induit par un sous-ensemble $\langle o \rangle$ de dimensions

- ▶ Sommets : orbites $\langle 1, 2 \rangle$
- ▶ Arêtes : orbites $\langle 0, 2 \rangle$
- ▶ Faces : orbites $\langle 0, 1 \rangle$

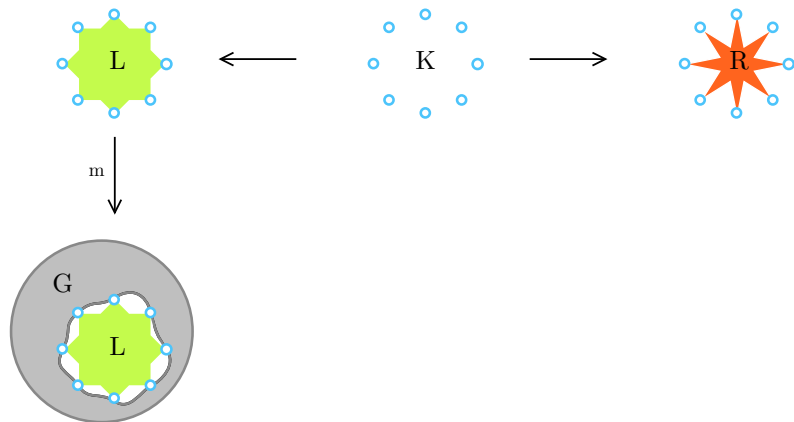
Légende : 0, 1, 2

Rappel transformation de graphes ¹



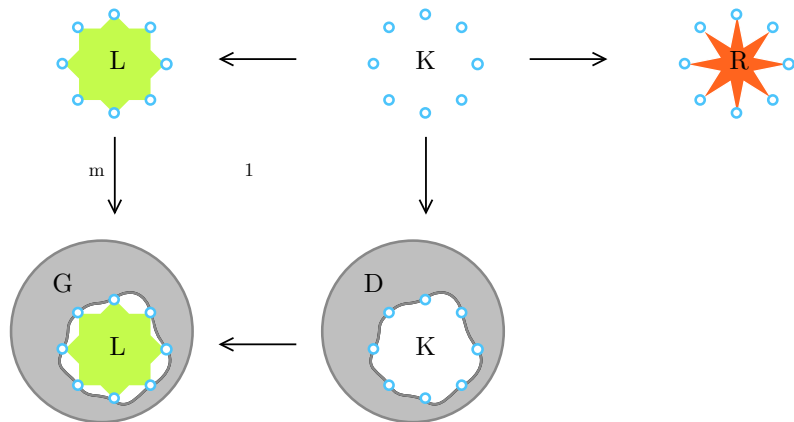
1. ROZENBERG 1997; EHRIG et al. 2006; HECKEL et al. 2020

Rappel transformation de graphes ¹



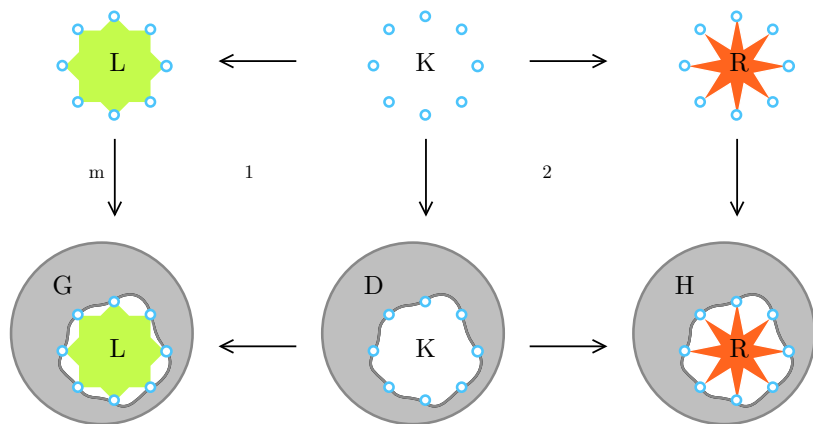
1. ROZENBERG 1997; EHRIG et al. 2006; HECKEL et al. 2020

Rappel transformation de graphes ¹



1. ROZENBERG 1997 ; EHRIG et al. 2006 ; HECKEL et al. 2020

Rappel transformation de graphes ¹



1. ROZENBERG 1997; EHRIG et al. 2006; HECKEL et al. 2020

Rappel sur la réécriture par DPO

Règle $p = L \xleftarrow{l} K \xrightarrow{r} R$, match $m: L \hookrightarrow G$

Alors $G \Rightarrow^{r,m} H$ si

$$\begin{array}{ccccc} L & \xleftarrow{l} & K & \xrightarrow{r} & R \\ m \downarrow & & \downarrow & & \downarrow m' \\ & PO & & PO & \\ G & \longleftarrow & D & \longrightarrow & H \end{array}$$

existe

Opération de modélisation géométrique

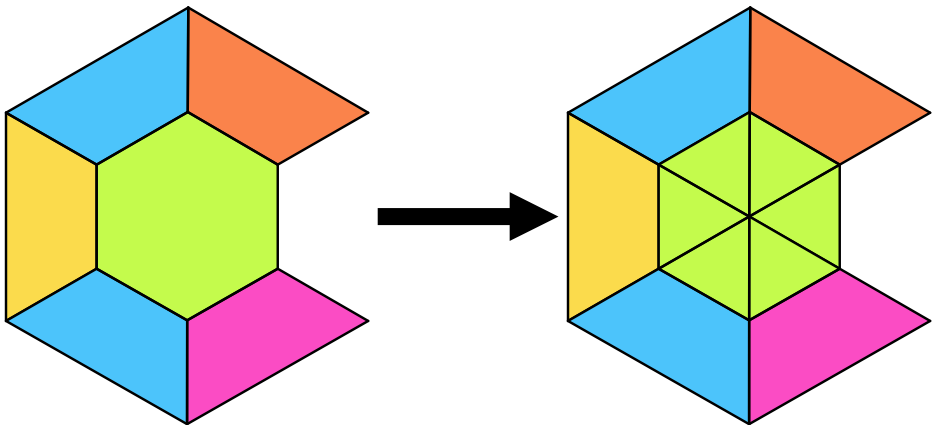
Opération de modélisation géométrique

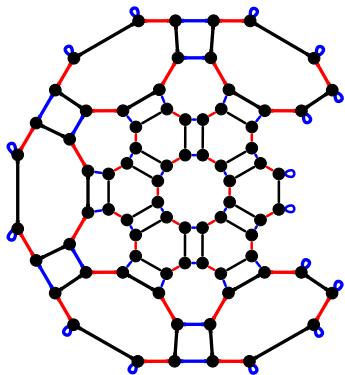
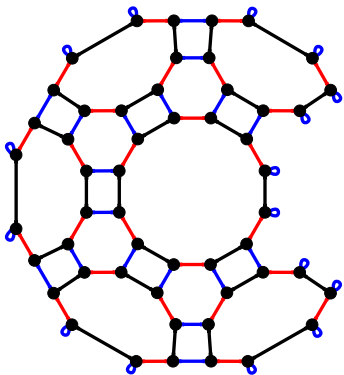
$$G \Rightarrow^{r,m} G'$$

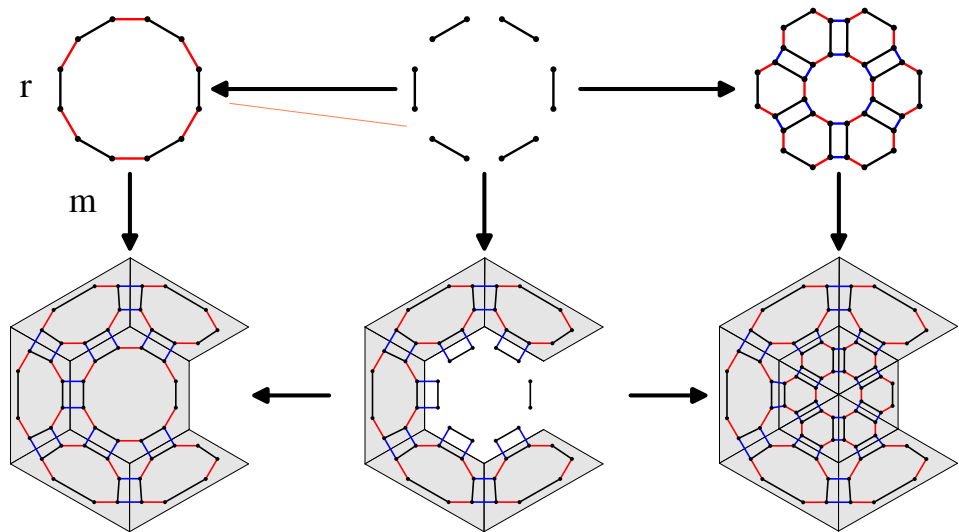
avec

- ▶ r règle de réécriture de graphes
- ▶ m match

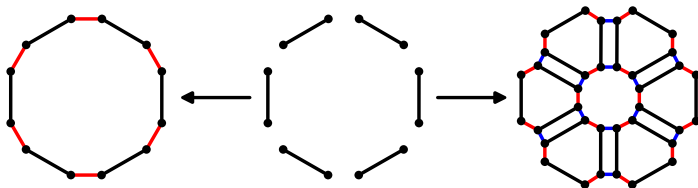
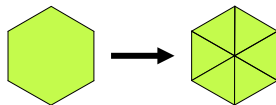
où G et G' sont des Gcartes bien formées



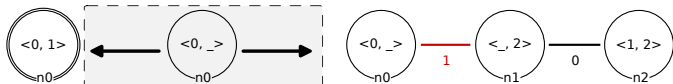
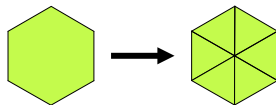




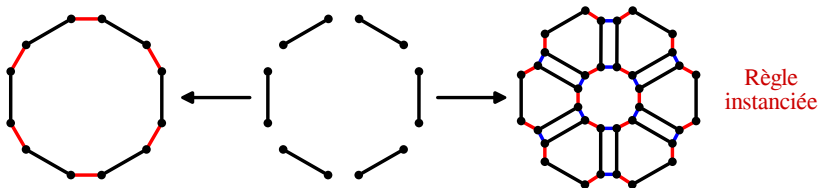
Réécriture d'orbites



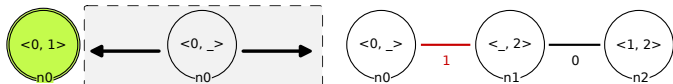
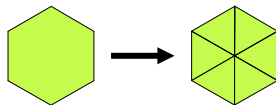
Réécriture d'orbites



Implicitement
calculée



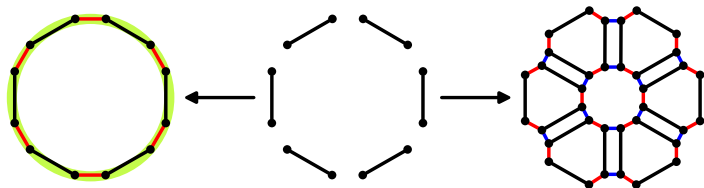
Réécriture d'orbites



Implicitement
calculée

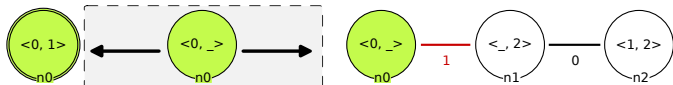
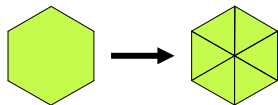


Local



Règle
instanciée

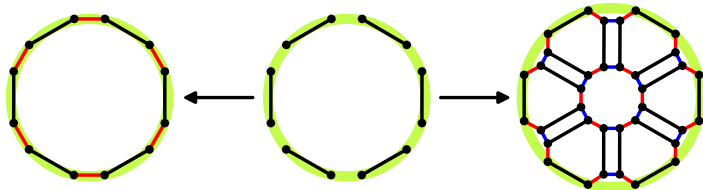
Réécriture d'orbites



Implicitement
calculée

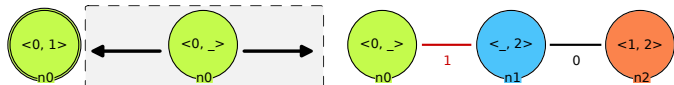
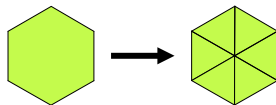


Local

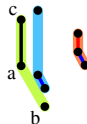


Règle
instanciée

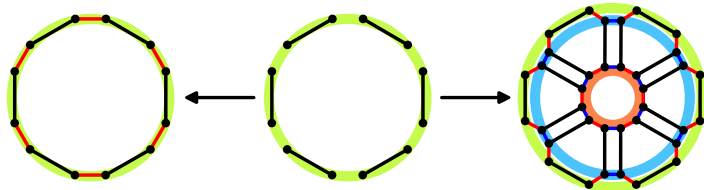
Réécriture d'orbites



Implicitement
calculée

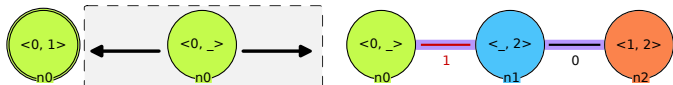
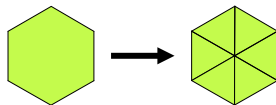


Local

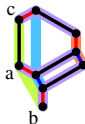


Règle
instanciée

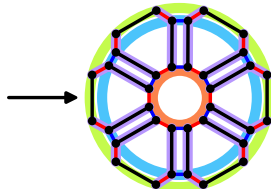
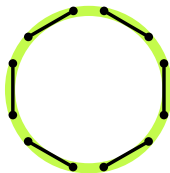
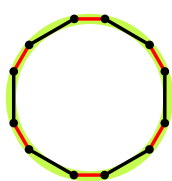
Réécriture d'orbites



Implicitement
calculée

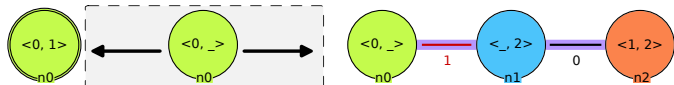
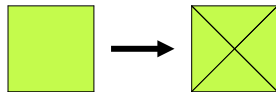


Local

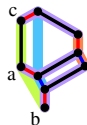


Règle
instanciée

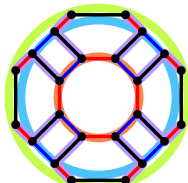
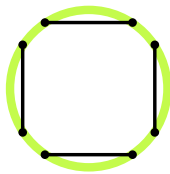
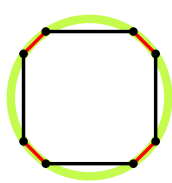
Réécriture d'orbites



Implicitement
calculée

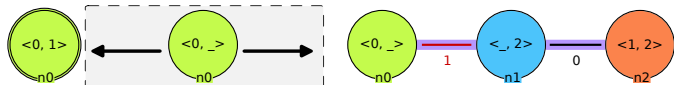
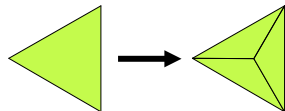


Local

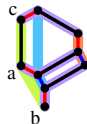


Règle
instanciée

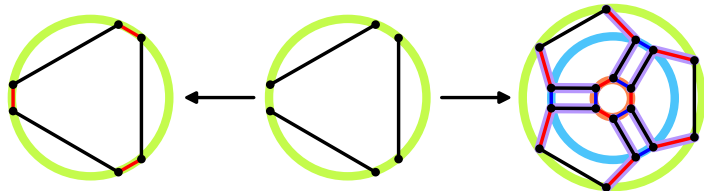
Réécriture d'orbites



Implicitement
calculée

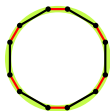
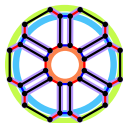


Local



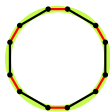
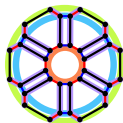
Règle
instanciée

Produits de graphes ¹



1. inspiré de BAUDERON 1995

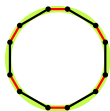
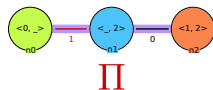
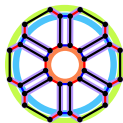
Produits de graphes ¹



P

1. inspiré de BAUDERON 1995

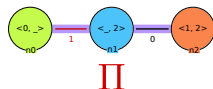
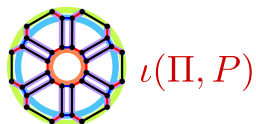
Produits de graphes ¹



P

1. inspiré de BAUDERON 1995

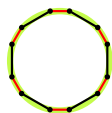
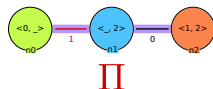
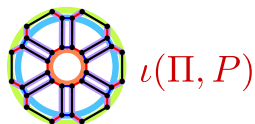
Produits de graphes ¹



► $\iota(\Pi, P)$: instantiation

1. inspiré de BAUDERON 1995

Produits de graphes ¹

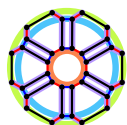


$$P \xrightarrow{\mathbb{E}_\Sigma} \mathbb{E}_\Sigma(P)$$

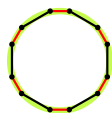
- ▶ $\iota(\Pi, P)$: instantiation
- ▶ \mathbb{E}_Σ : foncteur de plongement

1. inspiré de BAUDERON 1995

Produits de graphes ¹



$\iota(\Pi, P)$



$$P \xrightarrow{\mathbb{E}_\Sigma} \mathbb{E}_\Sigma(P) \xrightarrow{!\mathbb{E}_\Sigma(P)} 1_{\Sigma^2}$$

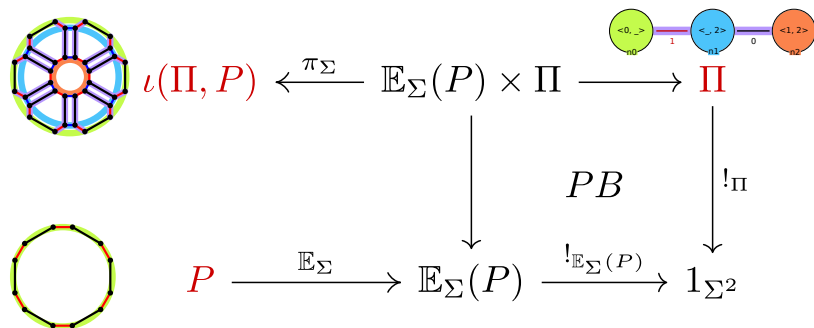
$$\begin{array}{ccc} \mathbb{E}_\Sigma(P) \times \Pi & \longrightarrow & \Pi \\ \downarrow & & \downarrow !\Pi \\ \mathbb{E}_\Sigma(P) & \xrightarrow{PB} & 1_{\Sigma^2} \end{array}$$



- ▶ $\iota(\Pi, P)$: instantiation
- ▶ \mathbb{E}_Σ : foncteur de plongement

1. inspiré de BAUDERON 1995

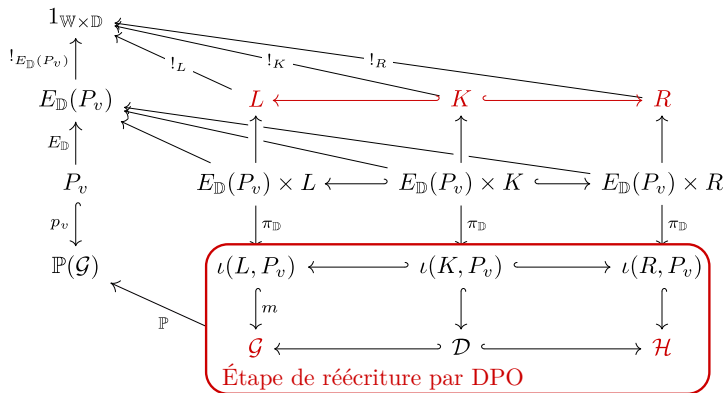
Produits de graphes ¹



- ▶ $\iota(\Pi, P)$: instantiation
- ▶ \mathbb{E}_Σ : foncteur de plongement
- ▶ π_Σ : foncteur de projection

1. inspiré de BAUDERON 1995

Construction complète



Préservation de la consistance

Modification d'un objet bien formé vers un objet bien formé

~ correction de typage d'un programme

Préservation de la consistance

Modification d'un objet bien formé vers un objet bien formé

~ correction de typage d'un programme

Soit une contrainte c , une règle $r = L \leftrightarrow K \hookrightarrow R$ et un morphisme $m: L \hookrightarrow G$.

Si $G \models c$, comment assurer que $H \models c$ avec $H := G \Rightarrow^{r,m} H$?

Préservation de la consistance

Modification d'un objet bien formé vers un objet bien formé

~ correction de typage d'un programme

Soit une contrainte c , une règle $r = L \leftrightarrow K \hookrightarrow R$ et un morphisme $m: L \hookrightarrow G$.

Si $G \models c$, comment assurer que $H \models c$ avec $H := G \Rightarrow^{r,m} H$?

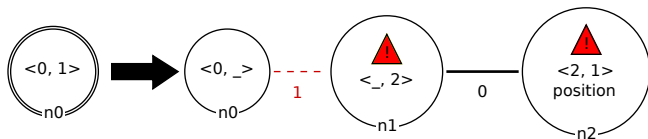
→ *Retour utilisateur*

Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles

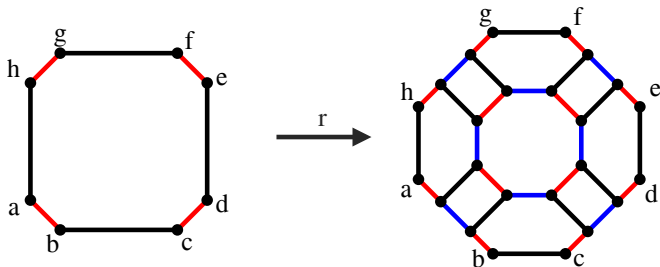
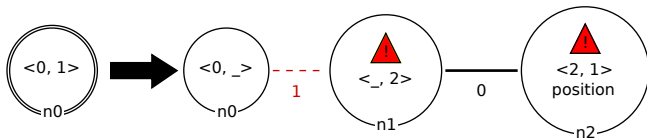
Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles



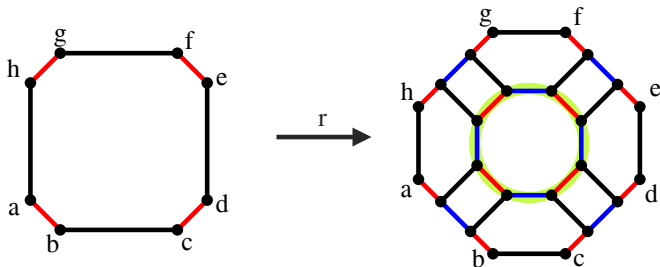
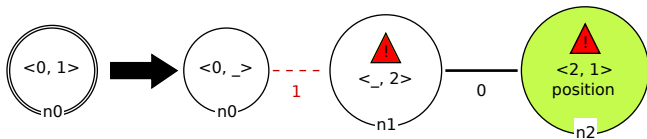
Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles



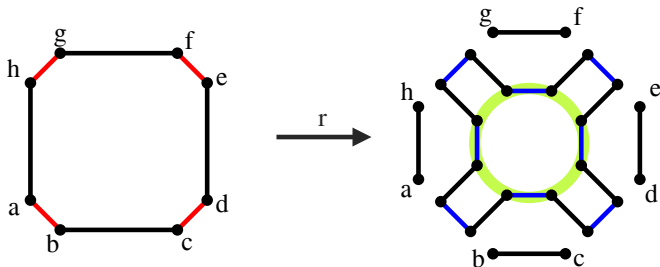
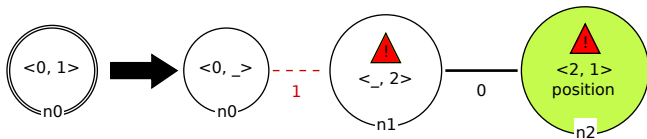
Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles

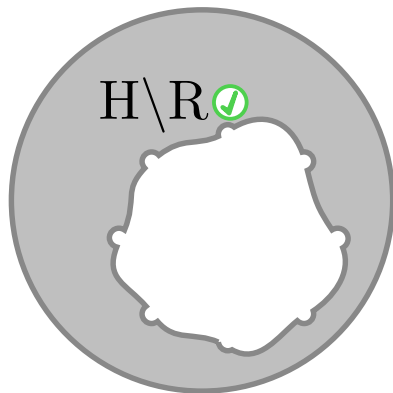


Inconsistances topologiques

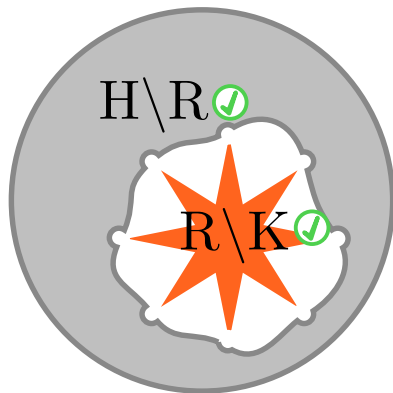
Contrainte de cycle : les chemins 0202 sont des cycles



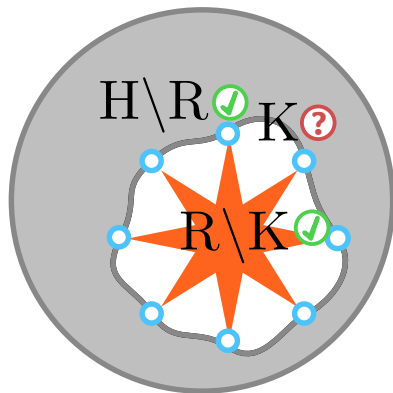
Intuition



Intuition



Intuition



Vérification statique et syntaxique

Non-orientation

Par convention on manipule toujours un arc et son inverse

Vérification statique et syntaxique

Non-orientation

Par convention on manipule toujours un arc et son inverse

Arcs incidents

nœuds créés contrainte imposée

nœuds préservés préservation des incidences

Vérification statique et syntaxique

Non-orientation

Par convention on manipule toujours un arc et son inverse

Arcs incidents

nœuds créés contrainte imposée

nœuds préservés préservation des incidences

Cycles

nœuds créés contrainte imposée

nœuds préservés préservation des chemins partiels

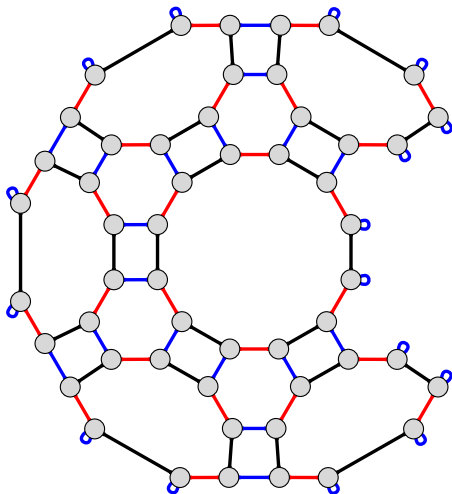
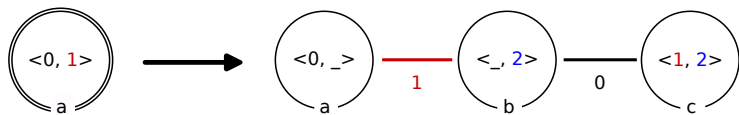
On peut maintenant considérer un schéma de règle \mathcal{S} et qu'on applique le foncteur $\iota(-, P)$ à \mathcal{S} .

Question :

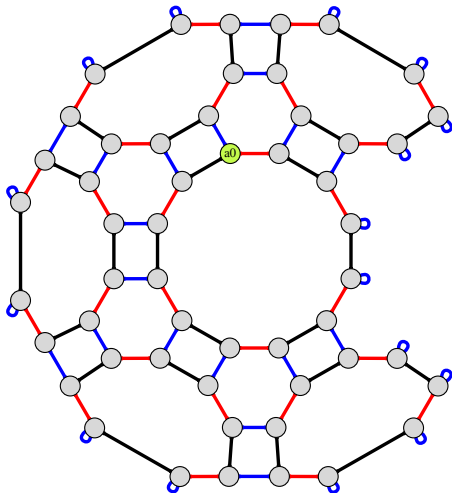
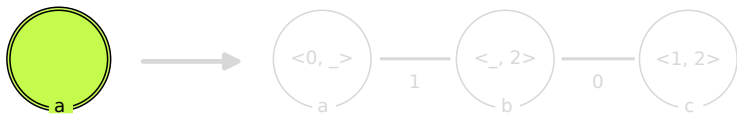
Comment récupérer P ?

Comment résoudre l'unicité à isomorphisme près ?

Instanciation : approche constructive

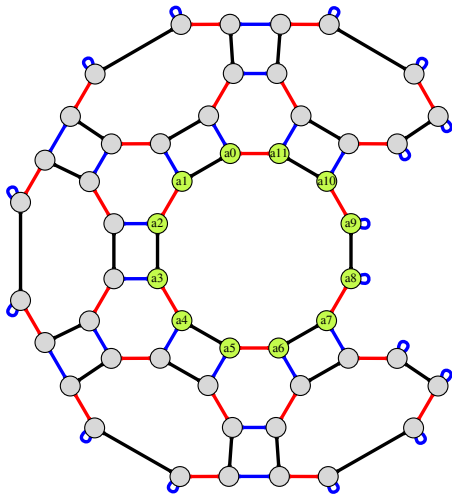
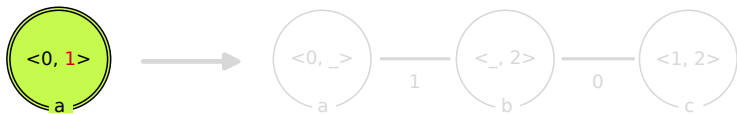


Instanciation : approche constructive



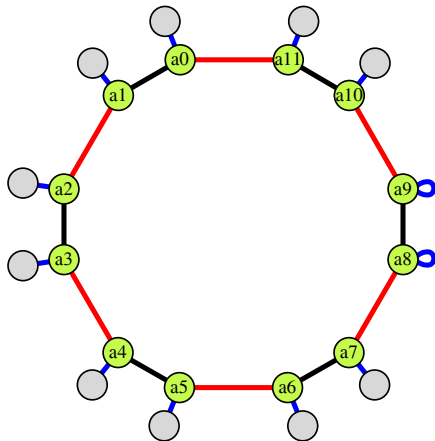
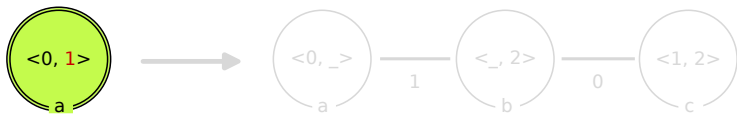
a_0		

Instanciation : approche constructive



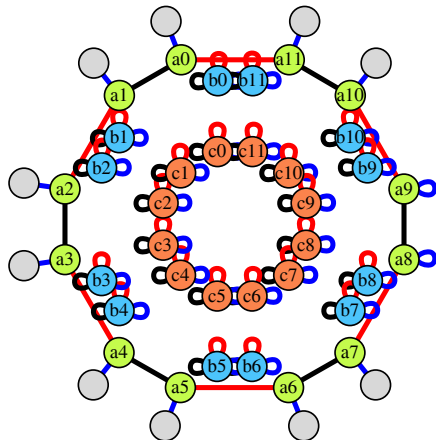
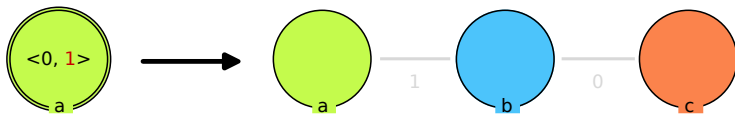
a		
a_0		
a_1		
a_2		
a_3		
a_4		
a_5		
a_6		
a_7		
\vdots		

Instanciation : approche constructive



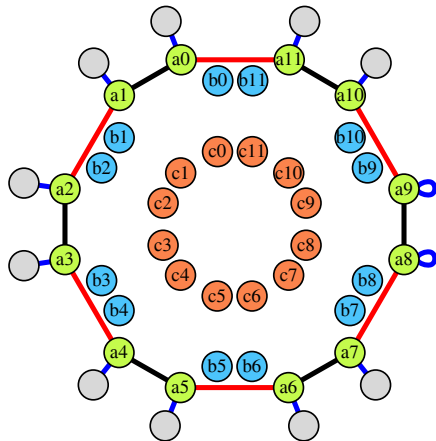
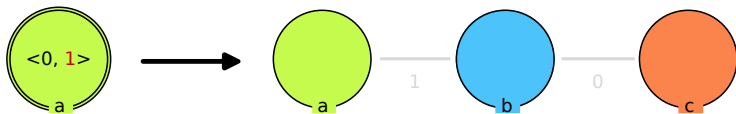
a		
a_0		
a_1		
a_2		
a_3		
a_4		
a_5		
a_6		
a_7		
\vdots		

Instanciación : approche constructive



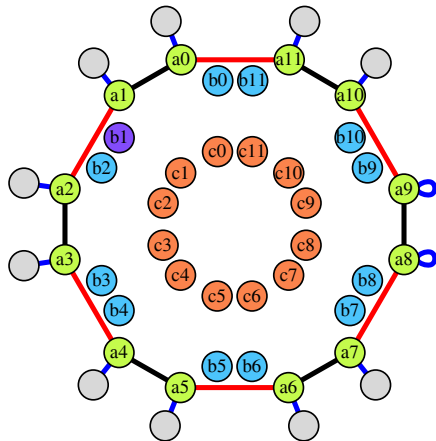
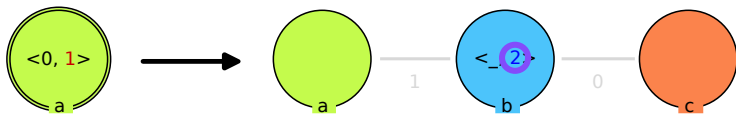
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
⋮	⋮	⋮

Instanciacion : approche constructive



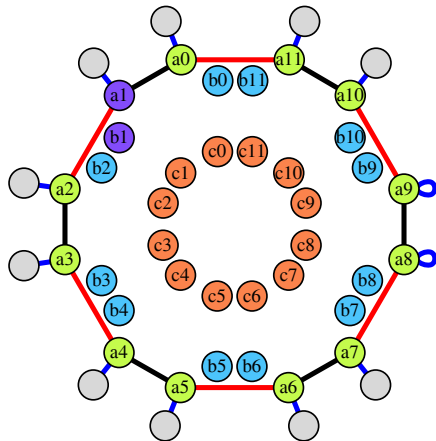
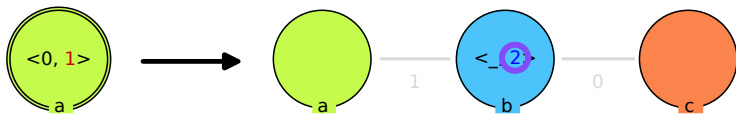
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
\vdots	\vdots	\vdots

Instanciation : approche constructive



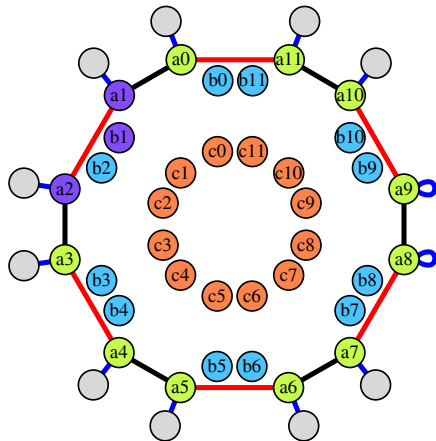
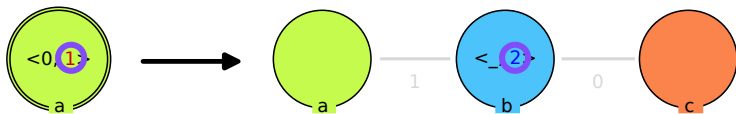
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
⋮	⋮	⋮

Instanciation : approche constructive



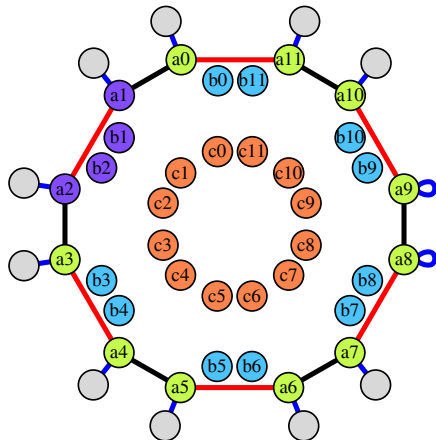
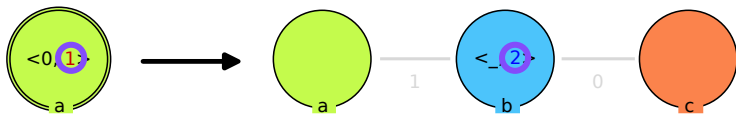
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
⋮	⋮	⋮

Instanciación : approche constructive



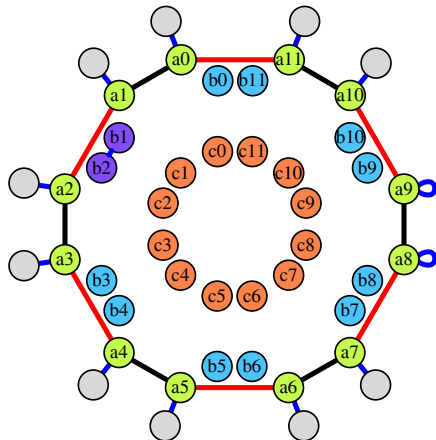
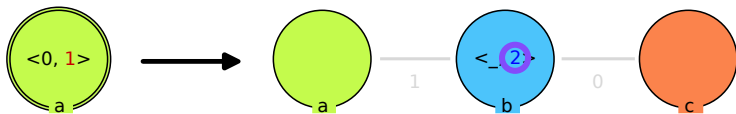
a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
\vdots	\vdots	\vdots

Instanciacion : approche constructive



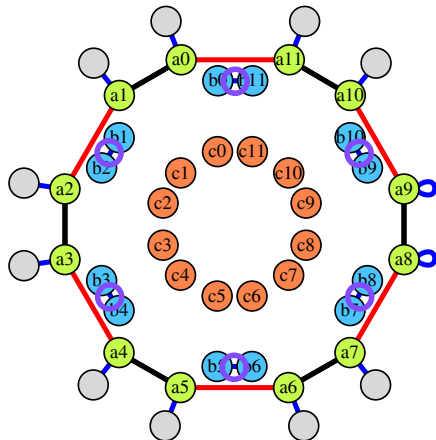
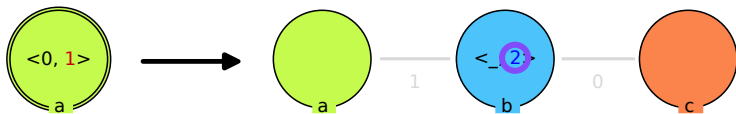
a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
\vdots	\vdots	\vdots

Instanciación : approche constructive



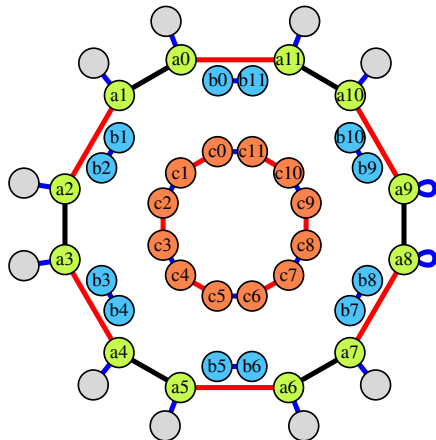
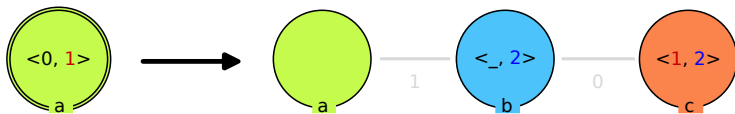
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
⋮	⋮	⋮

Instanciación : approche constructive



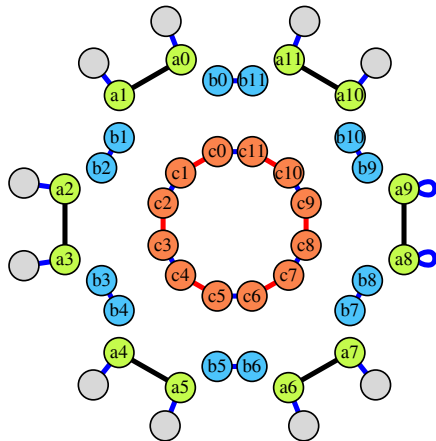
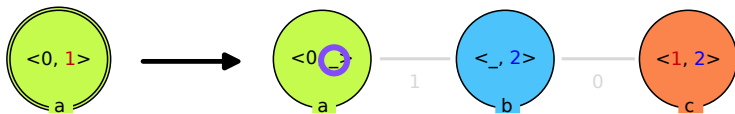
a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
\vdots	\vdots	\vdots

Instanciación : approche constructive



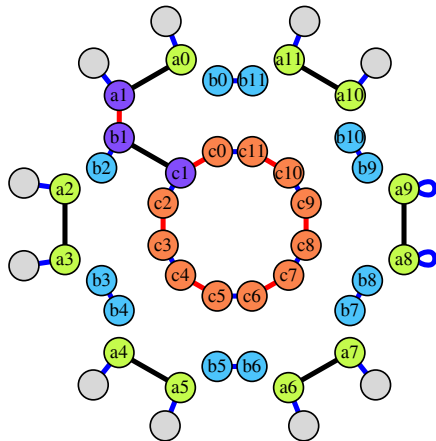
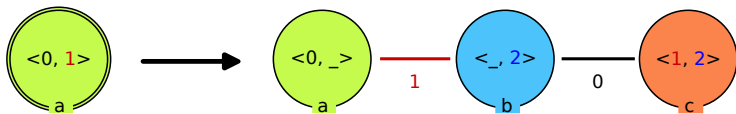
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
⋮	⋮	⋮

Instanciación : approche constructive



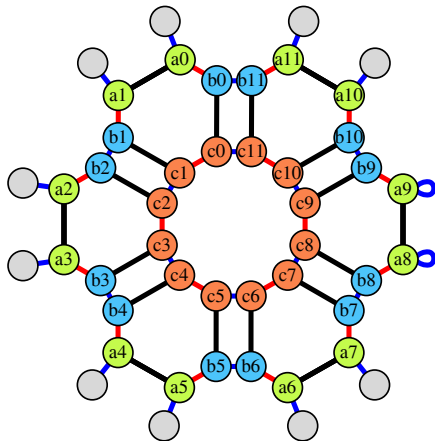
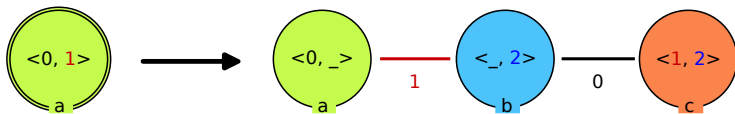
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
\vdots	\vdots	\vdots

Instanciación : approche constructive



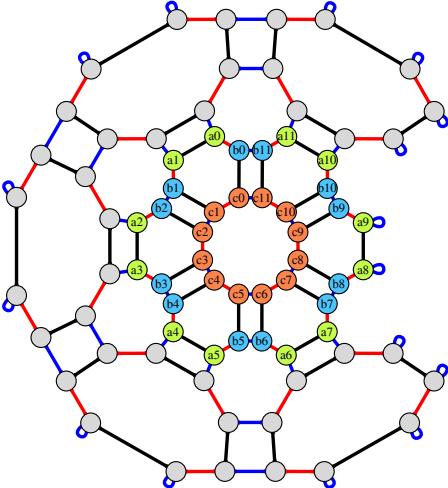
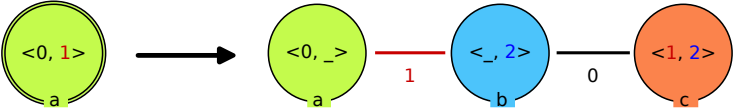
<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
\vdots	\vdots	\vdots

Instanciation : approche constructive



<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
\vdots	\vdots	\vdots

Instanciation : approche constructive



<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i> ₀	<i>b</i> ₀	<i>c</i> ₀
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₃	<i>b</i> ₃	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₄	<i>c</i> ₄
<i>a</i> ₅	<i>b</i> ₅	<i>c</i> ₅
<i>a</i> ₆	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₇	<i>b</i> ₇	<i>c</i> ₇
⋮	⋮	⋮

Algorithme d'application des règles

1. Récupérer l'orbite de support du *hook* principal
2. Par parcours de la structure, construire le tableau décrivant le membre gauche
3. Créer les brins correspondant aux nouveaux nœuds
4. Construire les arcs de renommage de l'orbite (liens verticaux), en terminant par le hook principal
5. Éliminer les brins des nœuds supprimés
6. Construire les arcs entre copies de nœuds distincts (liens horizontaux)

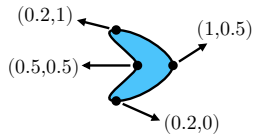
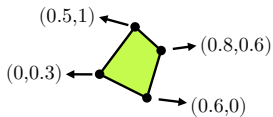
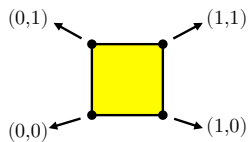
Géométrie

- ▶ Représentation et modification des valeurs associées aux objets

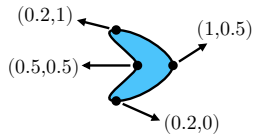
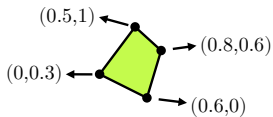
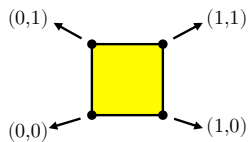
Géométrie



Géométrie



Géométrie



Géométrie



(255,251,0)

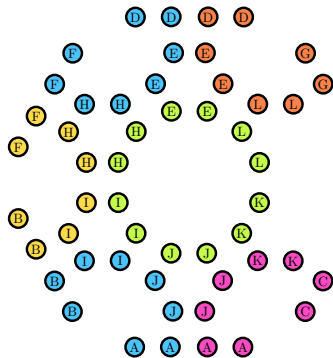


(196,251,76)



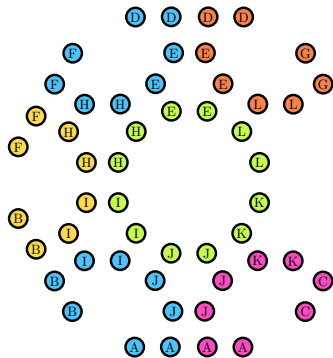
(76,196,251)

Plongements



Légende : 0, 1, 2

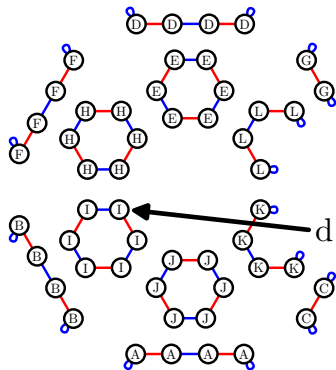
Plongements



Plongement : fonction
 $\pi : \langle o_\pi \rangle \rightarrow \tau_\pi$ avec τ un type de
données abstrait

Légende : 0, 1, 2

Plongements



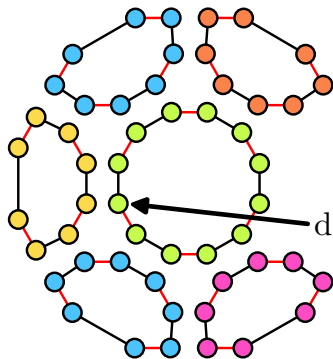
Plongement : fonction

$\pi : \langle o_\pi \rangle \rightarrow \tau_\pi$ avec τ un type de données abstrait

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

Légende : 0, 1, 2

Plongements



Plongement : fonction

$\pi : \langle o_\pi \rangle \rightarrow \tau_\pi$ avec τ un type de données abstrait

- ▶ position : $\langle 1, 2 \rangle \rightarrow \text{Point3}$
- ▶ color : $\langle 0, 1 \rangle \rightarrow \text{ColorRGB}$

Légende : 0, 1, 2

Question : Quelle représentation (abstraite) pour les plongements ?

Rappel sur les types de données (Syntaxe)

Signature $\Omega = (S, F)$ d'un type de données

- ▶ ensemble S de noms de types
- ▶ ensemble F de noms de fonctions munis d'un profil
 $\rho: F \rightarrow S^+$

Pour $f \in F$ tel que $\rho(f) = (s_1, \dots, s_m, s)$, on note

$$f: s_1 \times \dots \times s_m \rightarrow s$$

Forment une catégorie (les morphismes sont les fonctions préservant les profils)

Signature $\Omega(\text{position}, \text{color})$

$S(\text{position}, \text{color}) = \{\text{Point3}, \text{Vector3}, \text{ColorRGB}\}$

$F(\text{position}, \text{color}) = \{\text{plus}, \text{midpoint}, \text{mix}\}$

- ▶ $\text{plus} : \text{Point3} \times \text{Vector3} \rightarrow \text{Point3}$
- ▶ $\text{midpoint} : \text{Point3} \times \text{Point3} \rightarrow \text{Point3}$
- ▶ $\text{mix} : \text{ColorRGB} \times \text{ColorRGB} \rightarrow \text{ColorRGB}$

Rappel sur les types de données (Sémantique)

Ω -algèbre pour une signature $\Omega = (S, F)$

- ▶ famille d'ensembles de valeurs $(\mathcal{A}_s)_{s \in S}$
- ▶ fonction $f^{\mathcal{A}}: \mathcal{A}_{s_1} \times \dots \times \mathcal{A}_{s_m} \rightarrow \mathcal{A}_s$ pour tout nom de fonction $f: s_1 \times \dots \times s_m \rightarrow s$ de F

Forment une catégorie (les morphismes sont les fonctions préservant la structure)

Algèbre $\mathcal{A}(\text{position}, \text{color})$

- ▶ $\mathcal{A}(\text{position}, \text{color})_{\text{Point3}} = \mathbb{R}^3$
- ▶ $\mathcal{A}(\text{position}, \text{color})_{\text{Vector3}} = \mathbb{R}^3$
- ▶ $\mathcal{A}(\text{position}, \text{color})_{\text{ColorRGB}} = [0, 1]^3$

- ▶ $\text{plus}^{\mathcal{A}(\text{position}, \text{color})}$: translation d'un point par un vecteur
- ▶ $\text{midpoint}^{\mathcal{A}(\text{position}, \text{color})}$: milieu du segment
- ▶ $\text{mix}^{\mathcal{A}(\text{position}, \text{color})}$: mélange de deux couleurs

Rappel sur les graphes attribués

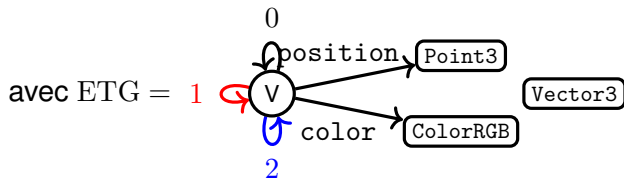
Grphe Ω -attribué $AG = (G, \mathcal{A})$ est défini par

- ▶ E-graphe $G = (V_G, D_G, E_G, A_G, s_G, t_G, sa_G, ta_G)$
- ▶ Ω -algèbre \mathcal{A} telle que $D_G = \bigsqcup_{s \in S} \mathcal{A}_s$

Gates plongées sur position et color

On considère la catégorie

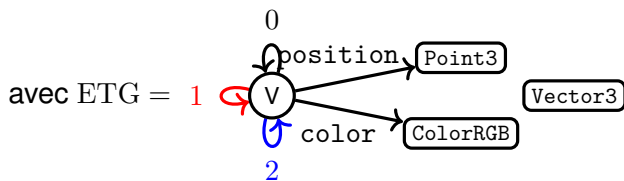
$$\Omega(\text{position, color})\text{-AGraph}/(\text{ETG}, \mathbf{1}\text{Alg}(\Omega(\text{position, color})))$$



Gates plongées sur position et color

On considère la catégorie

$$\Omega(\text{position, color})\text{-AGraph}/(\text{ETG}, \mathbf{1}\text{Alg}(\Omega(\text{position, color})))$$

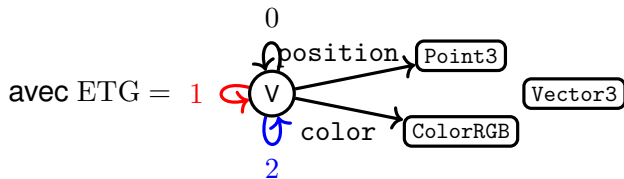


Contraintes pour l'existence et l'unicité des valeurs

Gartes plongées sur position et color

On considère la catégorie

$$\Omega(\text{position, color})\text{-AGraph}/(\text{ETG}, \mathbf{1}\text{Alg}(\Omega(\text{position, color})))$$



Contraintes pour l'existence et l'unicité des valeurs

Contrainte de cohérence au type d'orbite

Question : Quel type de données pour la modélisation géométrique ?

Question : Quel type de données pour la modélisation géométrique ?

Signature et algèbre étendues 'à la volée'

Question : Comment modifier le type de données ?

Question : Quel type de données pour la modélisation géométrique ?

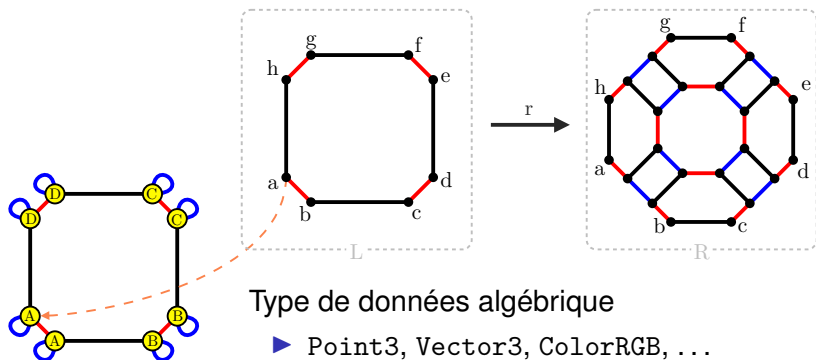
Signature et algèbre étendues 'à la volée'

Question : Comment modifier le type de données ?

Ajout d'un nom de type (resp. nom de fonction) dans la signature et d'un support (resp. d'une fonction) dans l'algèbre

→ implémentation de la fonction

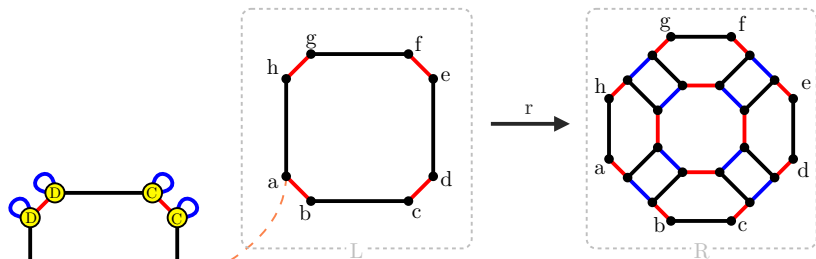
Modification d'un attribut



Type de données algébrique

- ▶ Point3, Vector3, ColorRGB, ...
- ▶ plus, midpoint, mix, ...

Modification d'un attribut



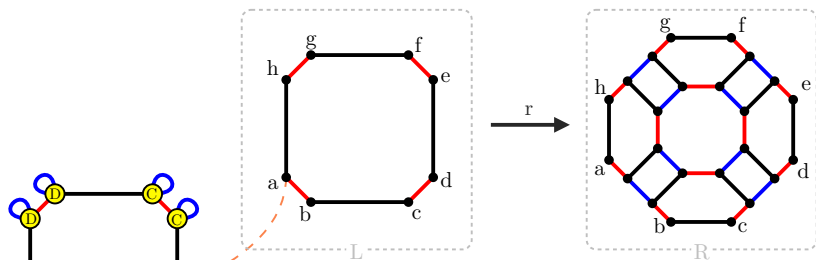
Etendu avec des opérateurs de structure

► Accesseurs

`a.position = A`

`a.color = ●`

Modification d'un attribut



Etendu avec des opérateurs de structure

▶ Accesseurs

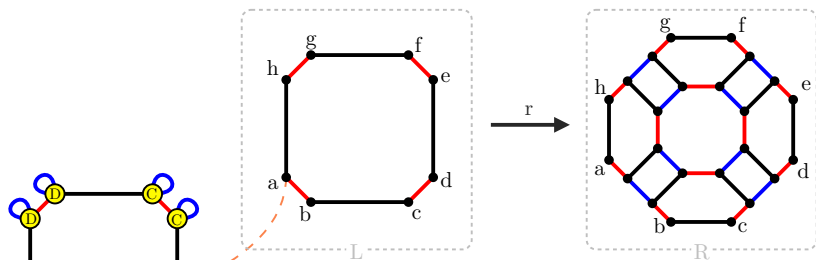
▶ Opérateurs de voisinage

$a@0@1@0.position =$

$f.position = C$

$a@1@0.color = c.color = \bullet$

Modification d'un attribut



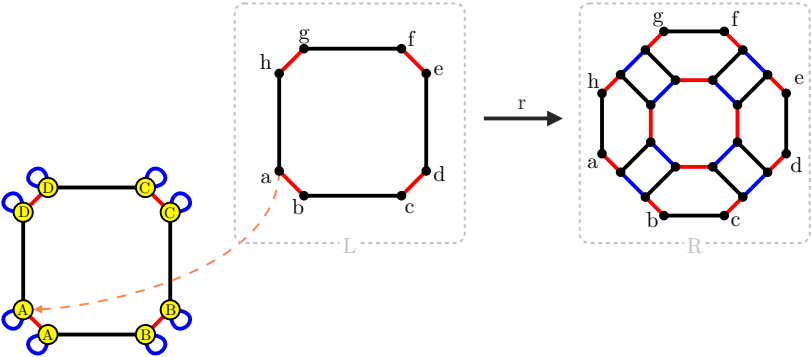
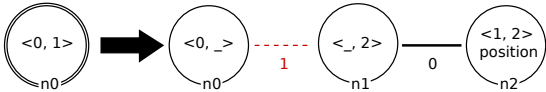
Etendu avec des opérateurs de structure

- ▶ Accesseurs
- ▶ Opérateurs de voisinage
- ▶ Opérateurs de collecte

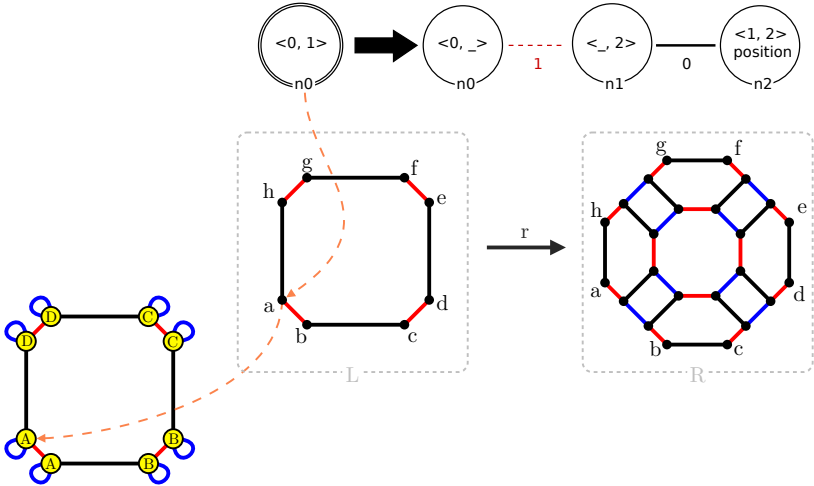
$\text{position}_{\langle 0,1 \rangle}(a) = \llbracket A, B, C, D \rrbracket$

$\text{color}_{\langle 0,1 \rangle}(a) = \llbracket \bullet \rrbracket$

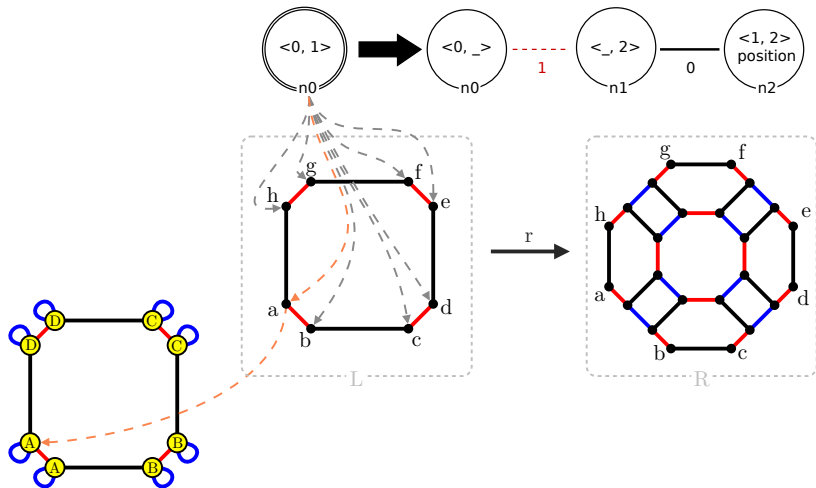
Extension aux schémas



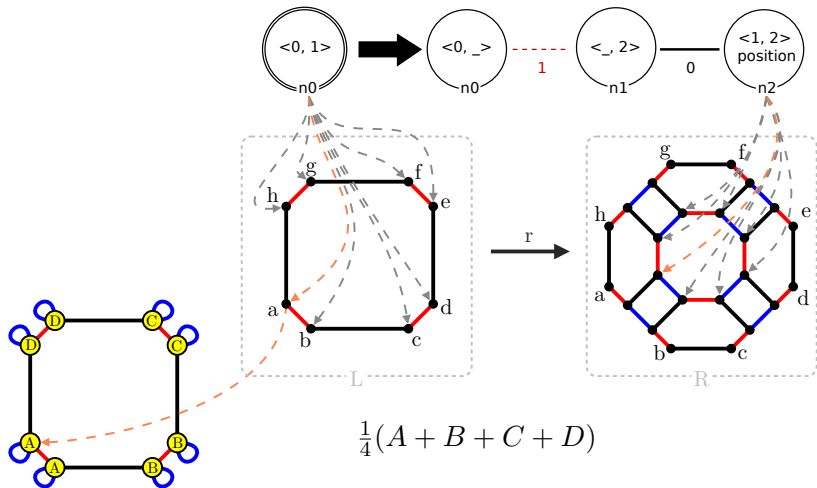
Extension aux schémas



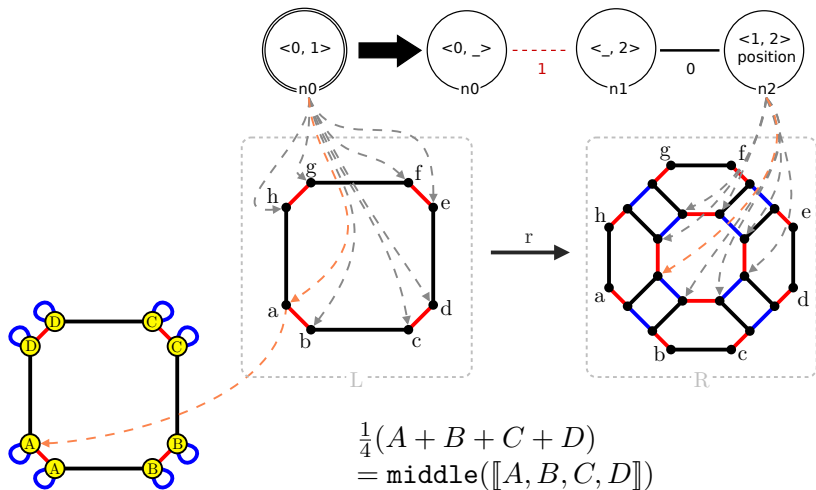
Extension aux schémas



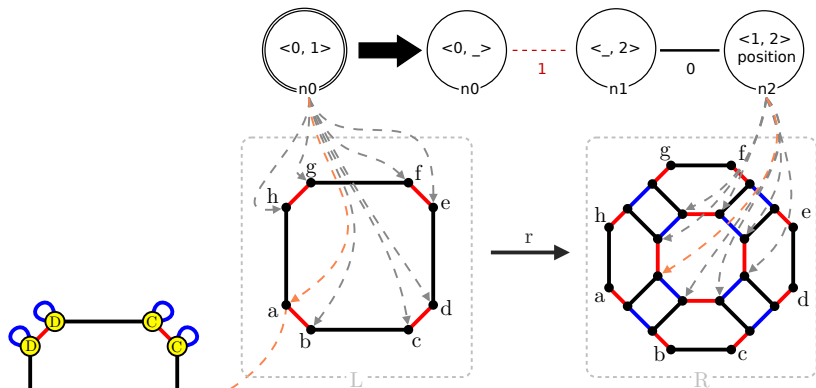
Extension aux schémas



Extension aux schémas

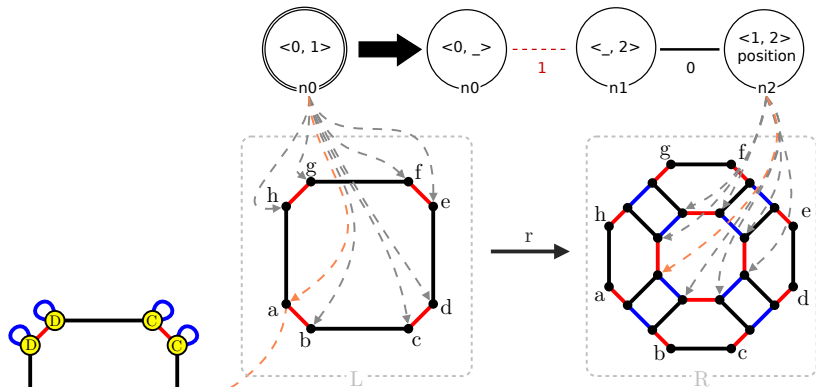


Extension aux schémas



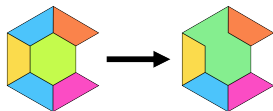
$$\begin{aligned}
 & \frac{1}{4}(A + B + C + D) \\
 &= \text{middle}([A, B, C, D]) \\
 &= \text{middle}(\text{position} \langle 0, 1 \rangle(a))
 \end{aligned}$$

Extension aux schémas

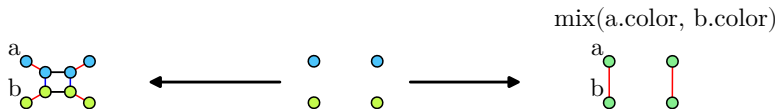


$$\begin{aligned}
 & \frac{1}{4}(A + B + C + D) \\
 &= \text{middle}([A, B, C, D]) \\
 &= \text{middle}(\text{position } \langle 0, 1 \rangle(a)) \\
 &= \text{middle}(\text{position } \langle 0, 1 \rangle(n_0))
 \end{aligned}$$

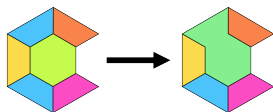
Inconsistance géométrique



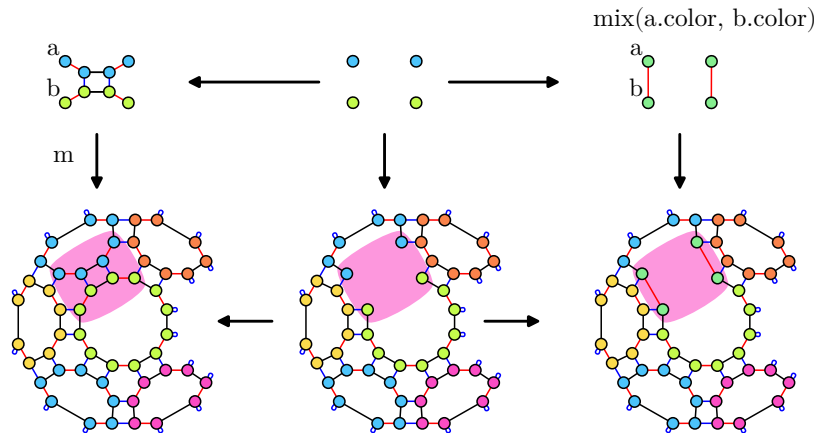
Contrainte brins d'une $\langle 0, 1 \rangle$ -orbite ont la même couleur



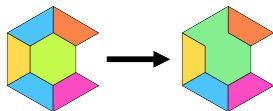
Inconsistance géométrique



Contrainte brins d'une $\langle 0, 1 \rangle$ -orbite ont la même couleur



Inconsistance géométrique

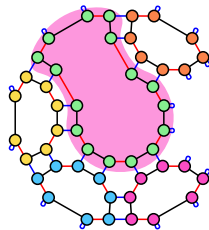
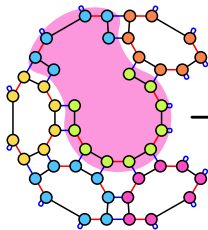
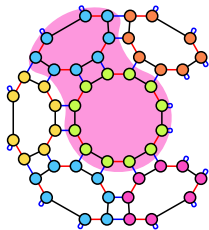


Contrainte brins d'une $\langle 0, 1 \rangle$ -orbite ont la même couleur

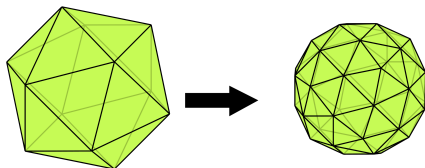
$\text{mix}(\text{a.color}, \text{b.color})$



↓ Rule completion



D'un langage abstrait vers un langage concret



```
int n = 0;
JerboaList <@ebd<position>> sommetsAdjacents ;
for (JerboaDart d : <1,2>_<2>(n0)) {
    sommetsAdjacents.add(d@0.position);
    n = n+1;
}
Point3 bary = Point3::middle(sommetsAdjacents);

if (n==2){
    return Point3::barycenter(n0.position, 3.0/4.0, bary, 1.0/4.0);
}
else if (n==3){
    return Point3::barycenter(n0.position, 7.0/16.0, bary, 9.0/16.0);
}
else {
    return Point3::barycenter(n0.position, 5.0/8.0, bary, 3.0/8.0);
}
```

Conclusion

Dans cette séance nous avons évoqué une mise en application
les transformations de graphes dans un domaine spécifique

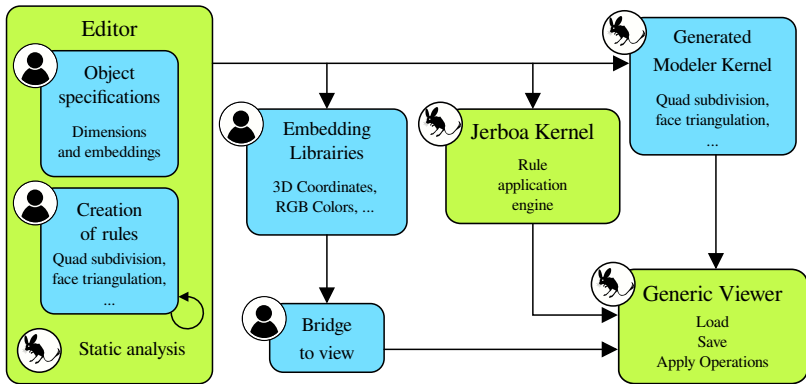
→ la modélisation géométrique

- ▶ Adapter la théorie générale à nos besoins
- ▶ Construction abstraite dédiées
- ▶ Implémentation par des algorithmes efficaces

Suite : mise en application

Jerboa

- ▶ Une plateforme de conception de modeleurs géométriques



Automated



User input



Generic



Specific

File View Misc Look and Feel

Rules:

Dart: Orbit: <> Dongling:

Count: 1

ID: 23

a0: 27	a1: 19
a2: 22	a3: 23

Move:

a0: 27	a1: 19
a2: 22	a3: 23

Embeddings:

point(0): <0.25;0.0;0.75>

color(1): Color<0.76862746;0.9843>

orient(2): true

normal(3): <0.0;1.0;0.0> (c)

Orbit selection:

alpha 0 alpha 1

alpha 2 alpha 3

Operations:

Center:

Console Memory Usage View settings Camera settings Camera backup Top view param

```
[175.06445;10008.484;0.6999743]
[-1744.6978;-99974.766;0.6999743]
```

Information on GMap: ArrayListG-Map dimension 3 load 40/2047 without hole:0 SIZE:40

Mem info:

File Edition Window

SewA0 SubdivideQuad

User name: remain

Modeler's Name: (a)

Demc]

Modeler's package: up.xlim.ig.jerboa.demo

Dimension: 3

Adv. param. Check all

Embeddings(4): (b)

point:<a1, a2, a3> -> up.xlim.ig.jerbc
 color:<a0, a1> -> up.xlim.ig.jerboa.d
 orient:<-> -> java.lang.Boolean
 normal:<a0, a1> -> up.xlim.ig.jerboa.

Rules:167

- UnSewA0
- UnSewA1
- UnSewA2
- UnSewA2All
- UnSewA3
- structural**
- subdivision**
 - DooSabin
 - DooSabin3D
 - Menger3D
 - Sqrt3
 - Sqrt3Copy
 - SubCatmullClarkConstra
 - SubCatmullClarkConstra
 - Subdivide3D
 - SubdivideCatmullClark
 - SubdivideCatmullClark2
 - SubdivideLoop
 - SubdivideQuad (c)
- suppression**

Filter: x


SubdivideQuad Undock (d)

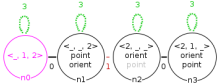
File Views


Scale x1 Composition CompositionSelf Shape: CIRCLE Grid size: MEDIUM

Name: n0 Orbit: <a0, a1, a2> Hook Precond

Name: n0 Orbit: <_, a1, a2> point:<a1, a2, a3> EDIT

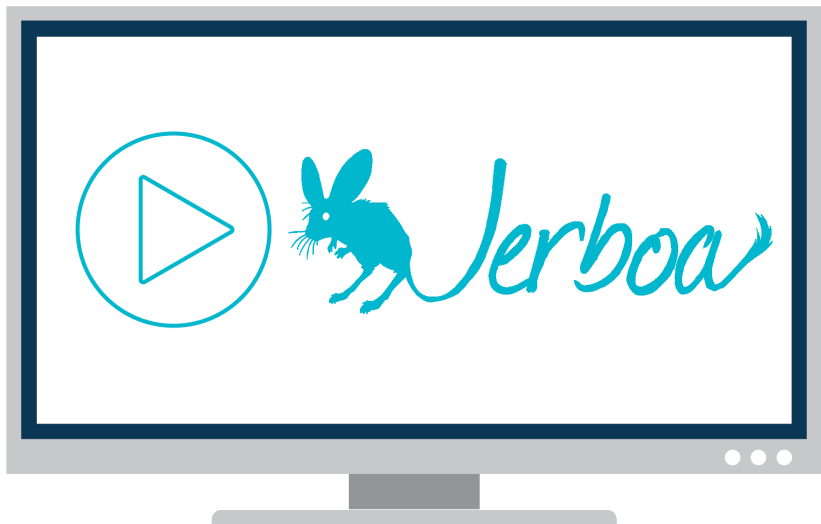
(b) 

(c) 




(e) 

(f) Details Topo. Param. Ebd. Param. Errors x




Informations:





References I

-  BAUDERON, Michel (1^{er} jan. 1995). « Parallel Rewriting of Graphs through the Pullback Approach ». In : *Electronic Notes in Theoretical Computer Science*. SEGRAGRA 1995 2, p. 19-26. ISSN : 1571-0661. DOI : [10.1016/S1571-0661\(05\)80176-8](https://doi.org/10.1016/S1571-0661(05)80176-8).
-  BEHR, Nicolas et al. (22 avr. 2021). « Compositionality of Rewriting Rules with Conditions ». In : *Compositionality 3.2*, p. 51. ISSN : 2631-4444. DOI : [10.32408/compositionality-3-2](https://doi.org/10.32408/compositionality-3-2). URL : <https://compositionality-journal.org/>.
-  BELHAOUARI, Hakim et al. (2014). « Jerboa: A Graph Transformation Library for Topology-Based Geometric Modeling ». In : *Graph Transformation*. ICGT 2014. Sous la dir. d'Holger GIESE et al. Lecture Notes in Computer Science. Cham : Springer International Publishing, p. 269-284. ISBN : 978-3-319-09108-2. DOI : [10.1007/978-3-319-09108-2_18](https://doi.org/10.1007/978-3-319-09108-2_18).



References II

-  COURCELLE, B. (fév. 1997). « The expression of graph properties and graph transformations in monadic second-order logic ». In : *Handbook of Graph Grammars and Computing by Graph Transformation: Volume I. Foundations*. USA : WORLD SCIENTIFIC, p. 313-400. ISBN : 978-981-02-2884-2. DOI : [10.1142/9789812384720_0005](https://doi.org/10.1142/9789812384720_0005).
-  DAMIAND, Guillaume et al. (19 sept. 2014). *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. CRC Press. 407 p. ISBN : 978-1-4822-0652-4.
-  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](https://doi.org/10.1007/3-540-31188-2).

References III

-  EHRIG, Hartmut et al. (août 2014). « M-adhesive transformation systems with nested application conditions. Part 1: parallelism, concurrency and amalgamation ». In : *Mathematical Structures in Computer Science* 24.4. ISSN : 0960-1295, 1469-8072. DOI : [10.1017/S0960129512000357](https://doi.org/10.1017/S0960129512000357).
-  HABEL, Annegret et al. (avr. 2009). « Correctness of high-level transformation systems relative to nested conditions ». In : *Mathematical Structures in Computer Science* 19.2. Publisher: Cambridge University Press, p. 245-296. ISSN : 1469-8072, 0960-1295. DOI : [10.1017/S0960129508007202](https://doi.org/10.1017/S0960129508007202).

References IV

-  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. URL : <http://link.springer.com/10.1007/978-3-030-43916-3> (visité le 25/06/2020).
-  LIENHARDT, Pascal (5 juin 1989). « Subdivisions of N-dimensional Spaces and N-dimensional Generalized Maps ». In : *Proceedings of the Fifth Annual Symposium on Computational Geometry*. SCG '89. New York, NY, USA : Association for Computing Machinery, p. 228-236. ISBN : 978-0-89791-318-8. DOI : 10.1145/73833.73859.

References V



ROZENBERG, Grzegorz, éd. (fév. 1997). *Handbook of Graph Grammars and Computing by Graph Transformation: Volume 1: Foundations*. T. Foundations. 1 t. USA : World Scientific. 572 p. ISBN : 978-981-02-2884-2. DOI : 10.1142/3303.