

# **Subdivisions Volumiques et Cartes Combinatoires Généralisées**

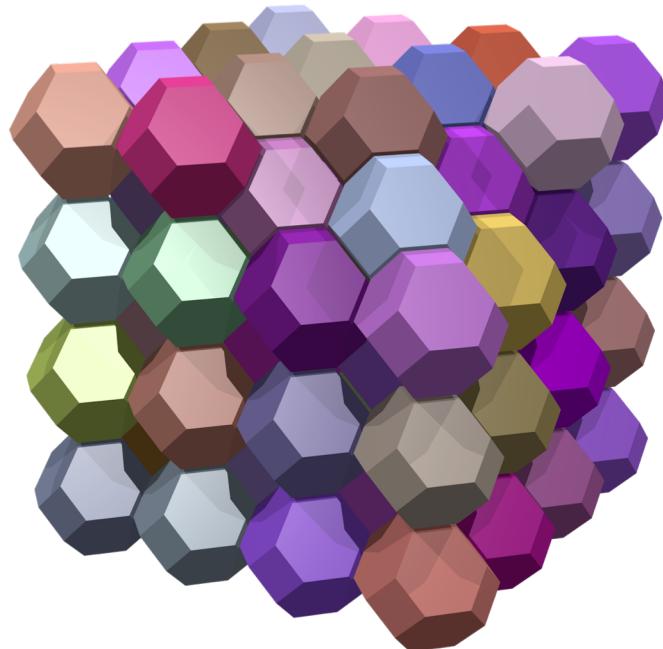
Fred Boudon

Inspiré des présentations de O. Terraz, G. Damiand et G. Cerutti.

## Subdivision d'un espace topologique

Partition d'un espace en un complexe formé de **cellules**

- Cellules de dimension 0, 1, 2, 3...
- Subdivision de  $\mathbb{R}^2$ : partition en sommets, arêtes et cellules (surficiques)
- Subdivision de  $\mathbb{R}^3$ : partition en sommets, arêtes, faces et cellules



Topologie / Géométrie :

- Adjacence
- Incidence
- Plongement

## Subdivision volumique

Très nombreuses structures pour la représentation de subdivisions topologiques:

- Surfaces orientables sans bord (dimension 2)
  - Graphe d'incidence des faces [AFF85]
  - Structure d'arête ailée [Baum74], [Weil85]
- Surfaces, orientables ou non, avec ou sans bords (dimension 2)
  - Quad-edge [GS85]
  - Extensions de cartes combinatoires [Tutt84], [Lien91a]
- Objets volumiques réguliers, orientables ou non, avec ou sans bords (dimension 3)
  - Structure facet-edge [DL87]
  - Structure cell\_tuple [Bris89]
  - Graphes d'incidence [Sobh89]
- Objets volumiques complexes (dimension 3)
  - Structure radial-edge [Weil86]
- Objets de dimension n... [LL90], [CR91]

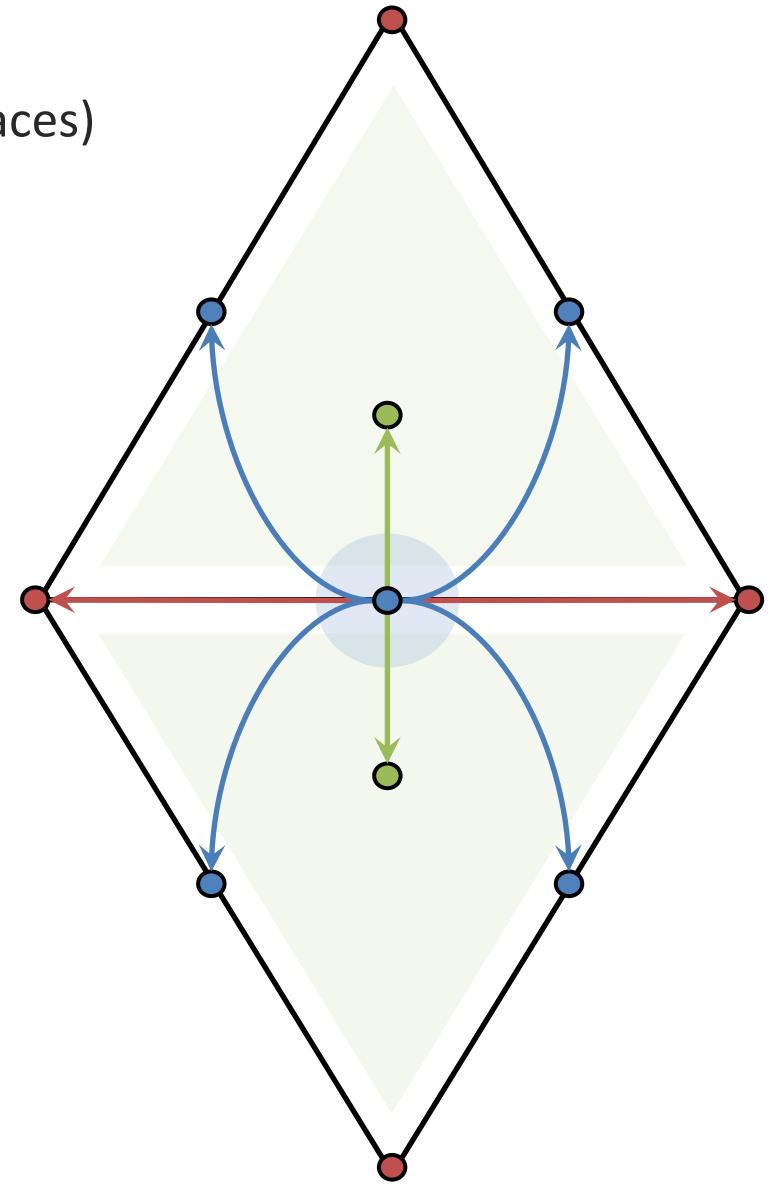
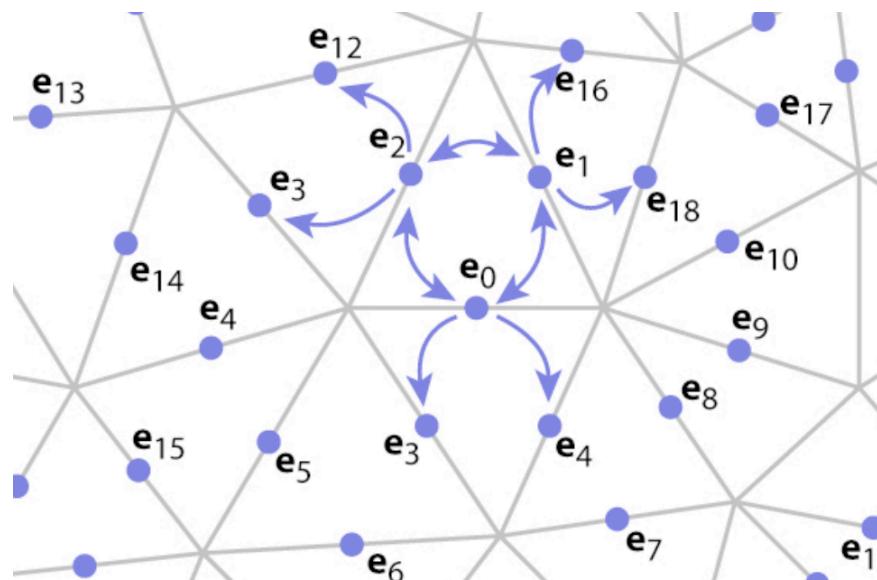
# Winged Edge [BAUMGART 74]

Représentation centrée sur les **arêtes** (et non les faces)

Chaque arête pointe sur:

- Sommets avant et arrière
- Faces gauche et droite
- Arêtes avant gauche et droite
- Arêtes arrières gauche et droite

Chaque sommet/face pointe sur **1** arête



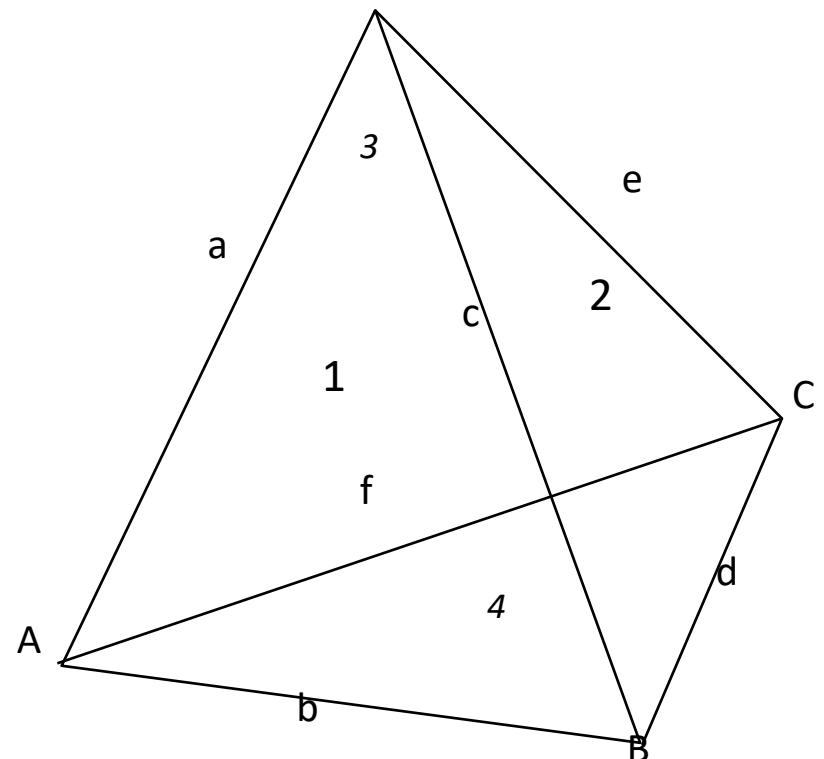
# Winged Edge [BAUMGART 74]

Volume de stockage

- Trois tableaux, 8 indices par arête

Vert	Edge
A	a
B	b
C	d
D	e

Face	Edge
1	a
2	c
3	a
4	b



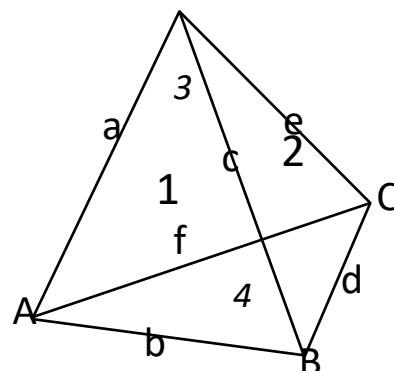
Edge	Vert		Face		Left		Right		
	Nom	Tail	Head	Left	Right	Tail	Head	Tail	Head
a	A	D	B	3	1	e	f	b	c
b	A	B	D	1	4	c	a	f	d
c	B	D	C	1	2	a	b	d	e
d	B	C	C	2	4	e	c	b	f
e	C	D	D	2	3	c	d	f	a
f	A	C	C	4	3	d	b	a	e

# Winged Edge [BAUMGART 74]

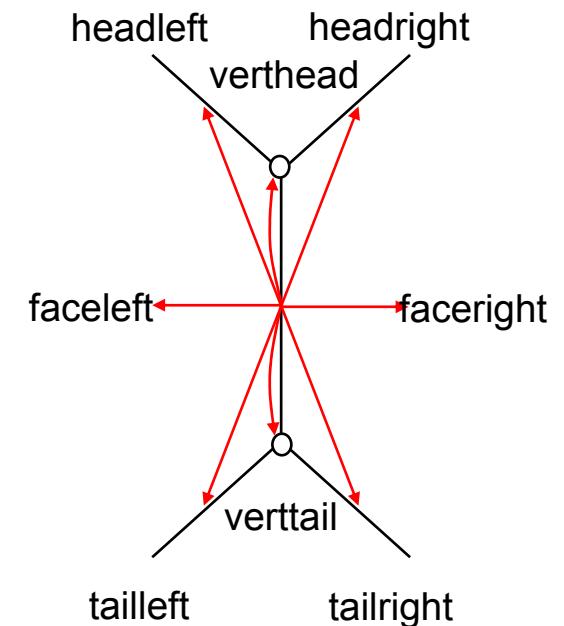
Ambigüité de l'orientation des arêtes

- Problème de la navigation : parcourir les arêtes d'une face ?

Face	Edge
1	a
2	c
3	a
4	b



Edge		Vert		Face		Left		Right	
Nom	Tail	Head	Left	Right	Tail	Head	Tail	Head	
a	A	D	3	1	e	f	b	c	
b	A	B	1	4	c	a	f	d	
c	B	D	1	2	a	b	d	e	
d	B	C	2	4	e	c	b	f	
e	C	D	2	3	c	d	f	a	
f	A	C	4	3	d	b	a	e	

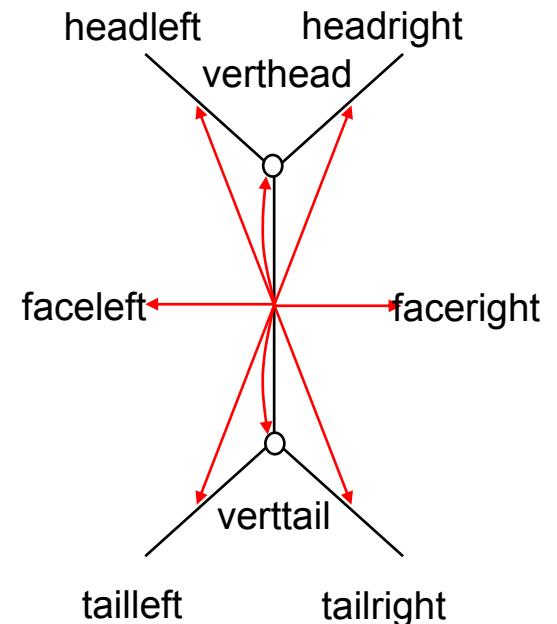


# Winged Edge [BAUMGART 74]

Ambigüité de l'orientation des arêtes

- Problème de la navigation : parcourir les arêtes d'une face :

```
def edges_of_a_face(f):
    e0 = f.edge
    e = e0
    while True:
        if e.faceleft == f:
            e = e.headleft
        else :
            e = e.tailright
        if e == e0: break
```



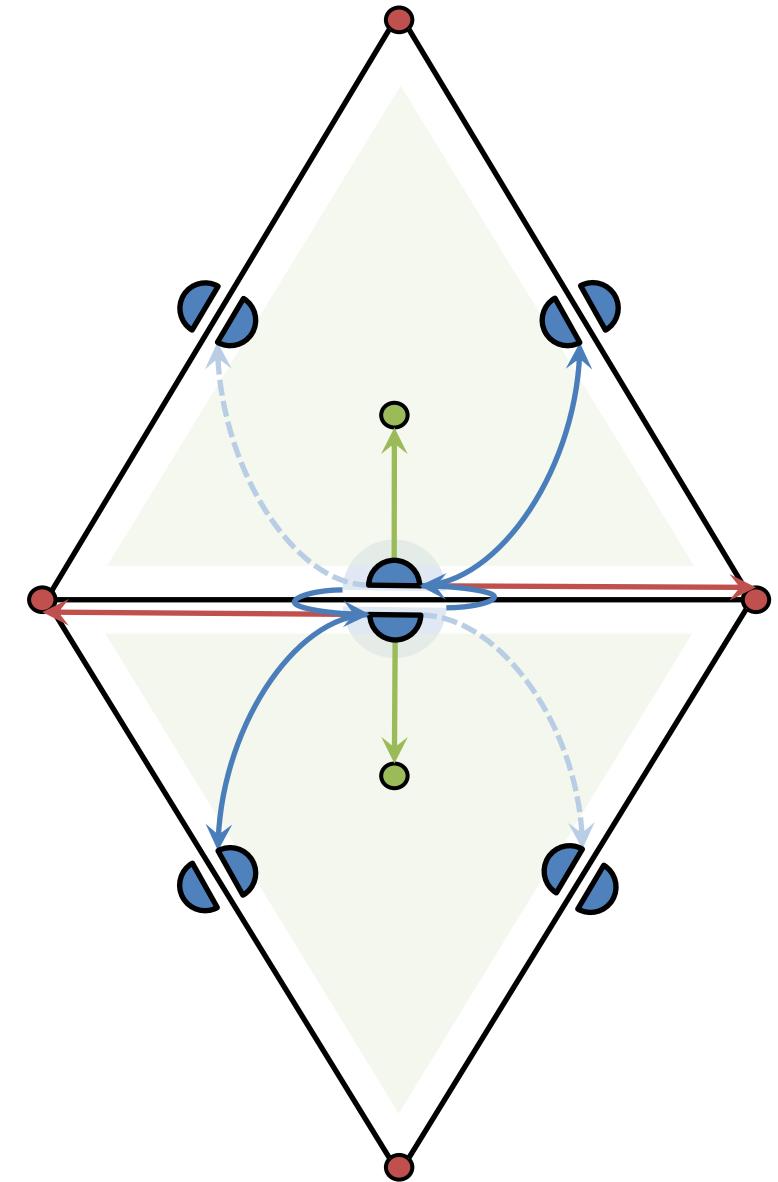
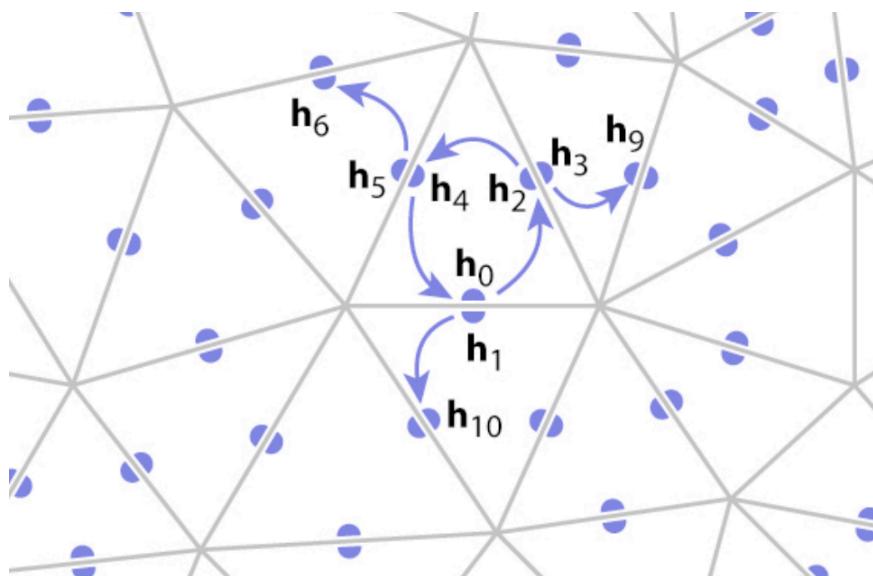
# Half Edge [Weiler 85]

Représentation simplifiée et désambiguisee

Chaque demi-arête pointe sur:

- Sommet avant
- Face incidente
- Demi-arête opposée
- Demi-arête avant (même face)
- Demi-arête arrière (optionnel)

Chaque sommet/face pointe sur **1** demi-arête



# Half Edge [Weiler 85]

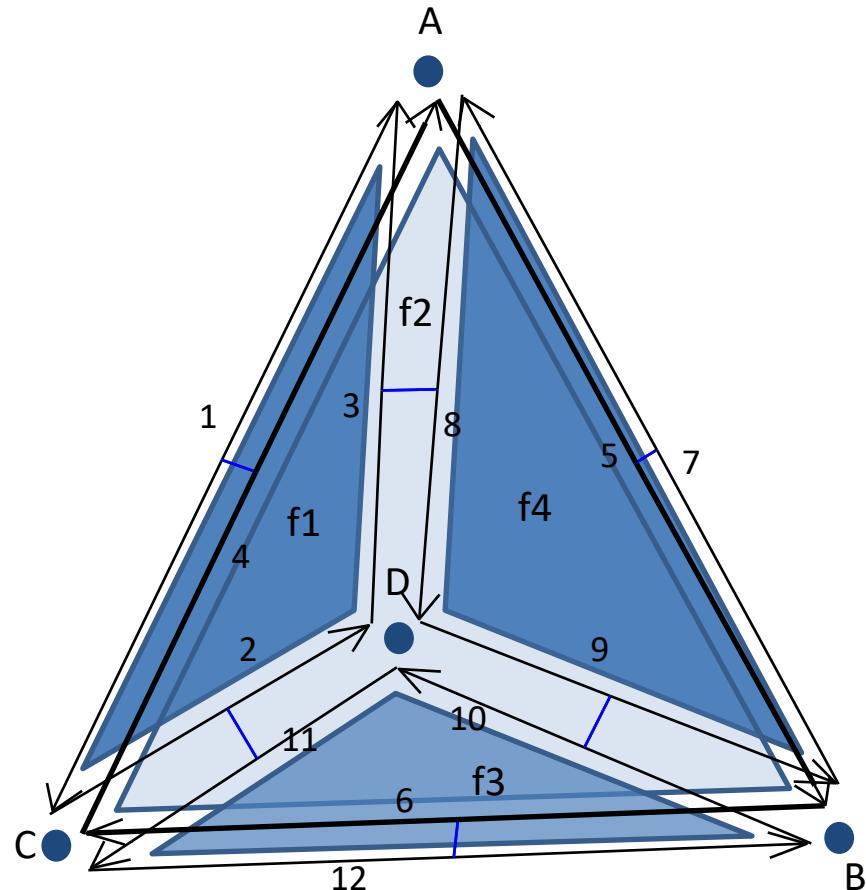
Volume de stockage comparable

- Trois tableaux, 4 indices par demi-arête

Vert	Demi-arêtes
A	8
B	10
C	2
D	9

Face	Demi-arêtes
f1	1
f2	5
f3	10
f4	4

Demi-arêtes	vert	face	opposite	head	(tail)
1	A	f1	4	2	3
2	C	f1	11	3	1
3	D	f1	8	1	2
4	C	f4	1	5	6
5	A	f4	7	6	4
6	B	f4	12	4	5
7	B	f2	5	8	9
8	A	f2	3	9	7
9	D	f2	10	7	8
10	B	f3	9	11	12
11	D	f3	2	12	10
12	C	f3	6	10	11

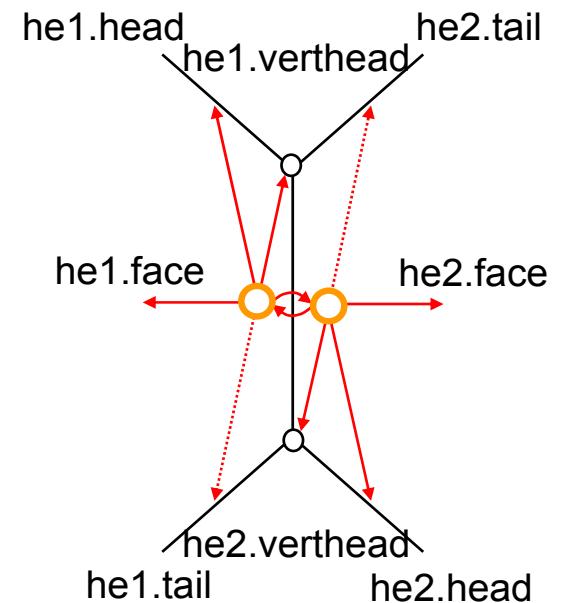


# Half Edge [Weiler 85]

Parcours plus simple

- Problème de la navigation : parcourir les arêtes d'une face :

```
def edges_of_a_face(f):
    e0 = f.edge
    e = e0
    while True:
        e = e.head
        if e == e0: break
```

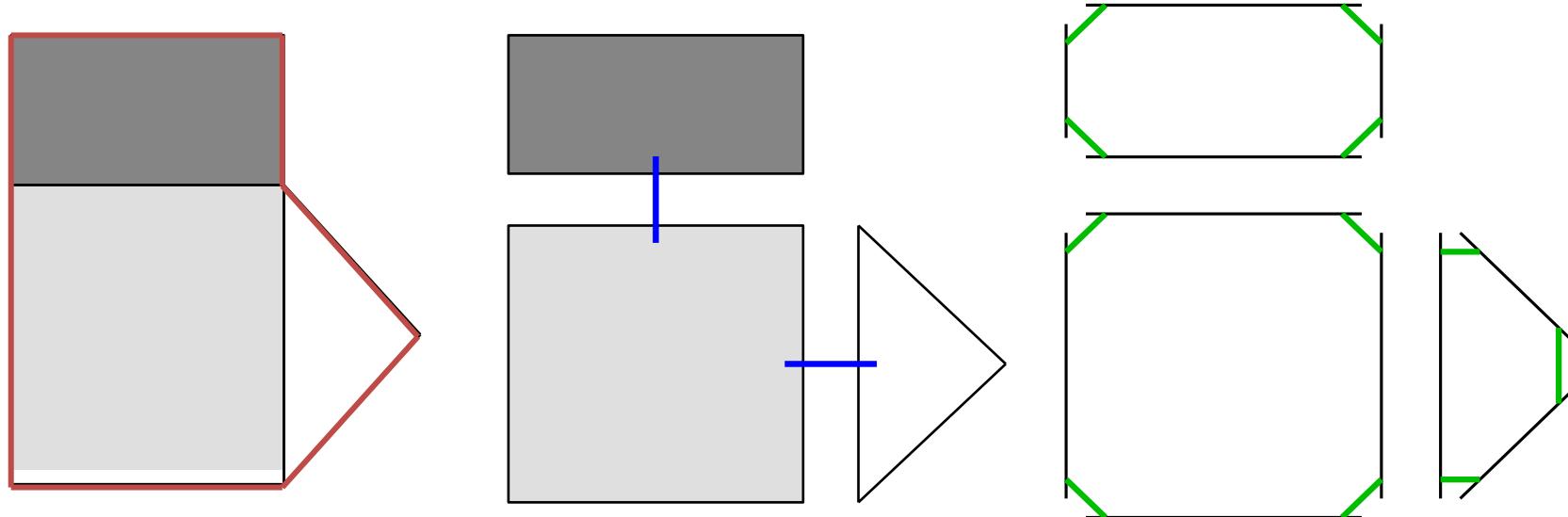


Décomposition en éléments plus simples pour mieux représenter la topologie

# Obtention intuitive d'une carte combinatoire

Décompositions successives des cellules d'un objet

- Exemple en 2D :



Objet 2D

Faces

Arêtes  
 $\Rightarrow$  Brins

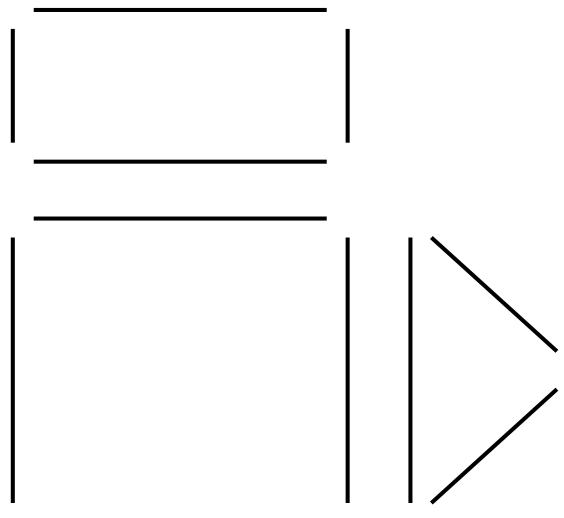
Représenter les cellules de l'objet avec uniquement les brins comme éléments

# Obtention intuitive d'une carte combinatoire

Construction d'une 2-Carte

- Brins

Intuitivement un brin peut être vu et être graphiquement représenté comme une « portion » d'arête :

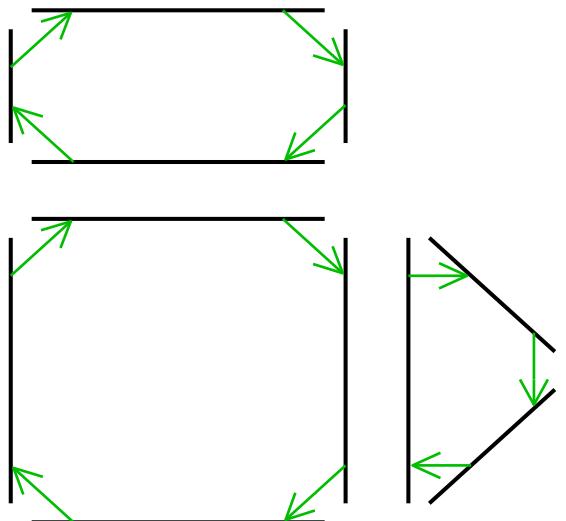


Opérateurs d'assemblage  $\beta_1$  et  $\beta_2$

# Obtention intuitive d'une carte combinatoire

## Construction d'une 2-Carte

- Report des relations d'adjacence sur les brins



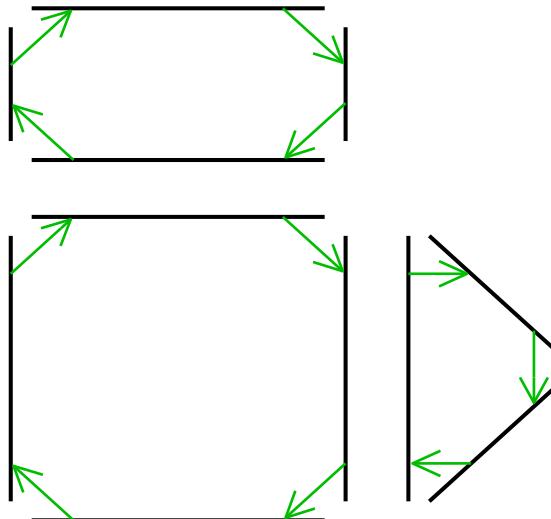
$\beta_1$  : brin  $\rightarrow$  brin suivant de la même face

$\beta_1$  est une permutation (bijection de  $B$  vers  $B$ ) représentant l'orientation de la face.

# Obtention intuitive d'une carte combinatoire

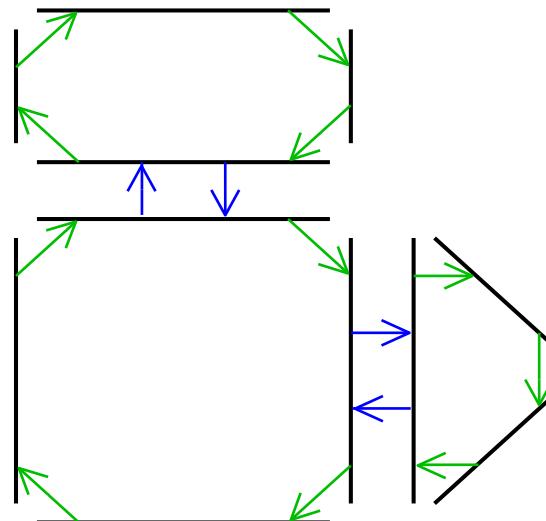
## Construction d'une 2-carte

- Report des relations d'adjacence sur les brins



$\beta_1$  : brin  $\rightarrow$  brin suivant de la même face

$\beta_1$  est une permutation (bijection de  $B$  vers  $B$ ) représentant l'orientation de la face.



$\beta_2$  : brin  $\rightarrow$  brin de l'autre face incidente à la même arête

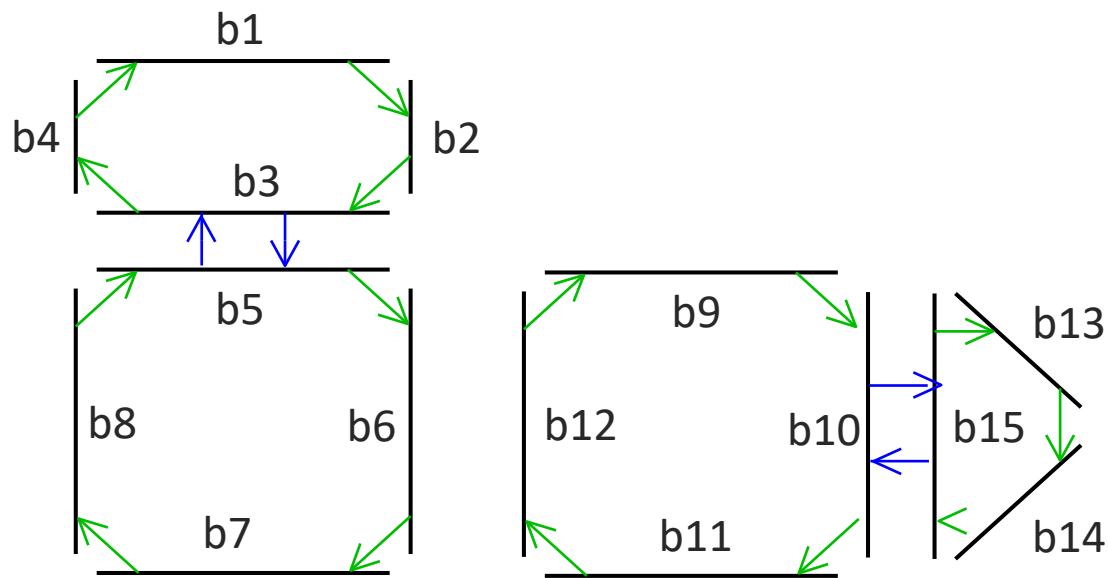
$\beta_2$  est une involution (bijection de  $B$  vers  $B$  qui est sa propre réciproque) représentant le collage de faces.

## 2-cartes : définition

- Une 2-carte est un triplet  $(B, \beta_1, \beta_2)$  où :
  - $B$  est un ensemble fini de brins ;
  - $\beta_1$  est une permutation (1) ;
  - $\beta_2$  est une involution avec ou sans point fixe (2) ;
- (1)  $\forall b, \exists p, \beta_1(b)^p = b$  (permutation) ;
- (2)  $\forall b \beta_2(\beta_2(b)) = b$  (involution) ; Si  $\beta_2(b) = b$  alors  $b$  est un point fixe par  $\beta_2$

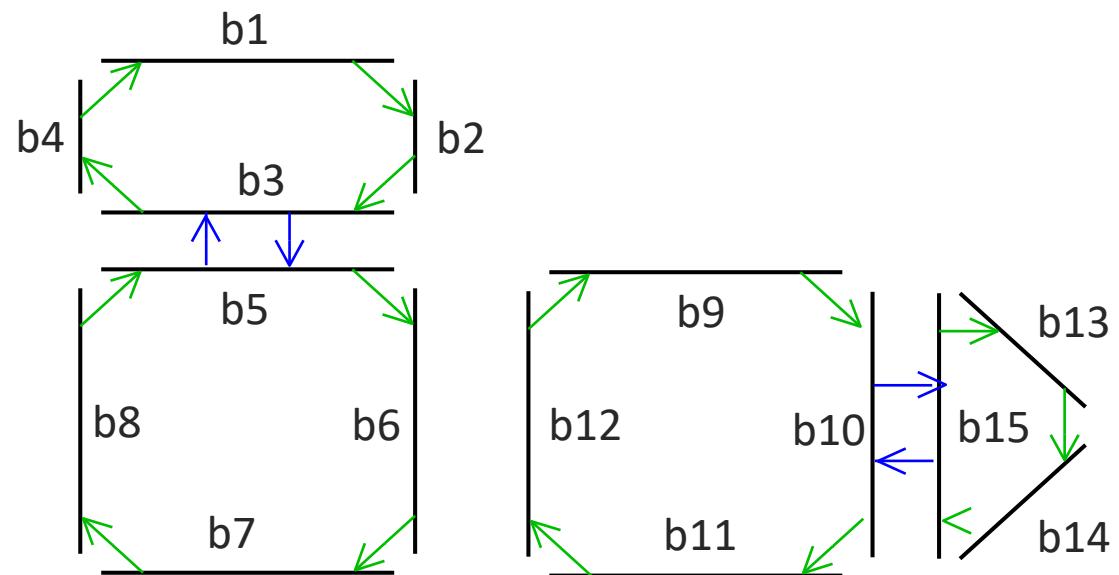
# Composante Connexe

- Une *composante connexe* d'une 2-carte  $(B, \beta_1, \beta_2)$  est l'ensemble des brins que l'on peut obtenir à partir de l'un d'eux par application de  $\beta_1$  et  $\beta_2$  dans un ordre quelconque.
- Exemple : Ici la 1-G-carte possède 2 composantes connexes :
  - $\{b_1, b_2, \dots, b_7, b_8\}$  et  $\{b_9, b_{10}, \dots, b_{14}, b_{15}\}$



# Orbite

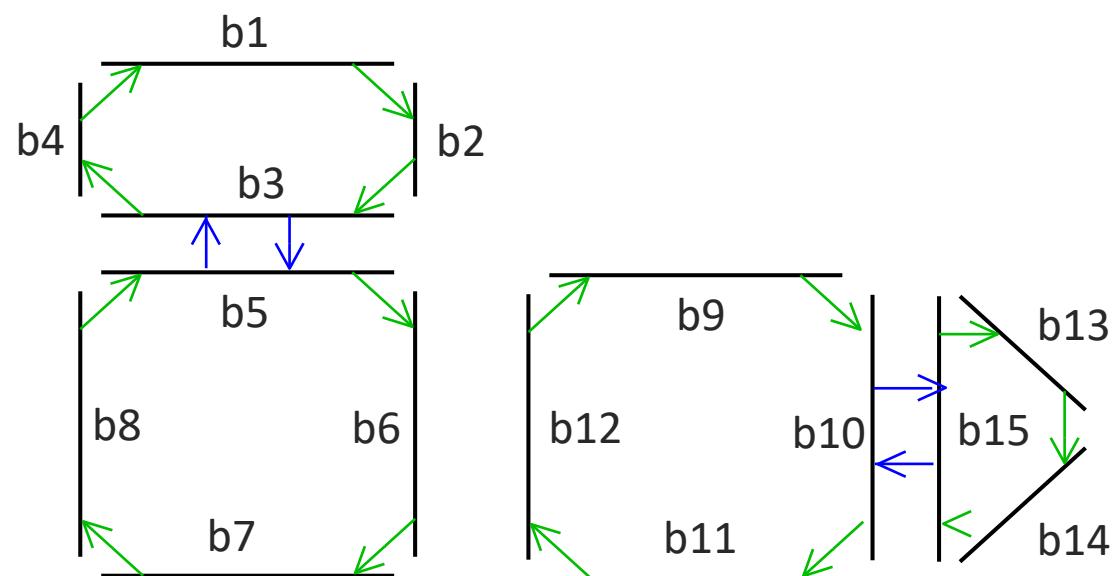
- Plus généralement, on définit *l'orbite* d'un brin  $b$  par rapport à un ensemble d'involutions  $\beta_1, \dots, \beta_j$  comme l'ensemble des brins que l'on peut obtenir à partir de  $b$  par application de  $\beta_1, \dots, \beta_j$  dans un ordre quelconque.  
On note cette orbite  $\langle \beta_1, \dots, \beta_j \rangle(b)$
- Exemples :
  - $\langle \beta_1, \beta_2 \rangle(b_1) = ?$
  - $\langle \beta_1, \beta_2 \rangle(b_{12}) = ?$



# Orbite

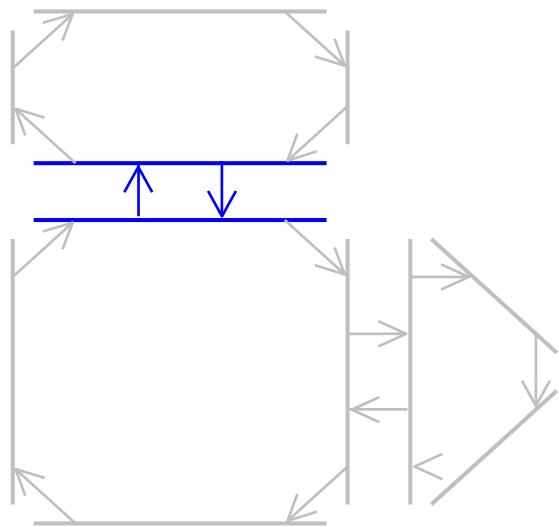
- Plus généralement, on définit *l'orbite* d'un brin  $b$  par rapport à un ensemble d'involutions  $\beta_1, \dots, \beta_j$  comme l'ensemble des brins que l'on peut obtenir à partir de  $b$  par application de  $\beta_1, \dots, \beta_j$  dans un ordre quelconque.  
On note cette orbite  $\langle \beta_1, \dots, \beta_j \rangle(b)$

- Exemples :
  - $\langle \beta_1, \beta_2 \rangle(b_1) = b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8$
  - $\langle \beta_1, \beta_2 \rangle(b_{12}) = b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}$

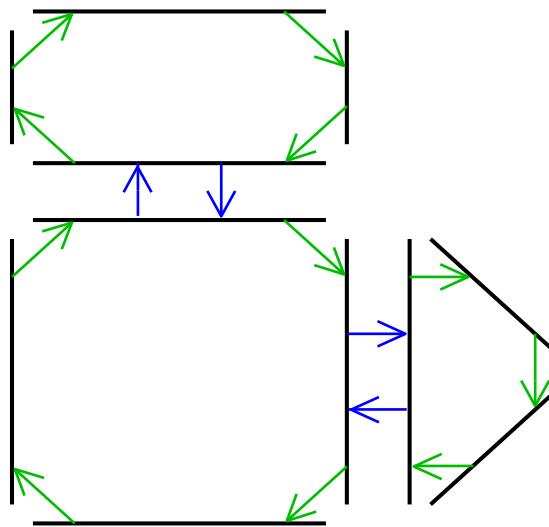


# Obtention intuitive d'une carte combinatoire

Les cellules sont des **orbites** de brins



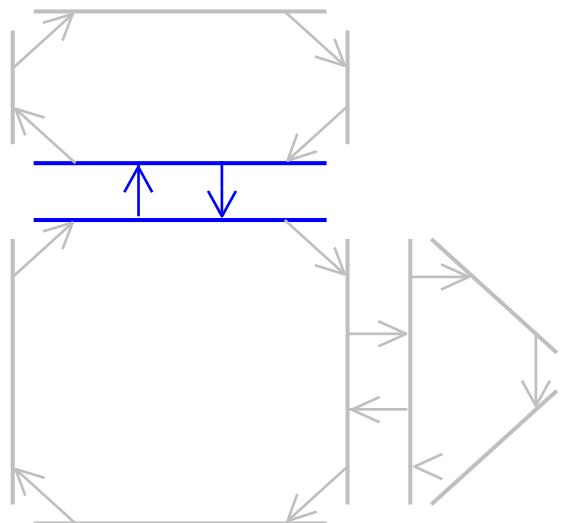
Arêtes (1-cellules) :  
orbites d'un brin suivant  
 $\beta_2$



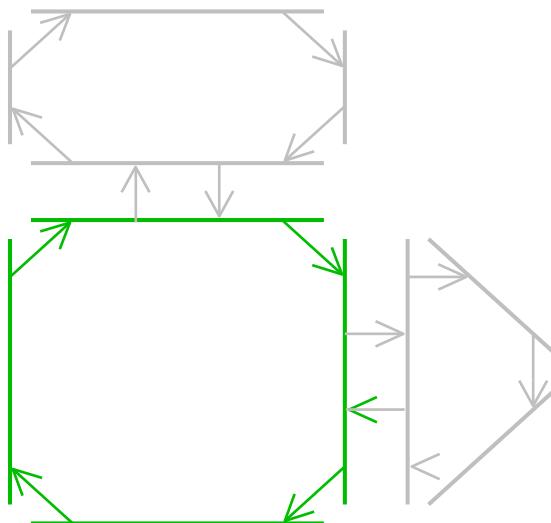
Faces (2-cellules)?

# Obtention intuitive d'une carte combinatoire

Les cellules sont des **orbites** de brins



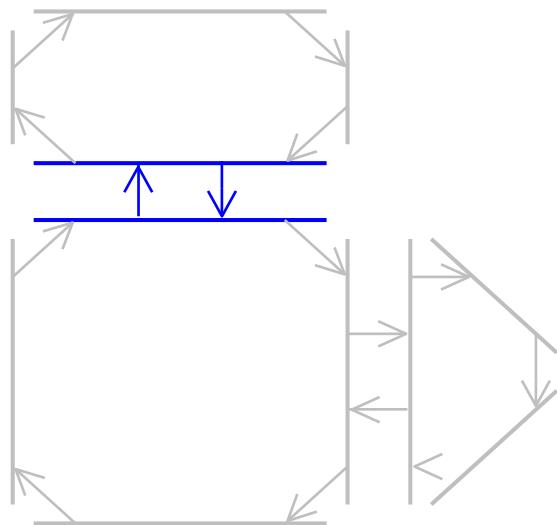
Arêtes (1-cellules) :  
orbites d'un brin suivant  
 $\beta_2$



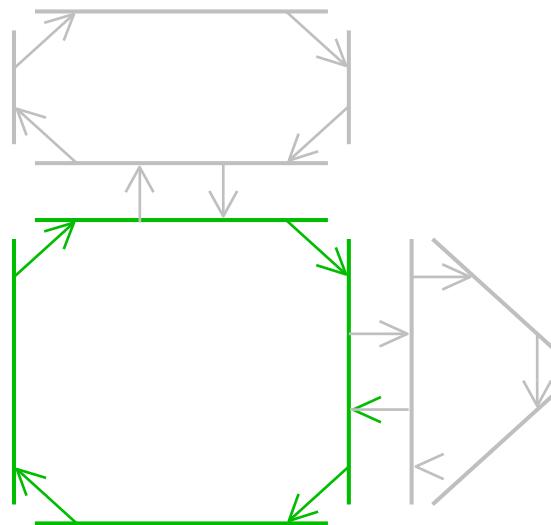
Faces (2-cellules) : orbites  
d'un brin suivant  $\beta_1$

# Obtention intuitive d'une carte combinatoire

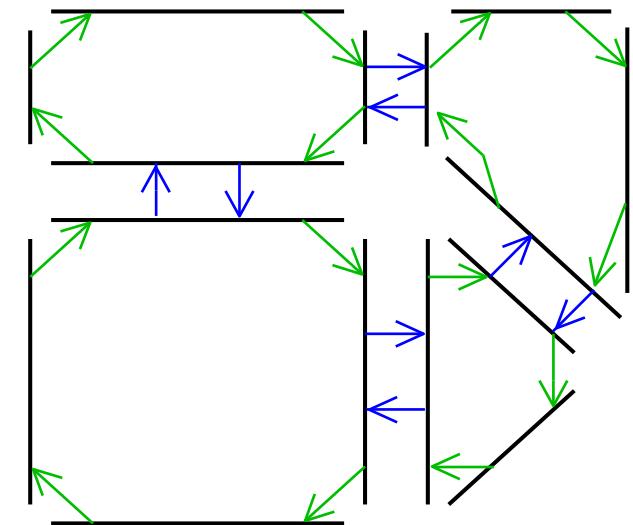
Les cellules sont des **orbites** de brins



Arêtes (1-cellules) :  
orbites d'un brin suivant  
 $\beta_2$



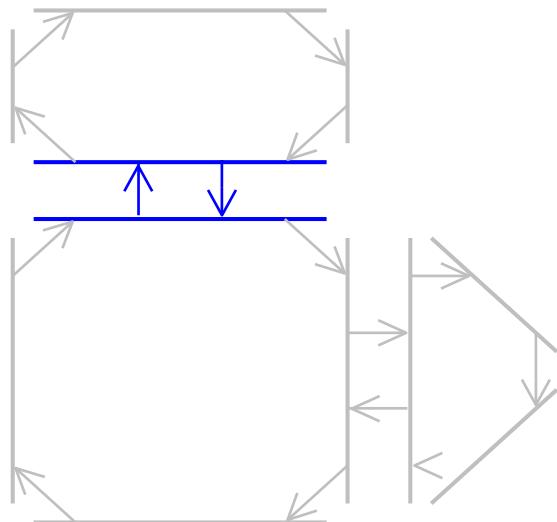
Faces (2-cellules) : orbites  
d'un brin suivant  $\beta_1$



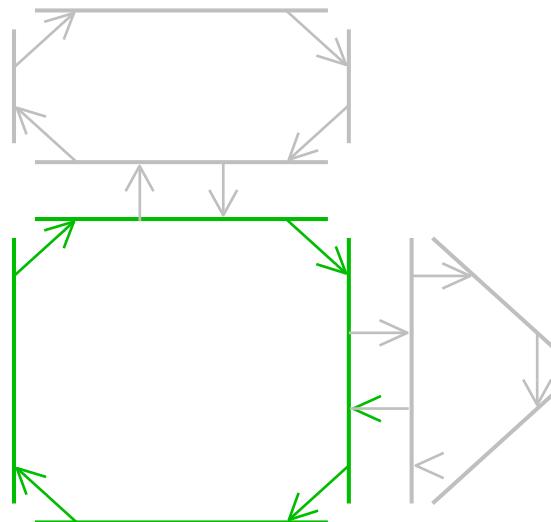
Sommets (0-cellules) : orbites  
plus complexes...  
(si relations définies partout)

# Obtention intuitive d'une carte combinatoire

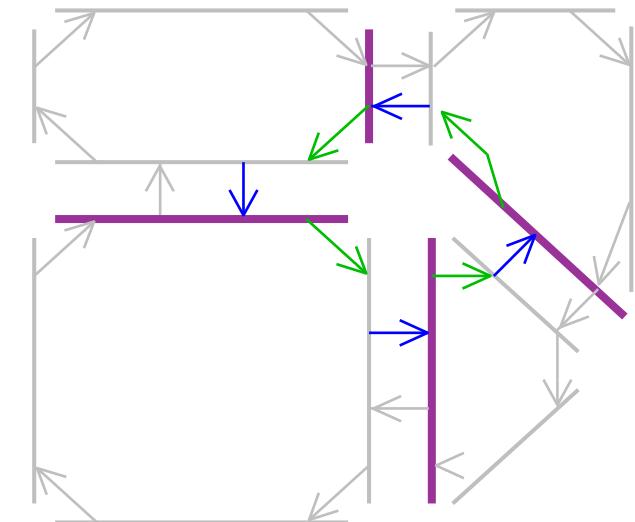
Les cellules sont des **orbites** de brins



Arêtes (1-cellules) :  
orbites d'un brin suivant  
 $\beta_2$



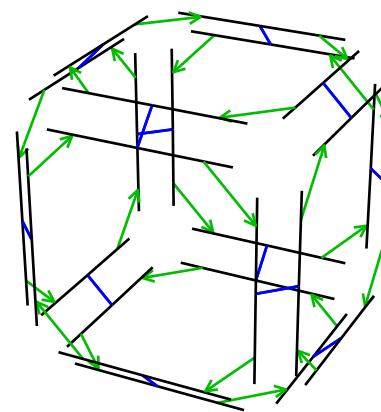
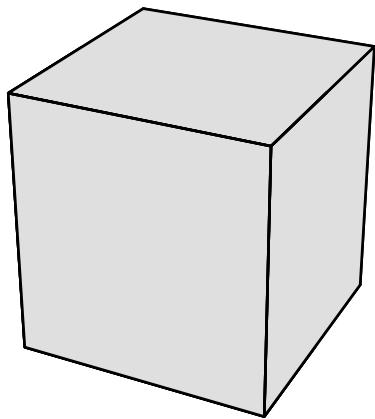
Faces (2-cellules) : orbites  
d'un brin suivant  $\beta_1$



Sommets (0-cellules) : orbites  
d'un brin suivant  $\beta_2 \circ \beta_1$

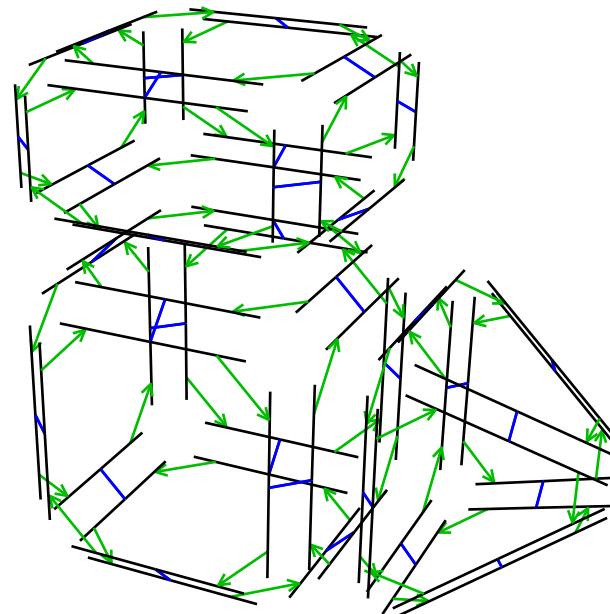
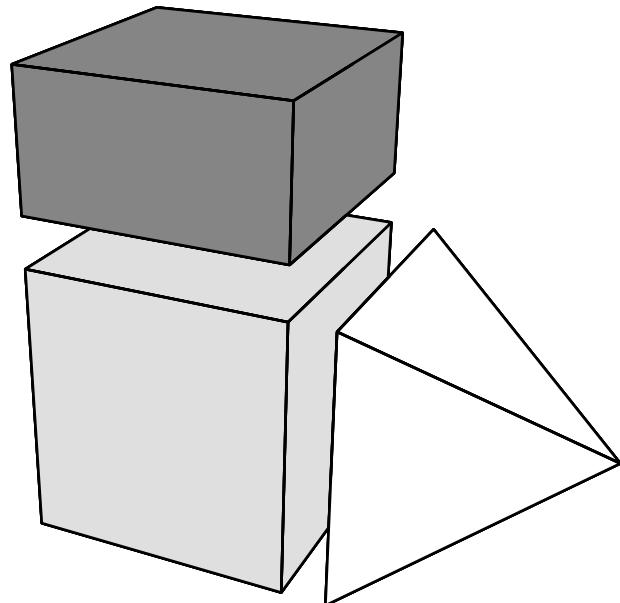
## Cartes combinatoires de dimension 3 (3-Cartes)

2-Carte (fermée) avec un plongement dans  $\mathbb{R}^3$  : représentation d'un volume (3-cellule)



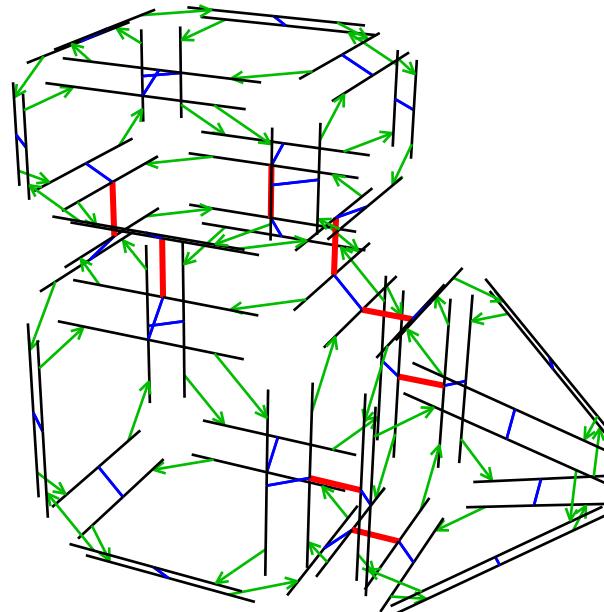
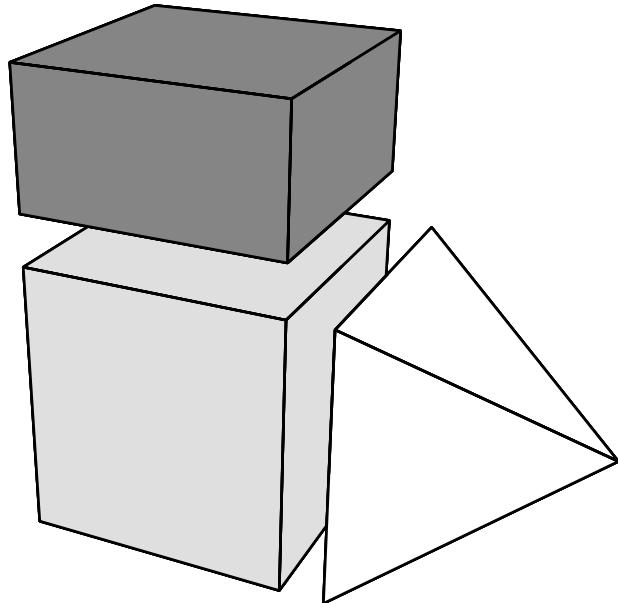
## Cartes combinatoires de dimension 3 (3-Cartes)

Ensemble de 2-Cartes : chacune représente un sous-volume d'un objet 3D



# Cartes combinatoires de dimension 3 (3-Cartes)

3-Carte : ensemble de 2-cartes reliées entre elles

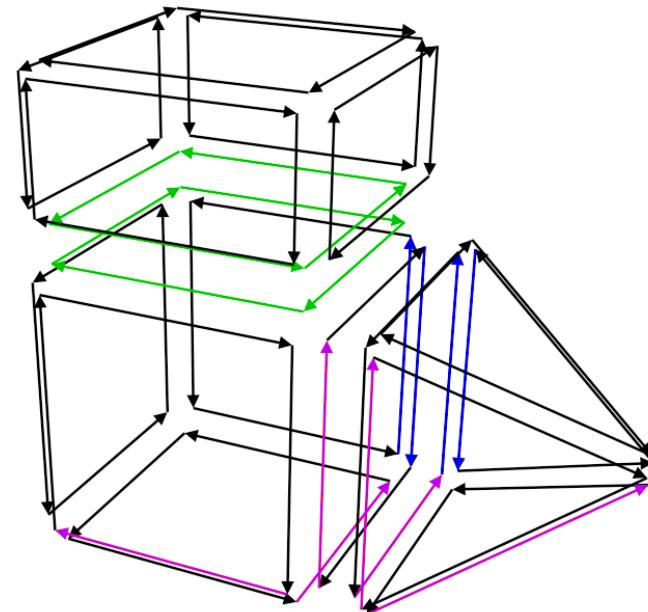
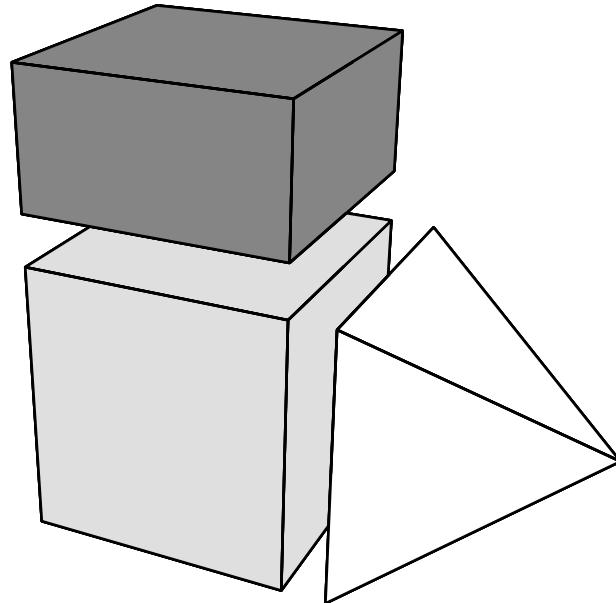


Ajout de la relation  $\beta_3$  : relie les volumes adjacents par leurs faces

- Ajout de contraintes de cohérence pour une topologie valide

# Cartes combinatoires de dimension 3 (3-Cartes)

Ensemble de 2-cartes : chacune représente un volume



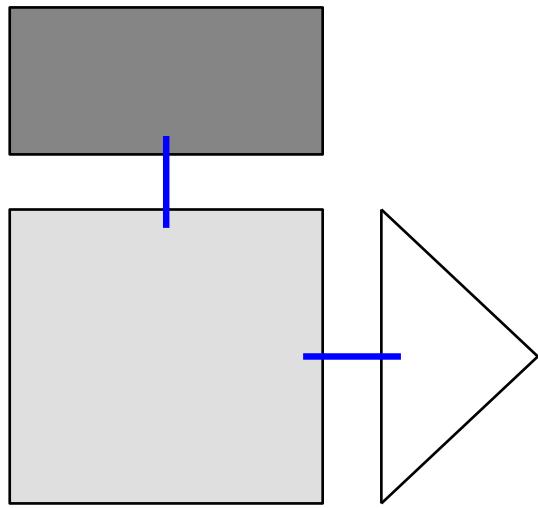
- Cellules comme des orbites de brins suivant plusieurs relations ( $\beta_1 + \beta_3$  pour les faces)
- Définition hétérogène :
  - Les relations sont toutes des **involutions** sauf  $\beta_1$  qui est une **permutation**
  - Les sommets (0-cellules) sont représentés par des orbites différentes

**Cartes combinatoires généralisées pour une définition plus homogène**

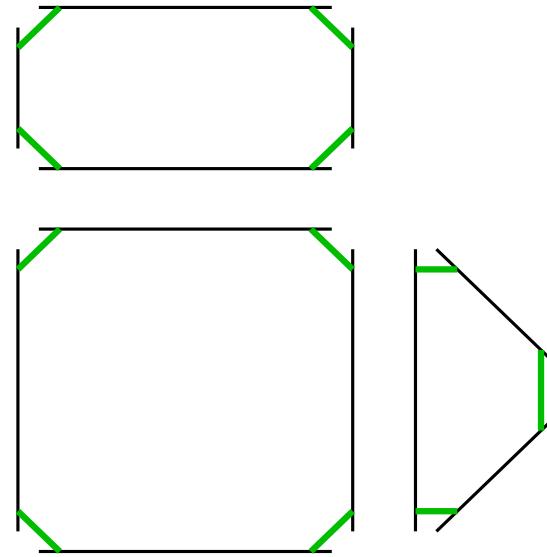
# Cartes généralisées (G-Cartes)

Introduction d'un niveau de décomposition supplémentaire

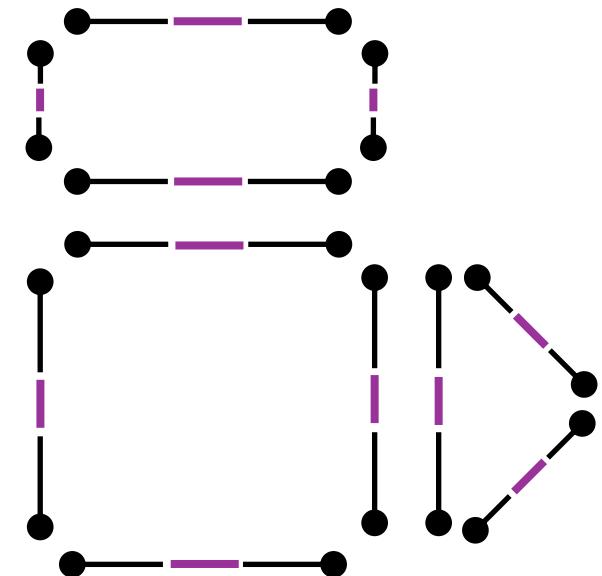
- Exemple en 2D :



Faces



Arêtes



Sommets  
⇒ Brins

**Les brins sont des éléments de "dimension" la plus petite possible**

# Cartes généralisées (G-Cartes)

Topologie de subdivisions d'objets réguliers de l'espace

- Orientables ou non
- Avec ou sans bord

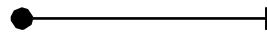
Définition générale des **n-G-Cartes** pour des objets de dimension n

- **Progressivement**, en commençant par les 1-G-cartes...

# Cartes généralisées de dimension 1 ou 1-G-Cartes

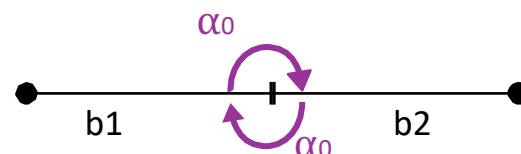
## Brins

Intuitivement un brin peut être vu (et être graphiquement représenté) comme une portion d'arête (ou une portion de sommet appartenant à une arête)



## Opérateurs d'assemblage $\alpha_0$ et $\alpha_1$

- La relation  $\alpha_0$  permet de relier deux brins entre eux pour former une arête.

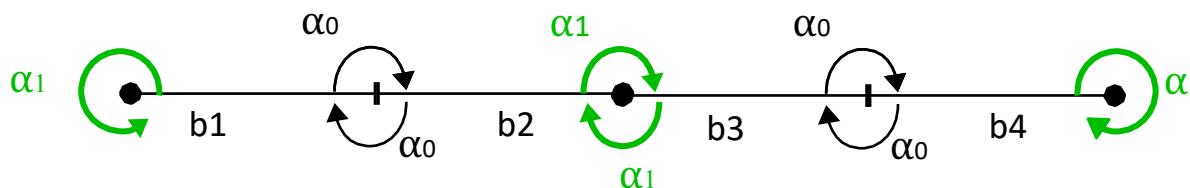


$$\begin{aligned}\alpha_0(b1) &= b2 \\ \alpha_0(b2) &= b1\end{aligned}$$

$\alpha_0$  est une **involution**

- La relation  $\alpha_1$  permet de relier deux arêtes entre elles.

$$\begin{aligned}\alpha_1(b1) &= b1 \\ \alpha_1(b2) &= b3 \\ \alpha_1(b3) &= b2 \\ \alpha_1(b4) &= b4\end{aligned}$$



$\alpha_1$  est une **involution**

## 1-G-Cartes : Définition

Une 1G-carte est un triplet  $(B, \alpha_0, \alpha_1)$  où :

- $B$  est un ensemble fini de brins ;
- $\alpha_0$  est une involution sans point fixe (1) ;
- $\alpha_1$  est une involution avec ou sans point fixe (2) ;

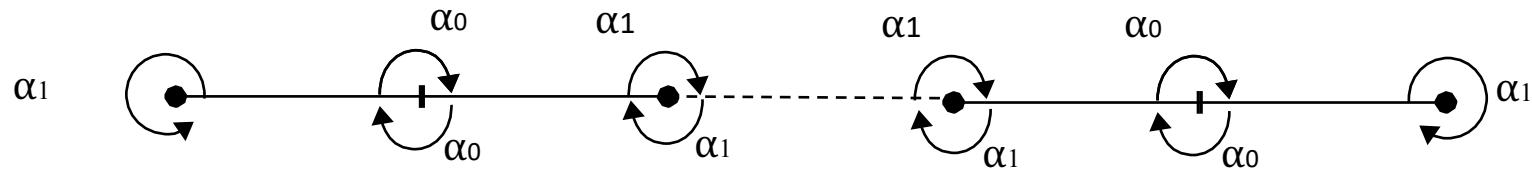
$$\forall b \in B, \alpha_0(\alpha_0(b)) = b \quad \alpha_0(b) \neq b \quad (1)$$

$$\forall b \in B, \alpha_1(\alpha_1(b)) = b \quad (2)$$

# 1-G-Cartes

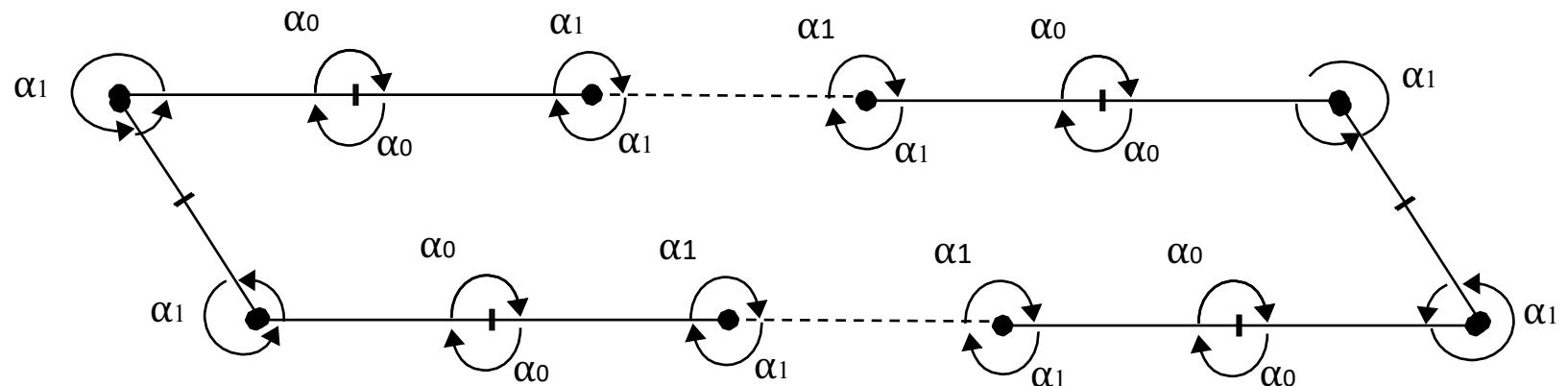
- Le modèle des 1-G-cartes permet de représenter :

- des arcs topologiques (1-cellule) :



Ici l'arc possède deux extrémités ( $\alpha_1$  possède 2 points fixes).

- ou des faces topologiques (2-cellule) :

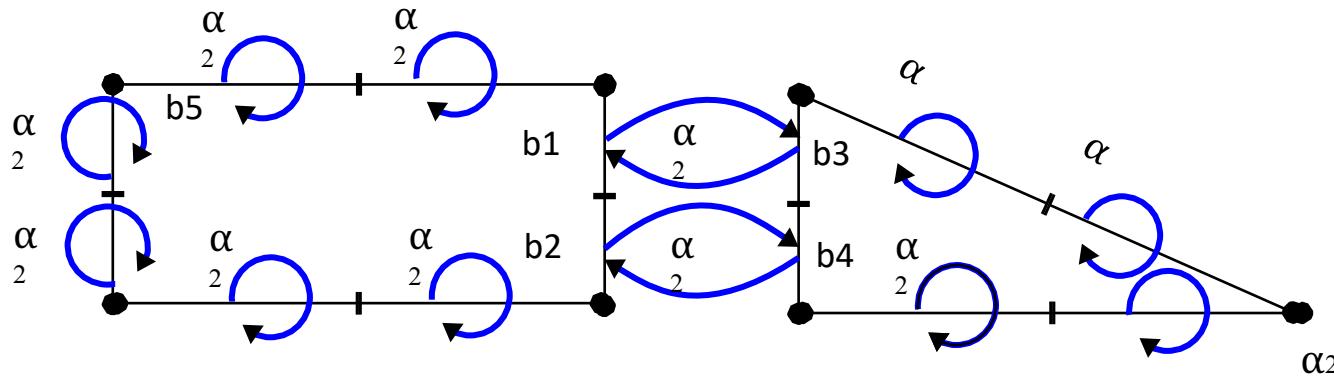


Ici  $\alpha_1$  ne possède pas de points fixes.

# Cartes généralisées de dimension 2 ou 2-G-Cartes

## Opérateur d'assemblage $\alpha_2$

- L'opérateur d'assemblage  $\alpha_2$  permet d'assembler 2 faces entre elles le long des arêtes de leur bord.
- On parle de **collage** ou de **couture** de faces entre elles. Exemple (ici les involutions  $\alpha_0$  et  $\alpha_1$  ne sont pas représentées) :



$$\alpha_2(b1) = b3$$

$$\alpha_2(b2) = b4$$

$$\alpha_2(b3) = b1$$

$$\alpha_2(b4) = b2$$

$$\alpha_2(b5) = \dots$$

## 2-G-Cartes : Définition

Une 2-G-Carte est un quadruplet  $(B, \alpha_0, \alpha_1, \alpha_2)$  où :

- $B$  est un ensemble fini de brins ;
- $\alpha_0, \alpha_1$  sont des involutions sans point fixe (1) ;
- $\alpha_2$  est une involution avec ou sans point fixe (2) ;
- $\alpha_0 \circ \alpha_2$  est une involution (3).

$$\forall b \in B, \alpha_0(\alpha_0(b)) = b \quad \alpha_0(b) \neq b$$

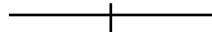
$$\forall b \in B, \alpha_1(\alpha_1(b)) = b \quad \alpha_1(b) \neq b \quad (1)$$

$$\forall b \in B, \alpha_2(\alpha_2(b)) = b \quad (2)$$

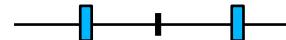
$$\forall b \in B, \alpha_0\left(\alpha_2\left( \alpha_0\left(\alpha_2(b)\right) \right)\right) = b \quad (3)$$

## 2-G-Cartes : Définition

Dans ce qui suit nous utiliserons une notation graphique simplifiée pour les involutions  $\alpha_0$ ,  $\alpha_1$  et  $\alpha_2$  :

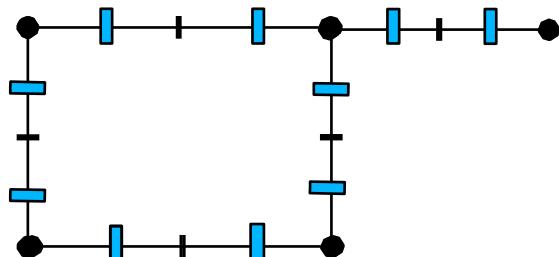
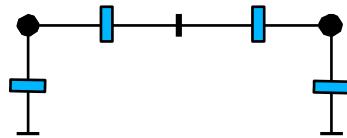
- $\alpha_0$  : 
- $\alpha_1$  : 
- $\alpha_2$  : 

avec point fixe :



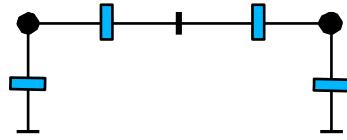
## 2-G-Cartes : Définition

Validité des 2-G-Cartes suivantes?



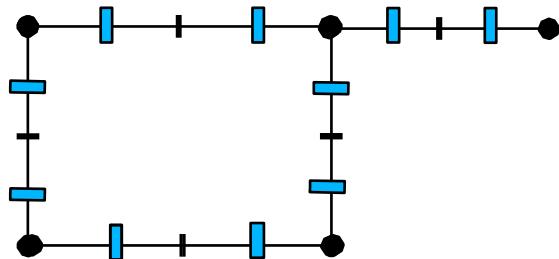
## 2-G-Cartes : Définition

Validité des 2-G-Cartes suivantes?



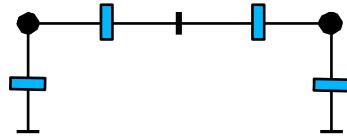
Non-valide selon (1)

$$\exists b \in B \mid \alpha_0(b) = b$$



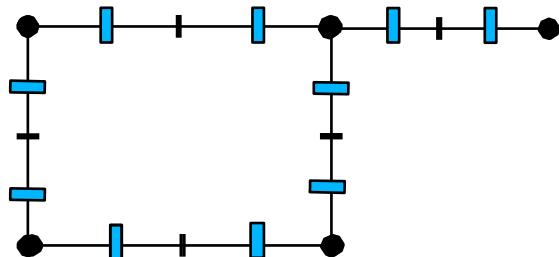
## 2-G-Cartes : Définition

Validité des 2-G-Cartes suivantes?



Non-valide selon (1)

$$\exists b \in B \mid \alpha_0(b) = b$$

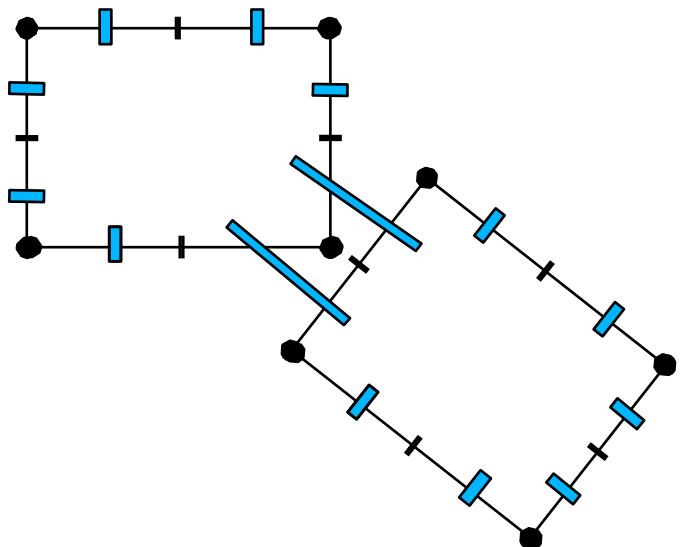
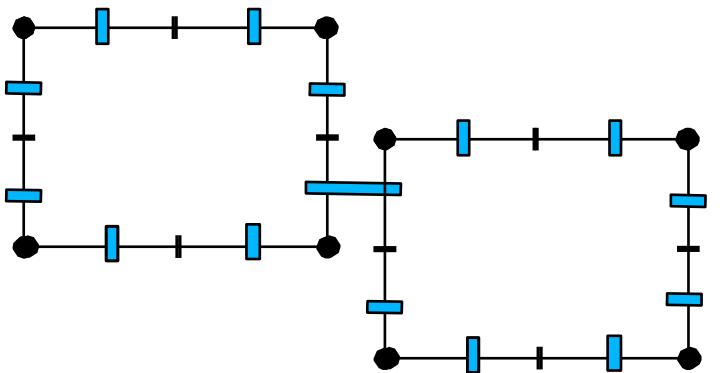


Non-valide selon (1)

$$\exists b \in B \mid \alpha_1(b) = b$$

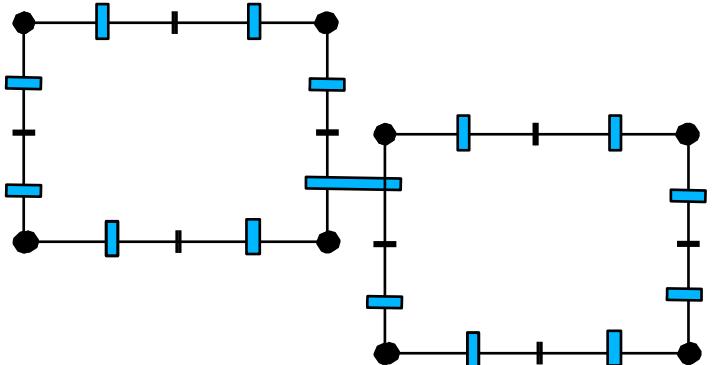
## 2-G-Cartes : Définition

Validité des 2-G-Cartes suivantes?



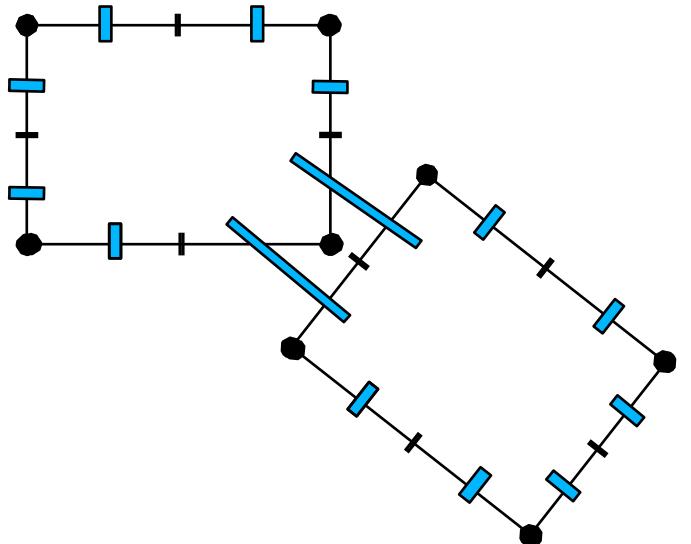
## 2-G-Cartes : Définition

Validité des 2-G-Cartes suivantes?



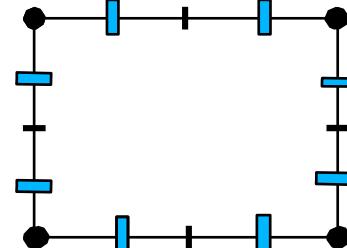
Non-valides selon (3)

$$\exists b \in B \mid \alpha_0 \circ \alpha_2 \circ \alpha_0 \circ \alpha_2(b) \neq b$$

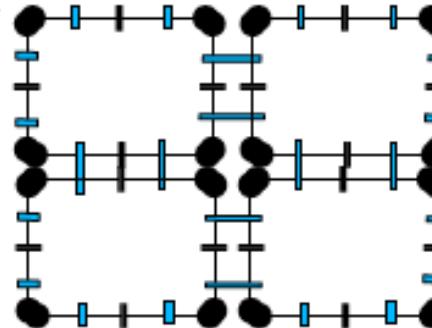


## 2-G-Cartes : Exemples

- Exemple 1 : Subdivision de surface avec bords et orientable ; surfaces homéomorphes à un disque topologique.

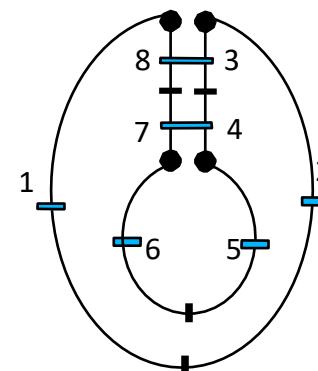
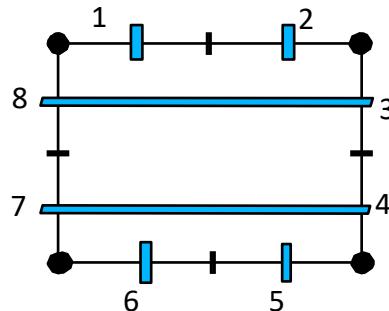


Avec une seule face.



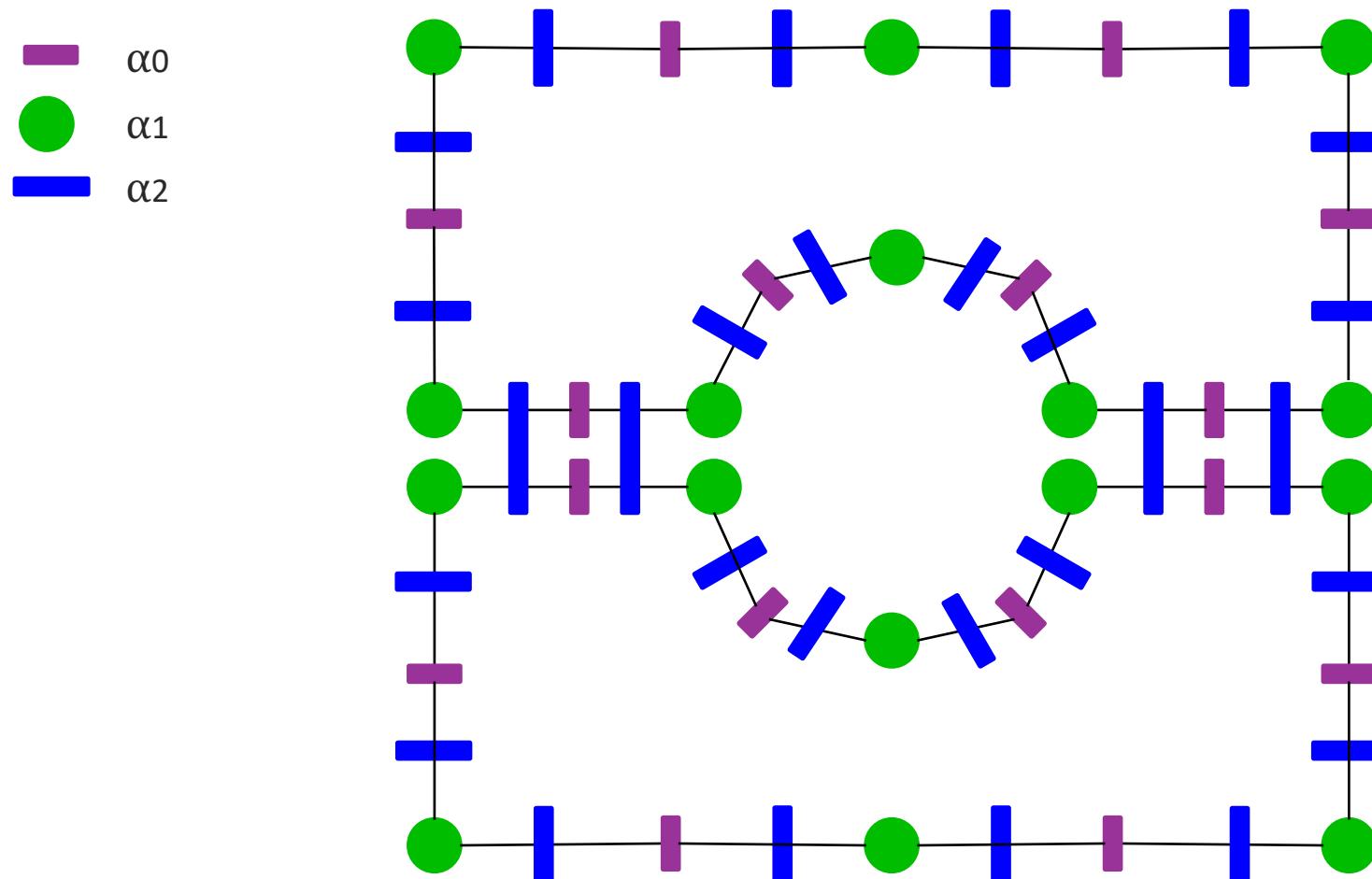
Avec 4 faces

- Exemple 2 : Subdivision de surface avec bords et orientable ; Surfaces homéomorphes à un anneau.



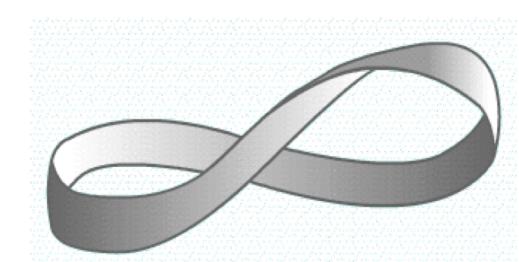
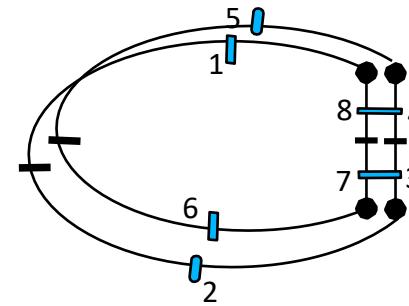
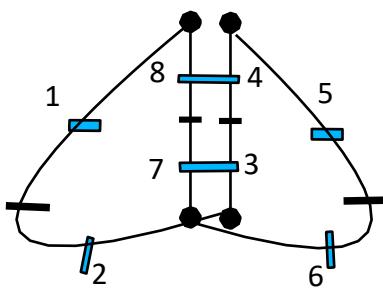
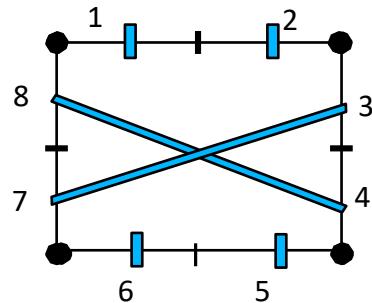
**Exercice :** Donner une 2-G-Carte représentant une subdivision d'une surface homéomorphe à un anneau et composée de 2 faces de degré 8 (le degré d'une face est égal à son nombre d'arêtes).

**Exercice :** Donner une 2-G-Carte représentant une subdivision d'une surface homéomorphe à un anneau et composée de 2 faces de degré 8 (le degré d'une face est égal à son nombre d'arêtes).

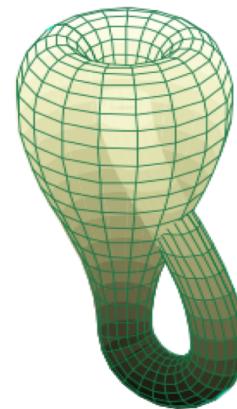
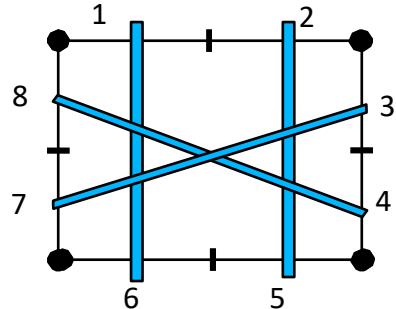


## 2-G-Cartes : Exemples

- Exemple 3 : Subdivision de surface avec bords et non orientable ; Surfaces homéomorphes à une bande de moebius.

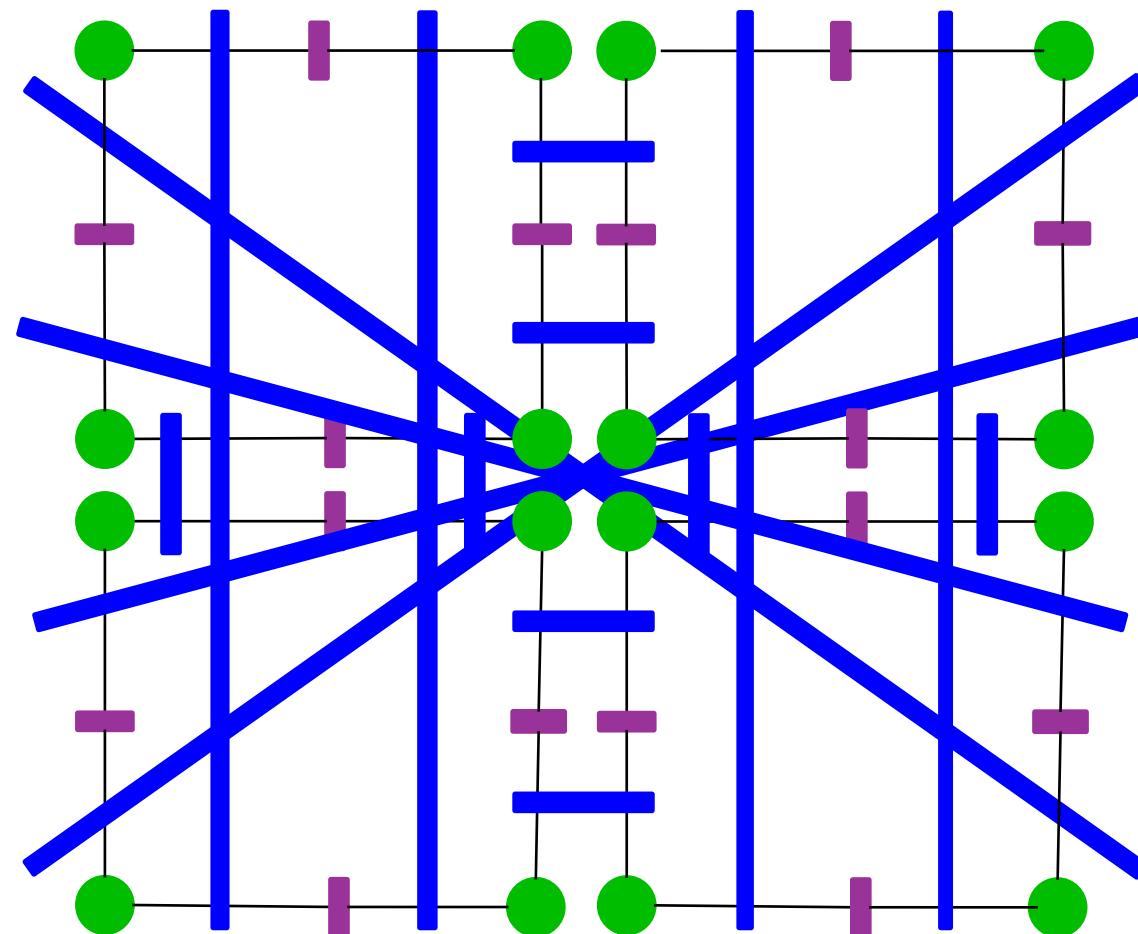


- Exemple 4 : Subdivision de surface sans bords et non orientable ; Surfaces homéomorphes à une bouteille de Klein.



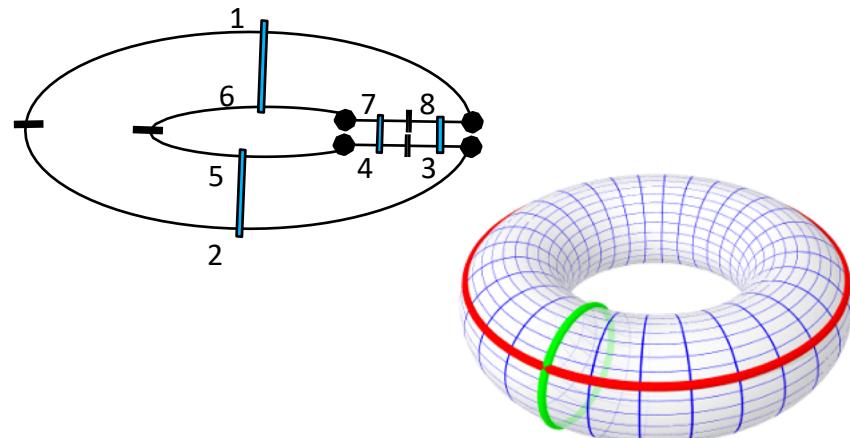
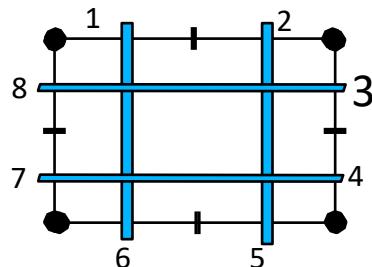
**Exercice :** Donner une 2-G-carte représentant une subdivision d'une surface homéomorphe à une bouteille de klein et composée de 4 faces de degré 4.

**Exercice :** Donner une 2-G-carte représentant une subdivision d'une surface homéomorphe à une bouteille de klein et composée de 4 faces de degré 4.

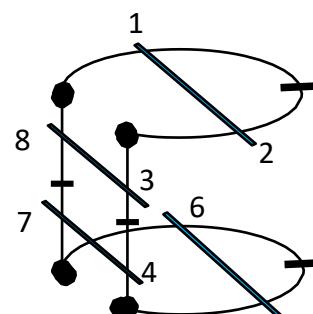
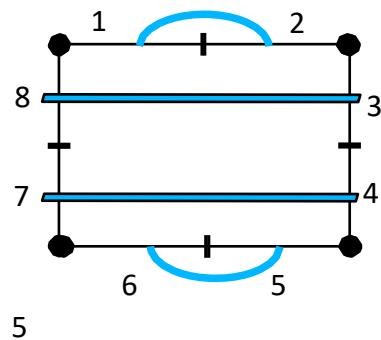


## 2-G-Cartes : Exemples

- Exemple 5 : Subdivision de surface sans bords et orientable ; Surfaces homéomorphes à un tore.



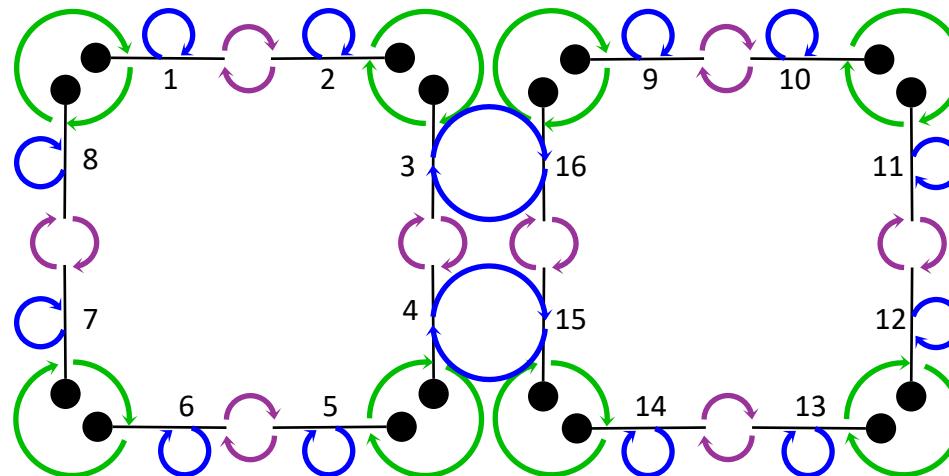
- Exemple 6 : Subdivision de surface sans bords et orientable ; Surfaces homéomorphes à une sphère



## 2-G-Cartes : Cellules et Orbites

Les n-cellules sont toujours représentées par des **orbites** selon plusieurs relations

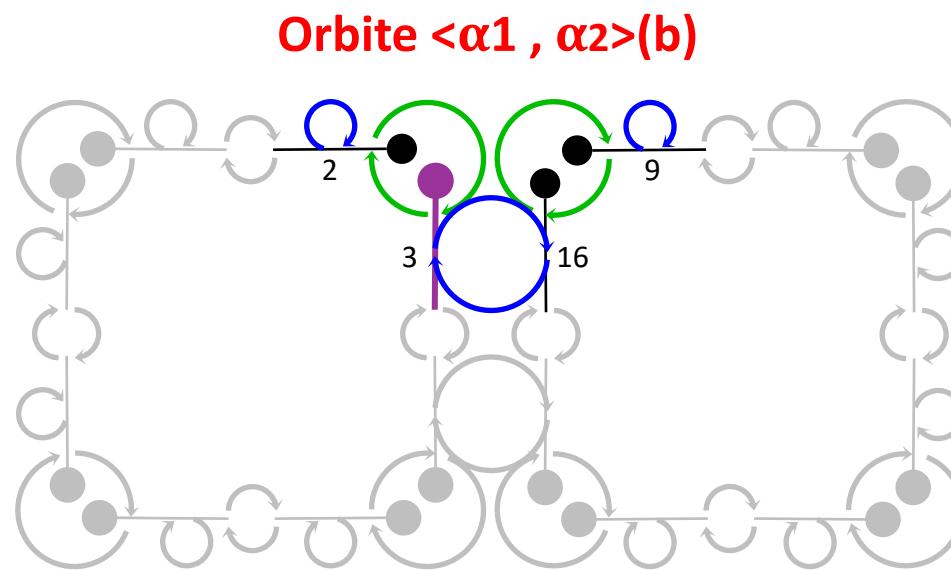
Ensemble de brins correspondant au **sommet (0-cellule)** incident à un brin b?



## 2-G-Cartes : Cellules et Orbites

Les n-cellules sont toujours représentées par des **orbites** selon plusieurs relations

Ensemble de brins correspondant au **sommet (0-cellule)** incident à un brin :



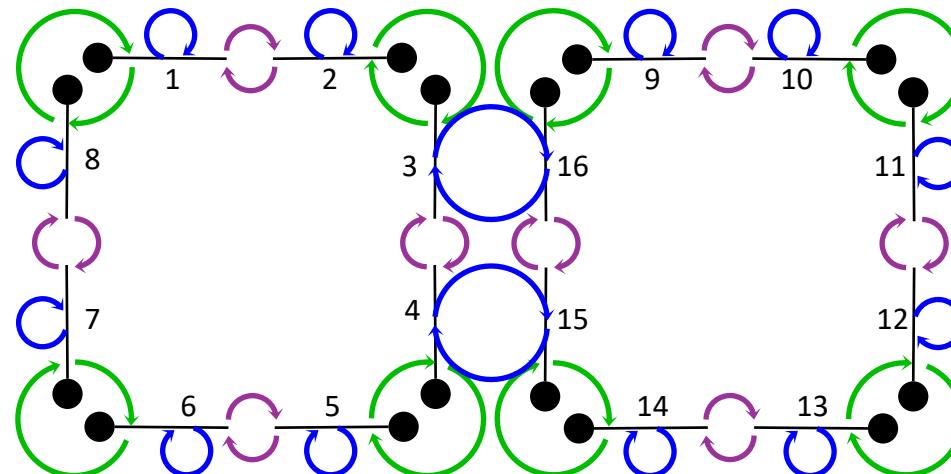
Sur l'exemple  $\langle \alpha_1, \alpha_2 \rangle(3) = \{2, 3, 16, 9\}$

Sommets d'une 2-G-Carte : **composantes connexes** de la 1-G-Carte  $(B, \alpha_1, \alpha_2)$

## 2-G-Cartes : Cellules et Orbites

Les n-cellules sont toujours représentées par des **orbites** selon plusieurs relations

Ensemble de brins correspondant à l'arête (**1-cellule**) incidente à un brin b?

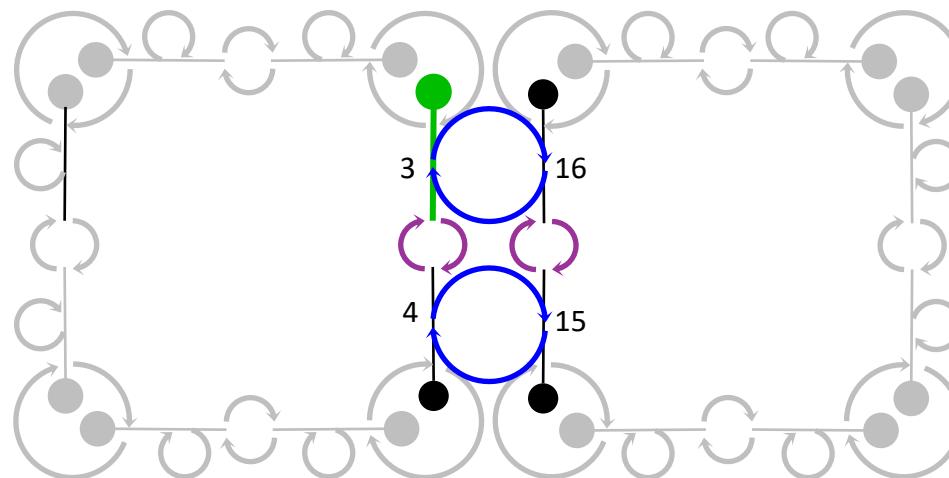


## 2-G-Cartes : Cellules et Orbites

Les n-cellules sont toujours représentées par des **orbites** selon plusieurs relations

Ensemble de brins correspondant à l'arête (**1-cellule**) incidente à un brin b :

**Orbite  $\langle\alpha_0, \alpha_2\rangle(b)$**



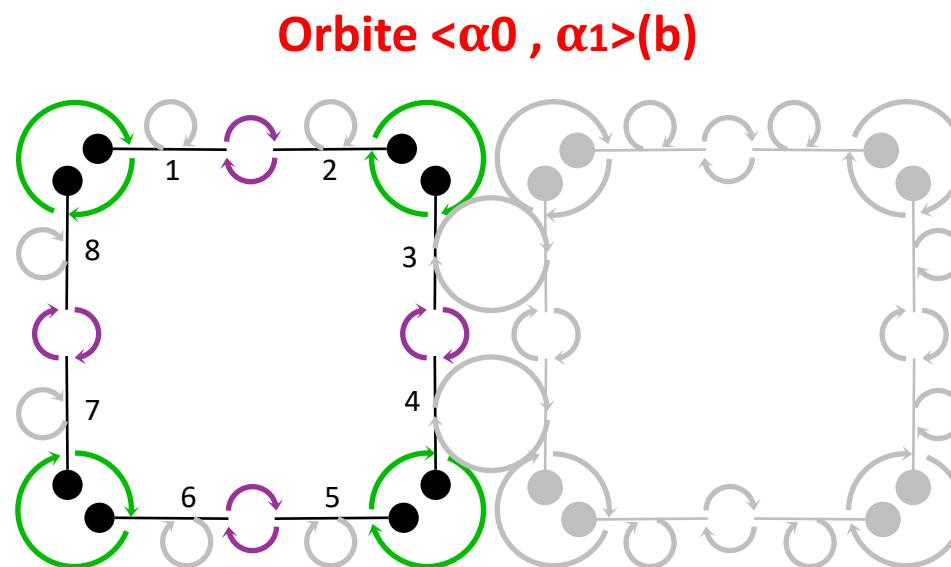
Sur l'exemple  $\langle\alpha_0, \alpha_2\rangle(3) = \{3, 4, 15, 16\}$

Arêtes d'une 2-G-Carte : **composantes connexes** de la 1-G-Carte  $(B, \alpha_0, \alpha_2)$

## 2-G-Cartes : Cellules et Orbites

Les n-cellules sont toujours représentées par des **orbites** selon plusieurs relations

Ensemble de brins correspondant à la **face (2-cellule)** incidente à un brin b :



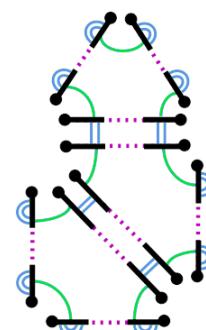
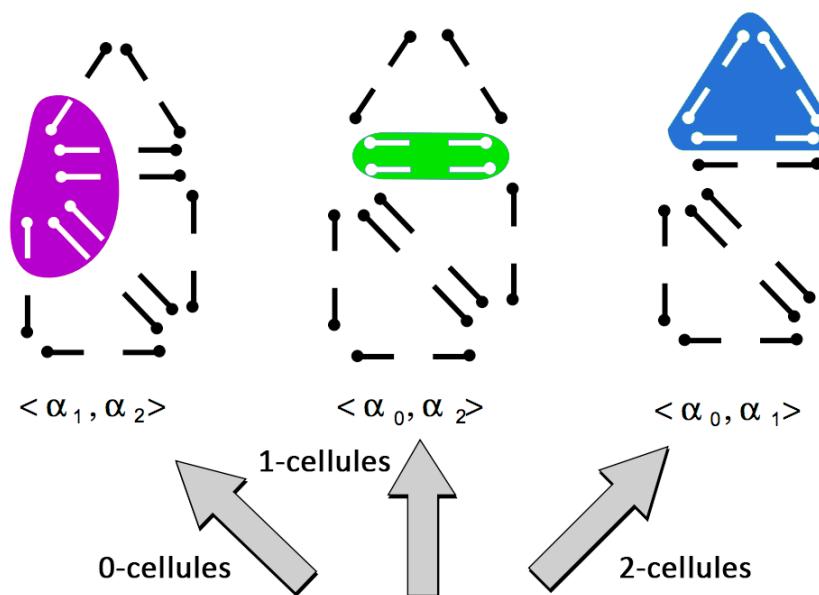
Sur l'exemple  $\langle \alpha_0, \alpha_1 \rangle(3) = \{1, 2, 3, 4, 5, 6, 7, 8\}$

Faces d'une 2-G-Carte : **composantes connexes** de la 1-G-Carte  $(B, \alpha_0, \alpha_1)$

## 2-G-Cartes : Cellules et Orbites

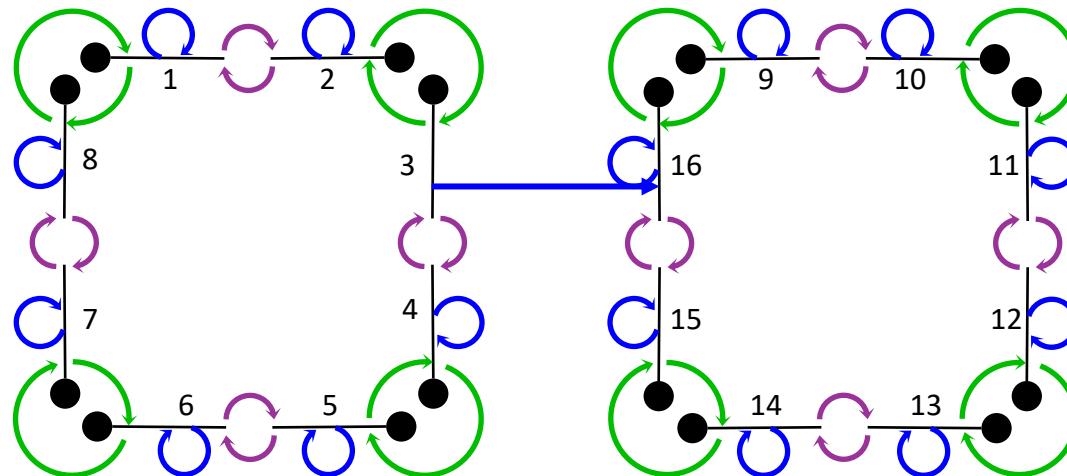
Définition générale :

Une i-cellule incidente à un brin b est définie par l'**orbite** de b utilisant toutes les involutions sauf  $\alpha_i$ .



## 2-G-Cartes : Couture de Faces

Assembler deux faces par une arête commune

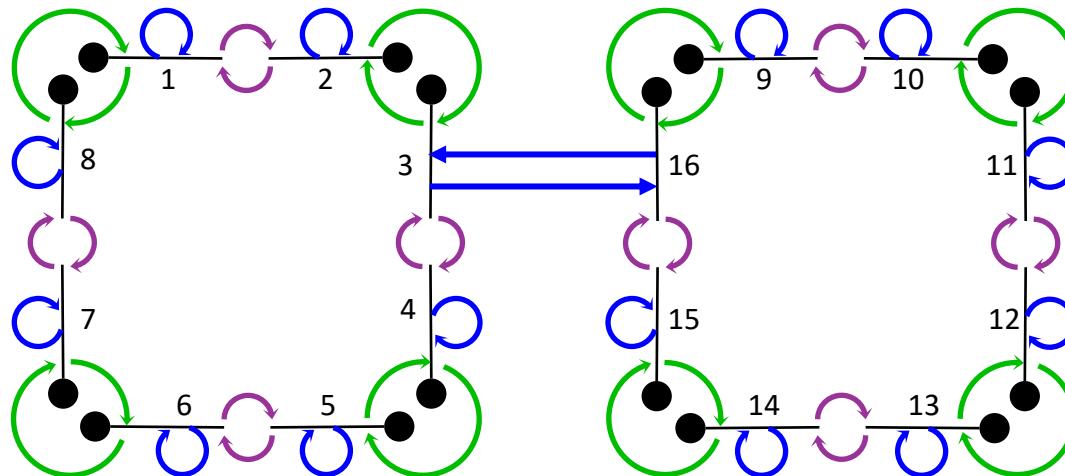


- Associer deux brins libres (exemple : 3 et 16)

$$\alpha_2(b) \leftarrow b'$$

## 2-G-Cartes : Couture de Faces

Assembler deux faces par une arête commune

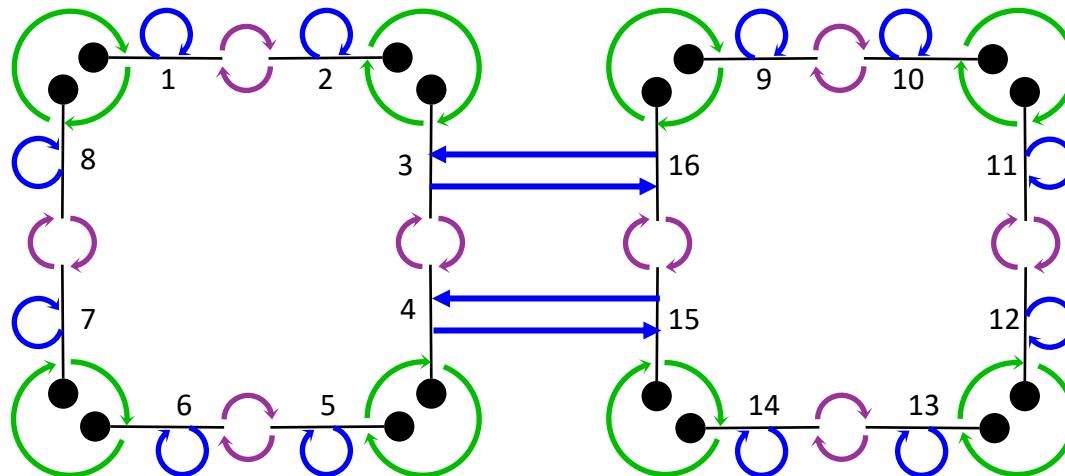


- Associer deux brins libres (exemple : 3 et 16)
- Assurer la contrainte  $\alpha_2$  involution

$$\begin{aligned}\alpha_2(b) &\leftarrow b' \\ \alpha_2(b') &\leftarrow b\end{aligned}$$

## 2-G-Cartes : Couture de Faces

Assembler deux faces par une arête commune

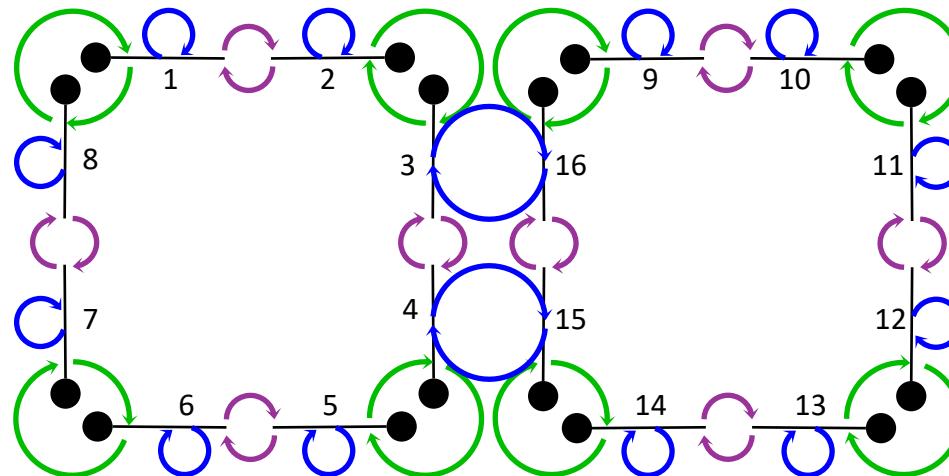


- Associer deux brins libres (exemple : 3 et 16)
- Assurer la contrainte  $\alpha_2$  involution
- Assurer les contraintes de cohérence  $\alpha_0 \circ \alpha_2$  involution

$$\begin{aligned}\alpha_2(b) &\leftarrow b' \\ \alpha_2(b') &\leftarrow b \\ \alpha_2(\alpha_0(b)) &\leftarrow \alpha_0(b') \\ \alpha_2(\alpha_0(b')) &\leftarrow \alpha_0(b)\end{aligned}$$

## 2-G-Cartes : Couture de Faces

Assembler deux faces par une arête commune



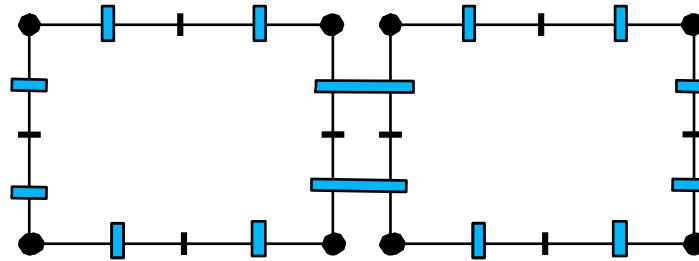
- Associer deux brins libres (exemple : 3 et 16)
- Assurer la contrainte  $\alpha_2$  involution
- Assurer les contraintes de cohérence  $\alpha_0 \circ \alpha_2$  involution

$$\begin{aligned}\alpha_2(b) &\leftarrow b' \\ \alpha_2(b') &\leftarrow b \\ \alpha_2(\alpha_0(b)) &\leftarrow \alpha_0(b') \\ \alpha_2(\alpha_0(b')) &\leftarrow \alpha_0(b)\end{aligned}$$

**Couture (dans les deux sens) par  $\alpha_2$  des orbites  $\langle\alpha_0\rangle$  de  $b$  et  $b'$**

## 2-G-Cartes : Bord

Définition du bord d'une 2-G-Carte



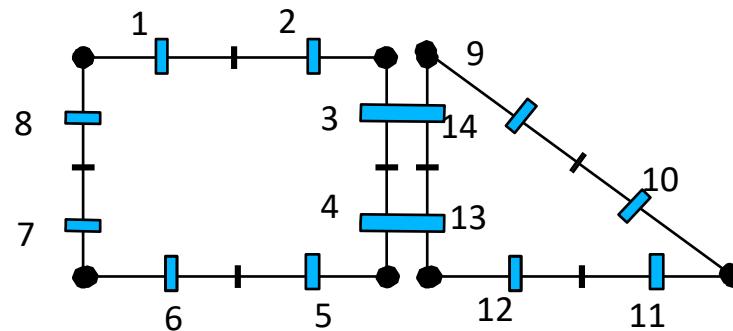
Le bord d'une 2-G-carte est constitué par l'ensemble des brins ayant un point fixe pour l'involution  $\alpha_2$ . On parle de brins libres par  $\alpha_2$ .

$$\delta B = \{b \in B \mid \alpha_2(b) = b\}$$

## **2-G-cartes : Bord**

Le bord d'une 2-G-Carte  $G(B, \alpha_0, \alpha_1, \alpha_2)$  est définie par la 1-G-Carte  $(B', \alpha'_0, \alpha'_1)$  où

- $B'$  est un sous-ensemble de  $B$  réduit aux brins du bord (tels que  $\alpha_2(b) = b$ ) ;
  - $\alpha'_0$  est la restriction de  $\alpha_0$  aux brins du bord ;
  - $\alpha'_1$  est la restriction de  $\alpha_1 \circ \alpha_2$  aux brins du bord ;

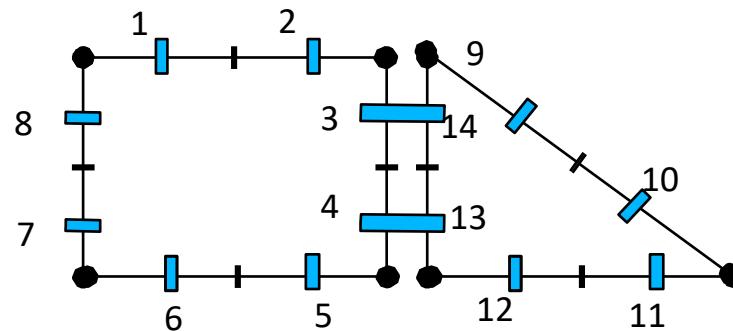


**Exercice :** donner la 1-G-Carte du bord de la 2-G-Carte ci-dessus.

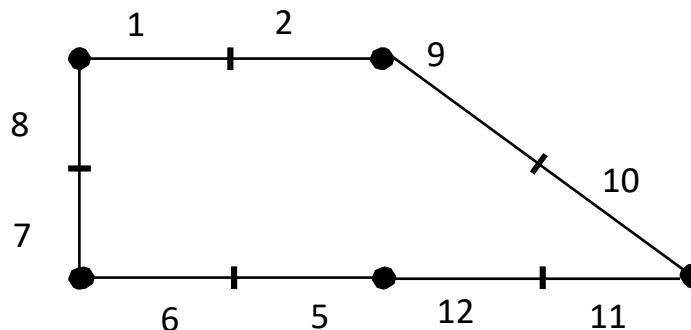
## **2-G-cartes : Bord**

Le bord d'une 2-G-Carte  $G(B, \alpha_0, \alpha_1, \alpha_2)$  est définie par la 1-G-Carte  $(B', \alpha'_0, \alpha'_1)$  où

- $B'$  est un sous-ensemble de  $B$  réduit aux brins du bord (tels que  $\alpha_2(b) = b$ ) ;
  - $\alpha'_0$  est la restriction de  $\alpha_0$  aux brins du bord ;
  - $\alpha'_1$  est la restriction de  $\alpha_1 \circ \alpha_2$  aux brins du bord ;



**Exercice :** donner la 1-G-Carte du bord de la 2-G-Carte ci-dessus.



## 2-G-Cartes : Exercice

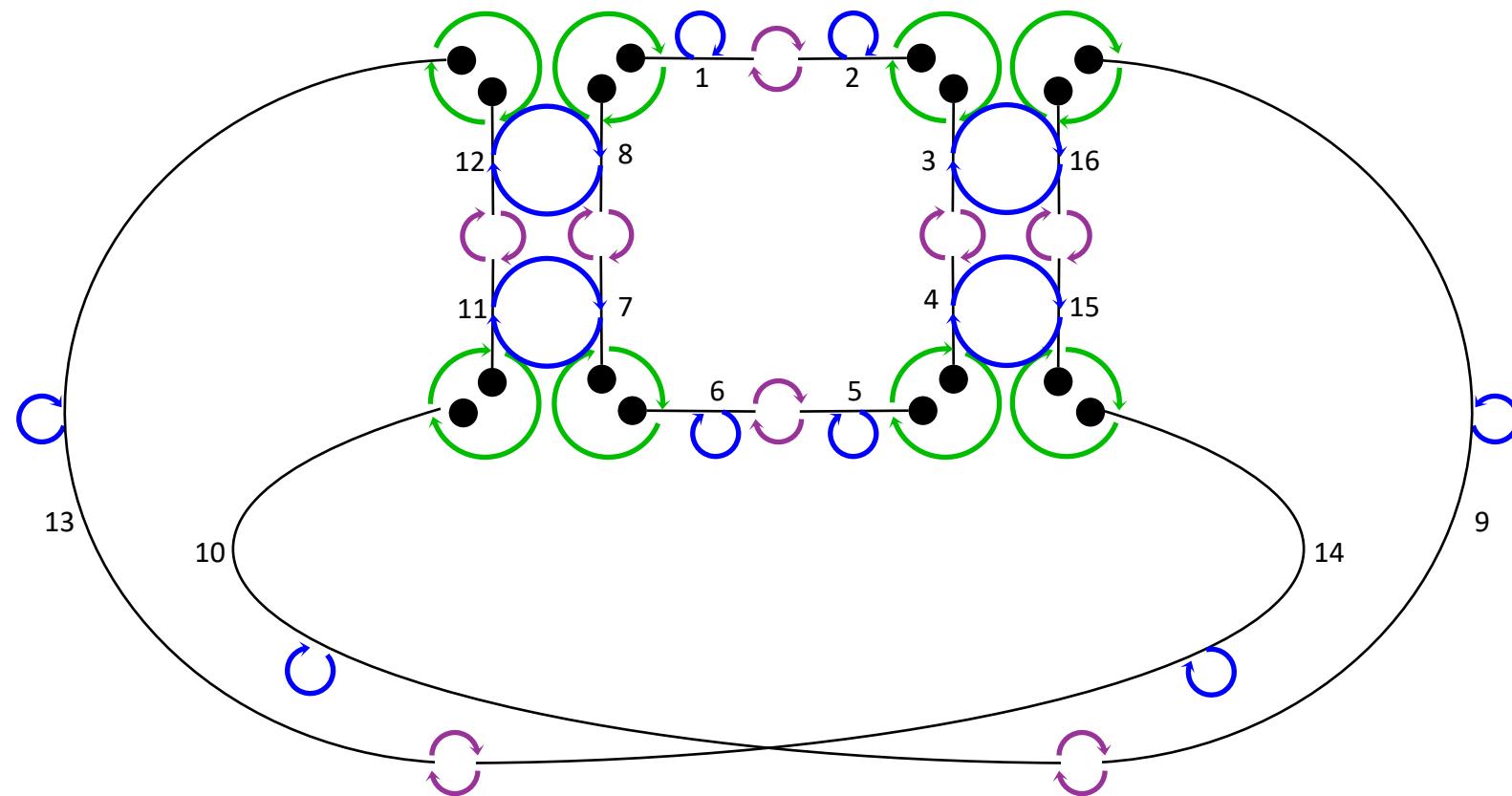
La 2-G-carte suivante est représentée par un tableau. Les brins sont numérotés de 1 à 16 et le tableau décrit les liens ( par exemple, la valeur 4 dans la case référencée  $[\alpha_0, 3]$  indique que  $\alpha_0(3) = 4$  ).

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>
$\alpha_0$	2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15
$\alpha_1$	8	3	2	5	4	7	6	1	16	11	10	13	12	15	14	9
$\alpha_2$	1	2	16	15	5	6	11	12	9	10	7	8	13	14	4	3

### Questions :

- Donnez une représentation graphique de la 2-G-cartes ci-dessus.
- Indiquez à quelle surface connue la surface représentée par cette carte est homéomorphe.

## 2-G-Cartes : Exercice



## 2-G-Cartes : Exercice

La 2-G-carte suivante est représentée par un tableau. Les brins sont numérotés de 1 à 16 et le tableau décrit les liens ( par exemple, la valeur 4 dans la case référencée  $[\alpha_0, 3]$  indique que  $\alpha_0(3) = 4$  ).

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>
$\alpha_0$	2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15
$\alpha_1$	8	3	2	5	4	7	6	1	16	11	10	13	12	15	14	9
$\alpha_2$	1	2	16	15	5	6	11	12	9	10	7	8	13	14	4	3

### Questions :

- Donnez une représentation graphique de la 2-G-cartes ci-dessus.
- Indiquez à quelle surface connue la surface représentée par cette carte est homéomorphe.
- On fait les modifications suivantes :  $\alpha_2(3) = 3 ; \alpha_2(4) = 4 ;$   
 $\alpha_2(15) = 15 ; \alpha_2(16) = 16 ;$

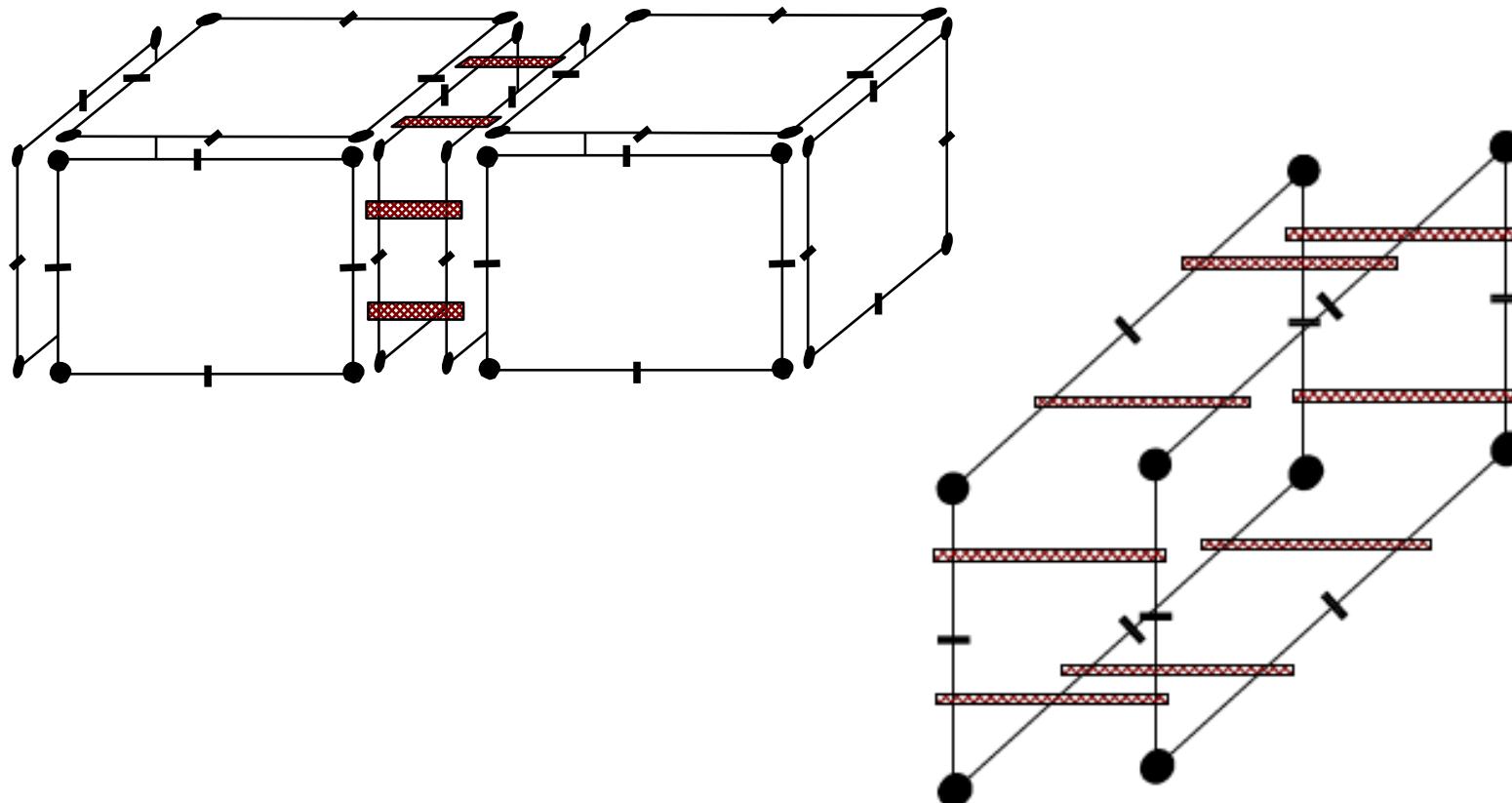
A quelle surface connue la 2-G-carte résultante est-elle homéomorphe ?

# **Cartes généralisées de dimension 3 ou 3-G-Cartes**

# Cartes généralisées de dimension 3 ou 3-G-Cartes

## Opérateur d'assemblage $\alpha_3$

L'opérateur d'assemblage  $\alpha_3$  permet d'assembler des volumes entre eux le long des faces de leur bords.



# Définition des 3-G-Cartes

Une 3G-carte est un n-uplet  $(B, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$

- $B$  est un ensemble fini de brins
- $\alpha_0, \alpha_1, \alpha_2$  sont des involutions sans point fixe (1)
- $\alpha_3$  est une involution avec ou sans point fixe (2)
- $\alpha_0 \circ \alpha_2, \alpha_0 \circ \alpha_3$  et  $\alpha_1 \circ \alpha_3$  sont des involutions (3)

$$\forall b \in B, \alpha_0(\alpha_0(b)) = b, \quad \alpha_0(b) \neq b$$

$$\forall b \in B, \alpha_1(\alpha_1(b)) = b, \quad \alpha_1(b) \neq b$$

$$\forall b \in B, \alpha_2(\alpha_2(b)) = b, \quad \alpha_2(b) \neq b$$

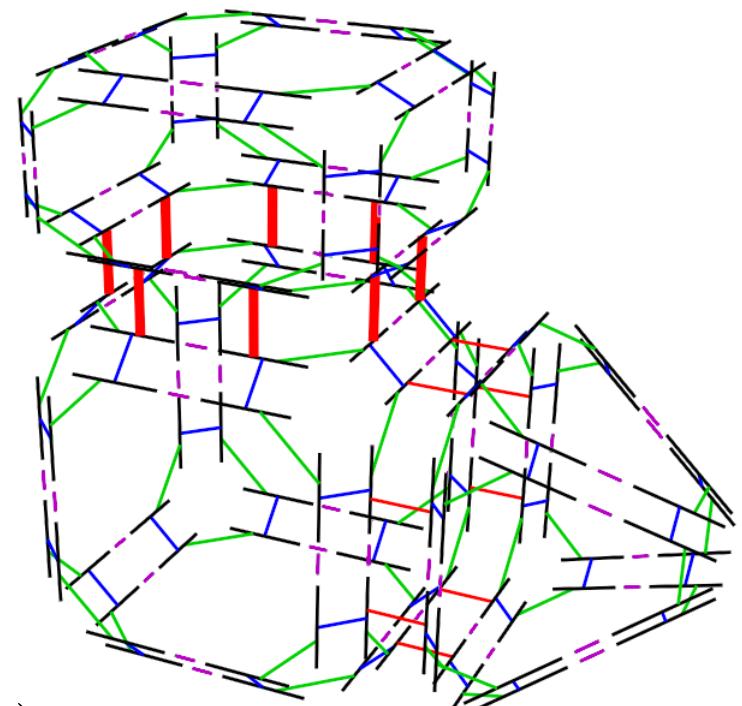
(1)

$$\forall b \in B, \alpha_3(\alpha_3(b)) = b \quad (2)$$

$$\forall b \in B, \alpha_0\left(\alpha_2\left(\alpha_0\left(\alpha_2(b)\right)\right)\right) = b$$

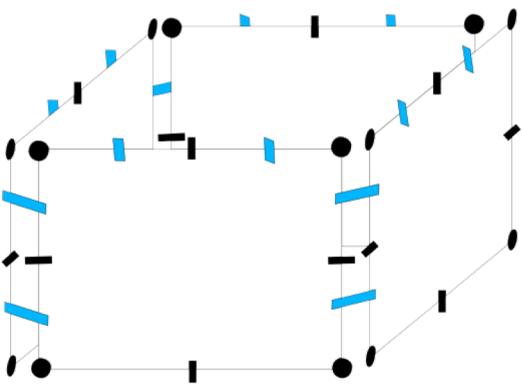
$$\forall b \in B, \alpha_0\left(\alpha_3\left(\alpha_0\left(\alpha_3(b)\right)\right)\right) = b$$

$$\forall b \in B, \alpha_1\left(\alpha_3\left(\alpha_1\left(\alpha_3(b)\right)\right)\right) = b \quad (3)$$



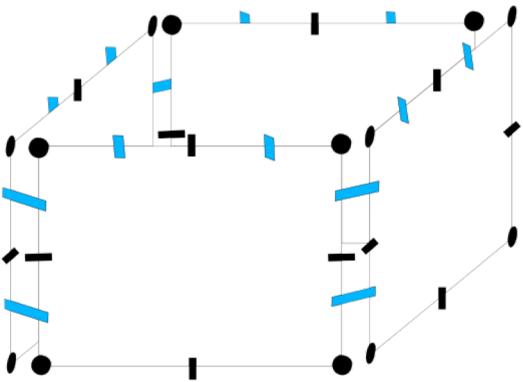
## 3-G-Cartes : Définition

Validité de la 3-G-Carte suivante?



## 3-G-Cartes : Définition

Validité de la 3-G-Carte suivante?

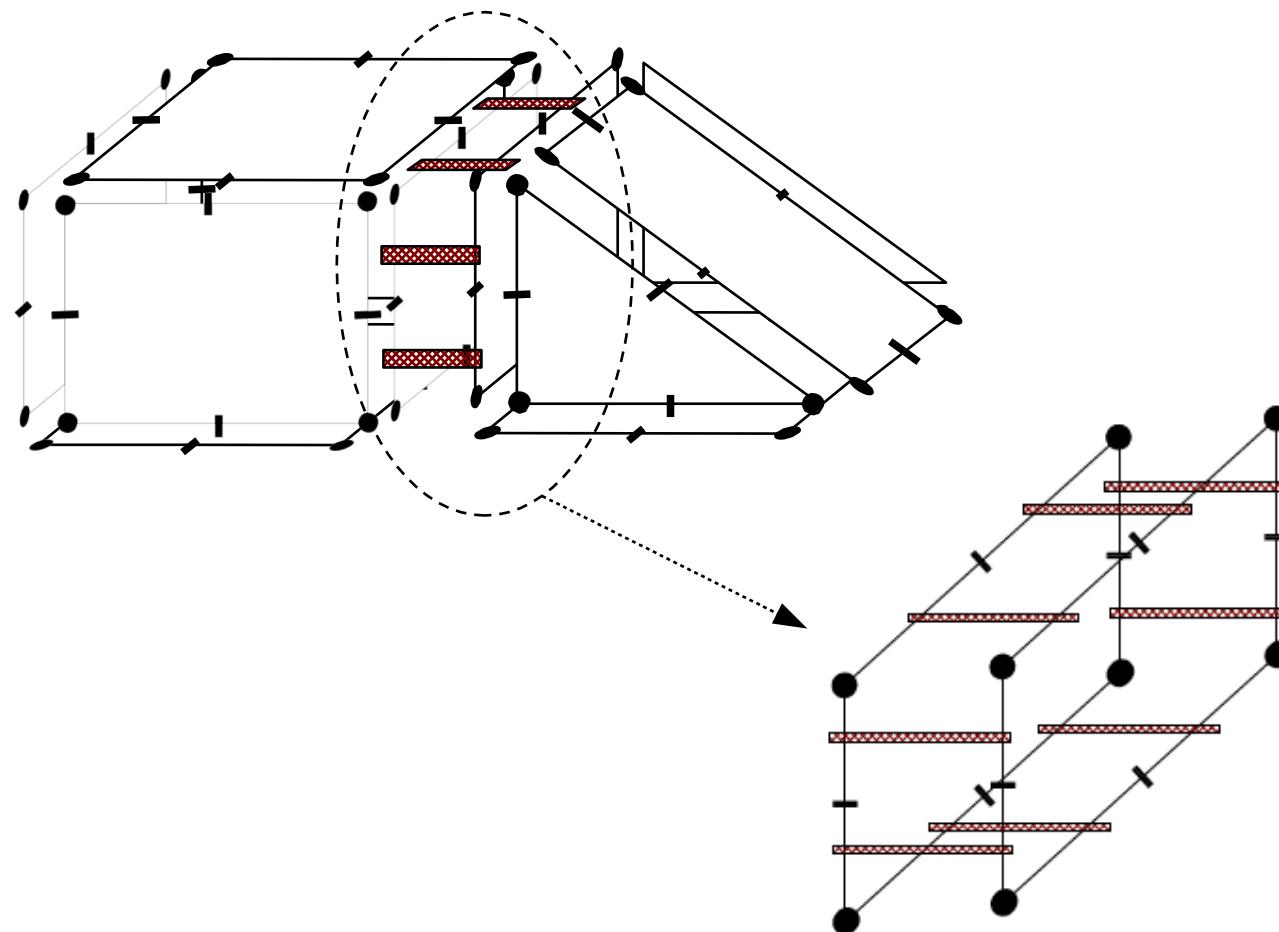


Non-valide selon (1)

$$\exists b \in B \mid \alpha_2(b) = b$$

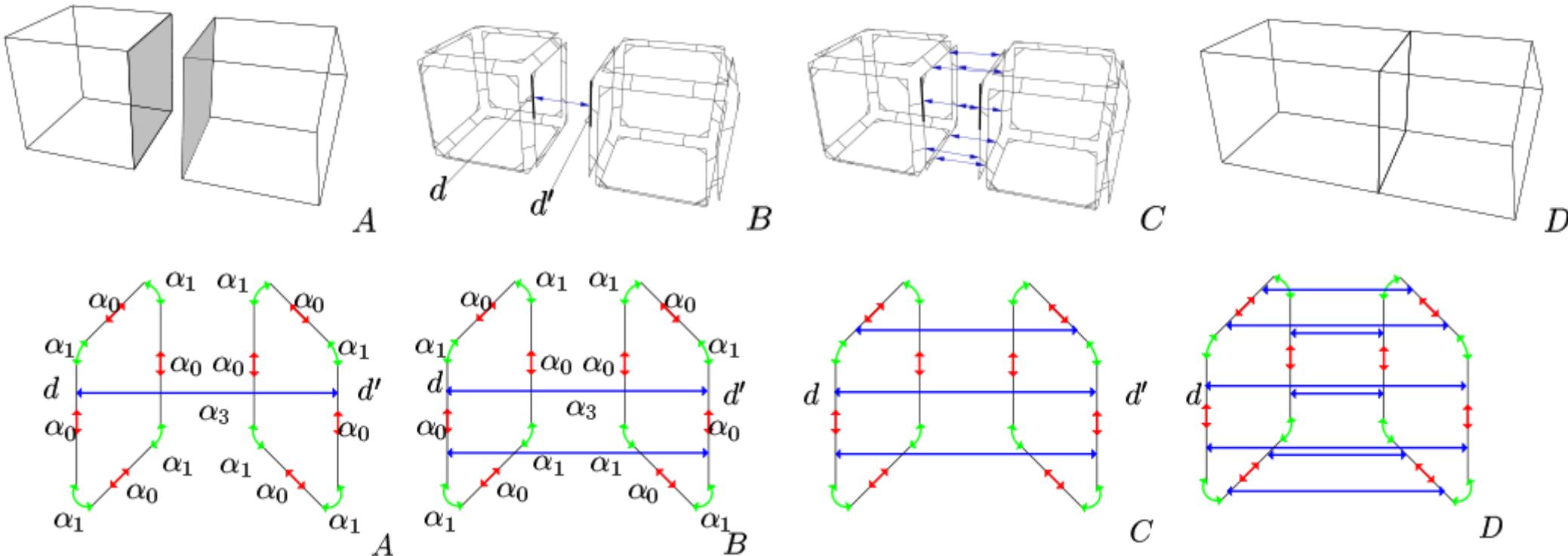
## 3-G-Cartes : Définition

- Du fait que  $\alpha_0 \circ \alpha_2$ ,  $\alpha_0 \circ \alpha_3$  et  $\alpha_1 \circ \alpha_3$  soient des involutions, cela oblige à coudre les volumes élémentaires pour tous les brins.
- Exemple de 3-G-carte (les liaisons  $\alpha_2$  ne sont pas représentées)



# Opération de Couture de face

- On détermine les brins de l'orbite qu'on lie par une relation  $\alpha_3$ .



Du fait que  $\alpha_0 \circ \alpha_2$ ,  $\alpha_0 \circ \alpha_3$  et  $\alpha_1 \circ \alpha_3$  soient des involutions, on doit coudre par  $\alpha_3$  tous les brins de l'orbite  $\langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle$  de  $d$  et  $d'$ .

**Couture par  $\alpha_i$  des brins sur les orbites  $\langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle$  de  $d$  et  $d'$ .**

## Propriétés des 3G-cartes

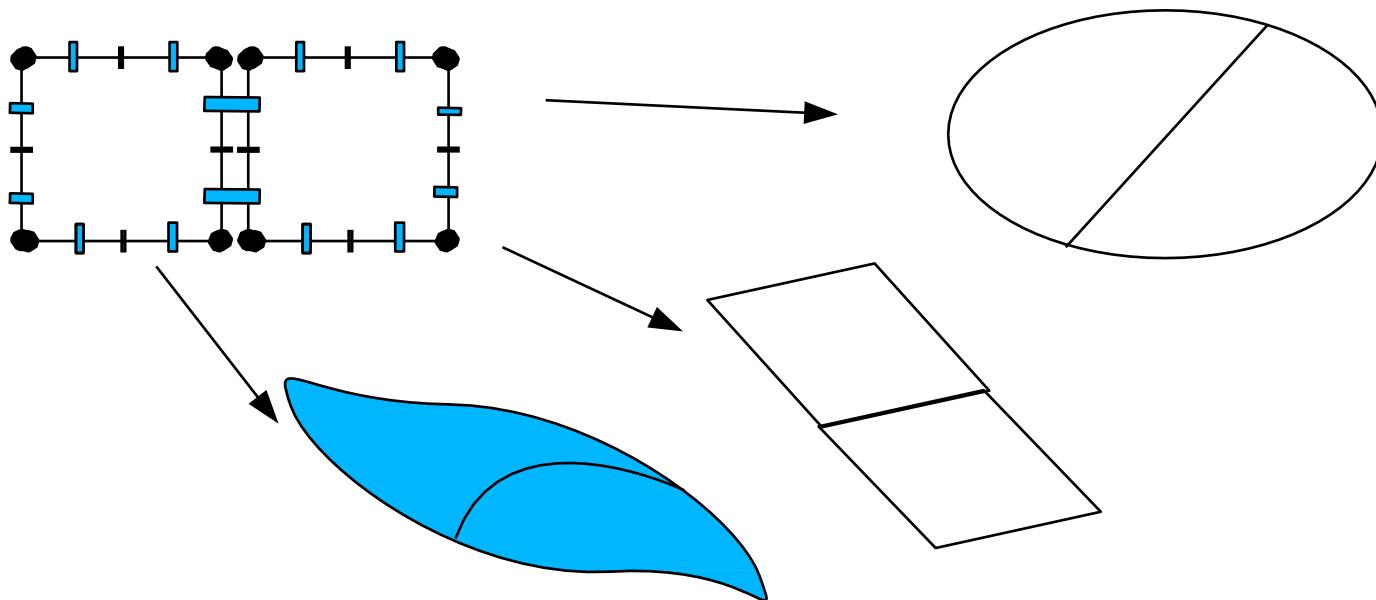
- On retrouve toutes les propriétés énoncées pour les 2-G-cartes, notamment les notions de cellules et de bords.
- Une  $i$ -cellule incidente à un brin  $b$  est définie par l'ensemble des brins incidents à l'orbite composée de toutes les involutions sauf  $\alpha_i$ .
- Le bord d'une 3-G-carte  $G(B, \alpha_0, \alpha_1, \alpha_2, \alpha_3)$  est définie par la 2-G-carte  $(B', \alpha'_0, \alpha'_1, \alpha'_2)$  où
  - $B'$  est un sous-ensemble de  $B$  réduit aux brins du bord (tels que  $\alpha_3(b) = b$ )
  - $\alpha'_0$  et  $\alpha'_1$  sont les restrictions de  $\alpha_0$  et  $\alpha_1$  aux brins du bord ;
  - $\alpha'_2$  est la restriction de  $\alpha_2\alpha_3$  aux brins du bord ;

## Cartes généralisées de dimension n ou n-G-cartes

- Une n-G-carte est un n-uplet  $(B, \alpha_0, \alpha_1, \dots, \alpha_n)$ 
  - $B$  est un ensemble fini de brins ;
  - Pour  $i$ ,  $0 \leq i < n-1$ ,  $\alpha_i$  est une involution sans point fixe ;
  - $\alpha_n$  est une involution avec ou sans point fixe ;
  - Pour  $i, j$ ,  $0 \leq i < i+2 \leq j \leq n$ ,  $\alpha_i \alpha_j$  est une involution.

# Plongement

- Plusieurs plongements géométriques possibles pour une même carte (topologie des objets)



- Plongement simple : 1 position géométrique par (groupe de) brin(s) associé à un sommet
- Gros avantage pour les algorithmes : traiter séparément topologie et géométrie

## Implémentation et opérations

- Structure de donnée très simple pour stocker la topologie d'une 3-G-Carte
  - Plongement simple à part : 1 point par brin ou par orbite de sommet

Class Brin

    Brin\* alpha[4]

    Point\* P

    bool marked

Class Point

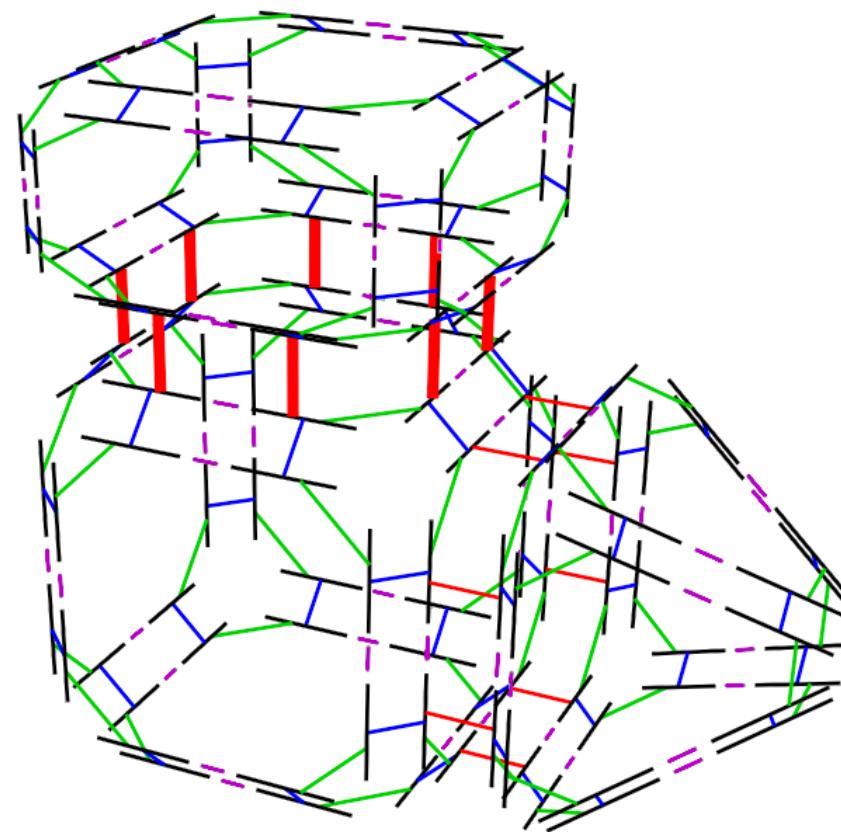
    x, y, z : double

- Manipulation par des opérations topologiques simples garantissant la cohérence
  - Parcours
  - Collage de 2 volumes par couture de faces
  - Éclatement de volume, de face, d'arête
  - Fusion de deux volumes
  - Dualisation
  - Chanfreinage de sommets ou d'arêtes
  - ...

# Parcours

Outil de base pour retrouver les orbites (et donc les cellules)

Exemple : parcours d'une face en 3D ?



# Parcours

Outil de base pour retrouver les orbites (et donc les cellules)

Exemple : parcours d'une face en 3D

**Algorithme 1** : Parcours de la face incidente à  $b$  (Version naïve)

$F :: \text{File}$

$\text{push}(F, b)$

Tant que  $F$  non-vide:

|     $b' \leftarrow \text{pop}(F)$

|    Si  $b'$  non-marqué:

|    |    (... traiter  $b'$  ...)

|    |    marquer  $b'$

|    |    Pour  $a$  dans  $[0, 1, 3]$ :

|    |    |     $\text{push}(F, \text{alpha}[a](b'))$

Démarquer les brins marqués

## Parcours

Outil de base pour retrouver les orbites (et donc les cellules)

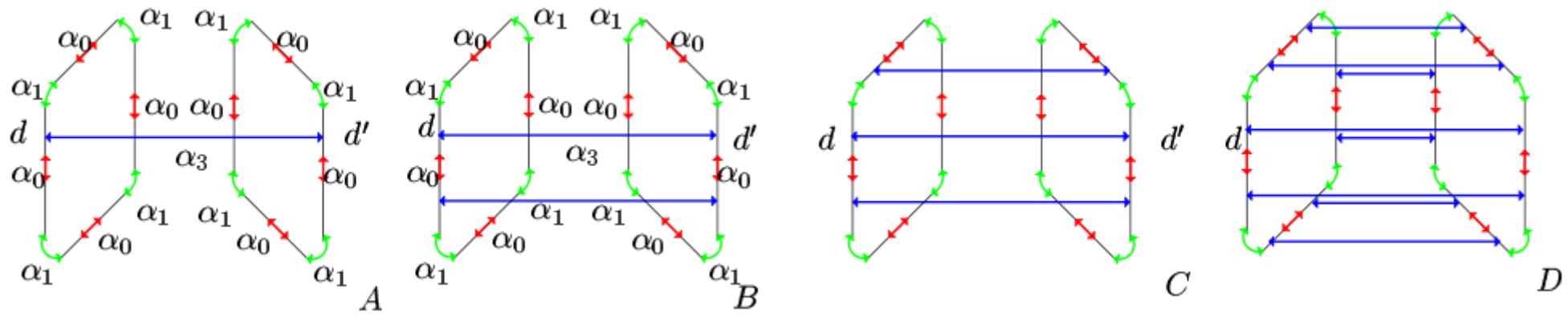
Exemple : parcours d'une face en 3D en utilisant le fait que  $\alpha_1 \circ \alpha_3$  et  $\alpha_1 \circ \alpha_3$  sont des involutions

**Algorithme 2** : Parcours de la face incidente à  $b$  (Version optimisée et ordonnée)

```
b' <- b
a_list = [3,0,3,1]
a_index = 0
Faire:
| (... traiter b' ...)
| a_index <- a_index+1 % 4
| a_suiv <- a_list[a_index]
| b' <- alpha[a_suiv](b')
Tant que b' != b
```

# Couture de faces

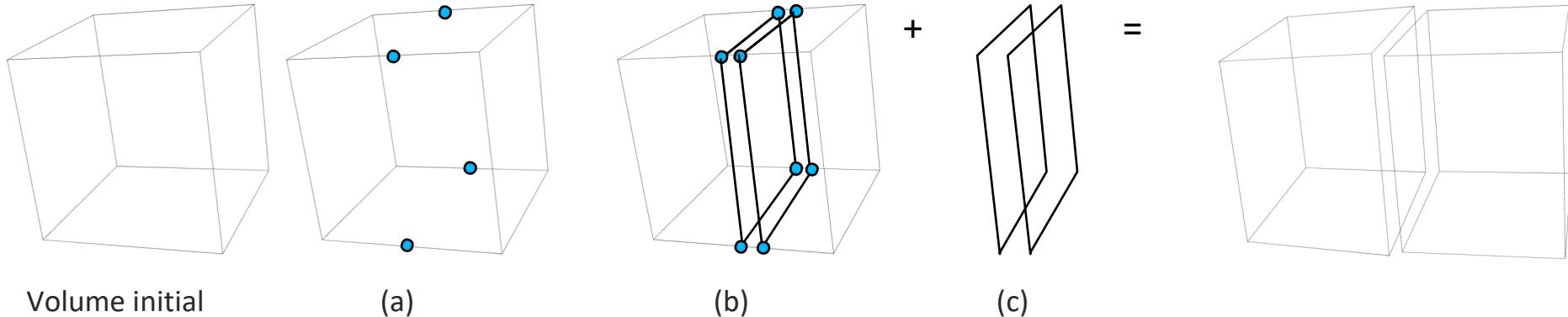
- Doit vérifier que les faces à coudre sont de même degré



- Dans le cas contraire : éclatement d'arête pour les rendre isomorphes

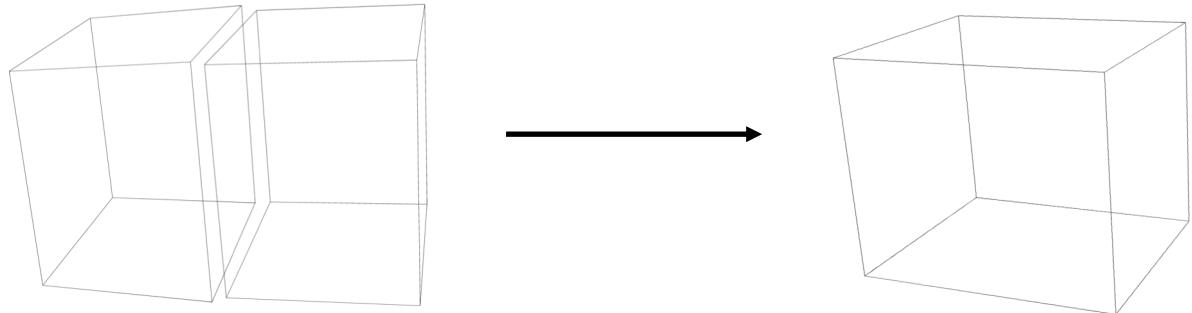
# Eclatement de volume

- L'opération d'éclatement de volume peut se décomposer en plusieurs opérations simples :
  - Éclatement d'arêtes (a)
  - Éclatement de faces (b)
  - Ajout de 2 faces (c)

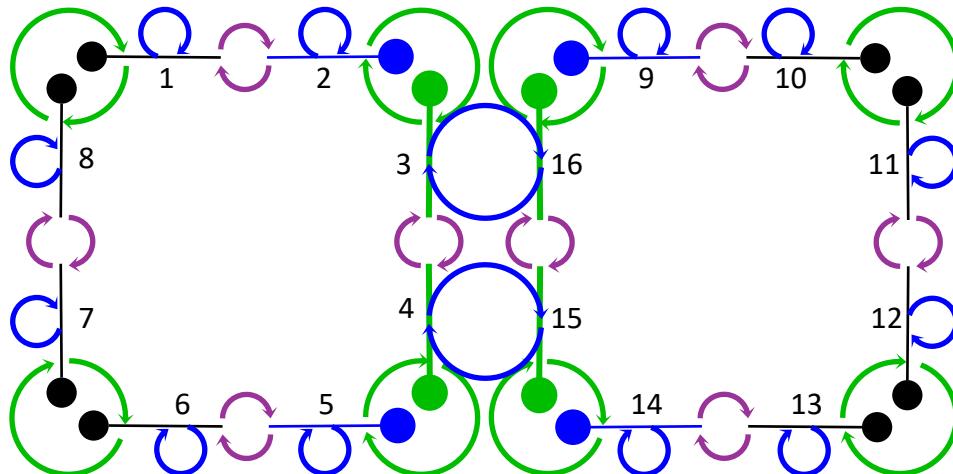


# Fusion de cellules

- Opération inverse :



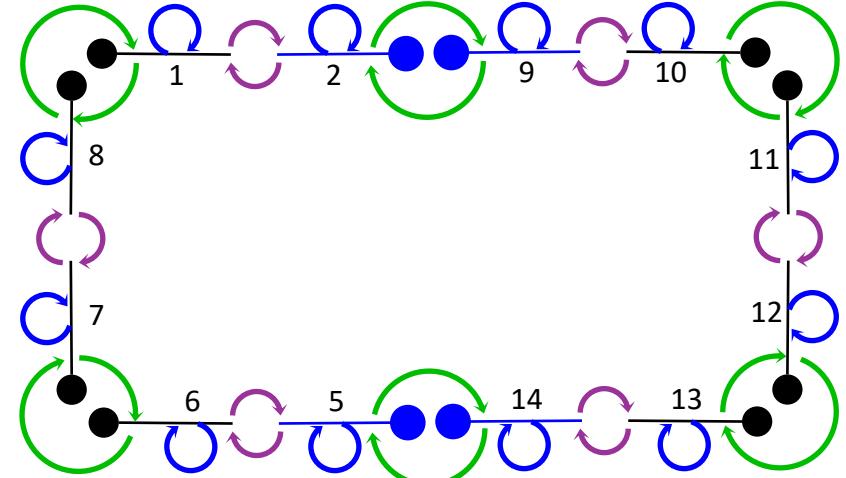
Bien déterminer les brins à supprimer et à recoller : exemple sur un 2-G-Carte



Suppression de l'arête incidente à b:

- Supprimer  $C = \langle \alpha_0, \alpha_2 \rangle(b)$
- Recoller  $B^S = \alpha_1(C) - C$

$$\forall b' \in B^S, \alpha_1(b') \leftarrow \alpha_1(\alpha_2(\alpha_1(b')))$$



# Suppression de cellules

- De façon générale : suppression d'une i-cellule ( $i < n$ ) incidente au brin b

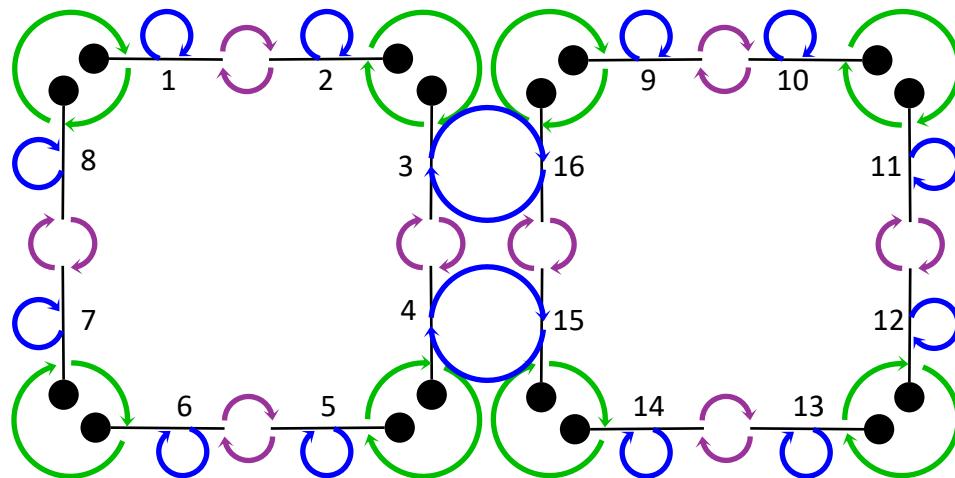
- Orbite à supprimer :  $C = \langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \alpha_n \rangle(b)$
- Vérifier la cohérence résultante :

$$\forall b' \in C, \alpha_{i+1}(\alpha_{i+2}(b')) = \alpha_{i+2}(\alpha_{i+1}(b'))$$

- Si valide, supprimer C et recoller  $B^S = \alpha_i(C) - C$  tel que:

$$\forall b' \in B^S, \alpha_i(b') \leftarrow \alpha_i(\alpha_{i+1}(\alpha_i(b')))$$

**Exercice** : suppression du sommet incident au brin 10



# Suppression de cellules

- De façon générale : suppression d'une i-cellule ( $i < n$ ) incidente au brin b

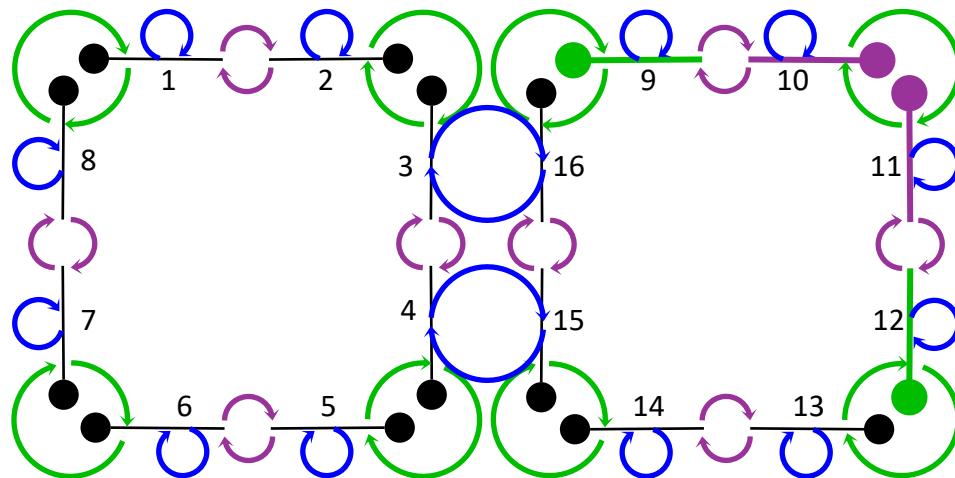
- Orbite à supprimer :  $C = \langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \alpha_n \rangle(b)$
- Vérifier la cohérence résultante :

$$\forall b' \in C, \alpha_{i+1}(\alpha_{i+2}(b')) = \alpha_{i+2}(\alpha_{i+1}(b'))$$

- Si valide, supprimer C et recoller  $B^S = \alpha_i(C) - C$  tel que:

$$\forall b' \in B^S, \alpha_i(b') \leftarrow \alpha_i(\alpha_{i+1}(\alpha_i(b')))$$

**Exercice** : suppression du sommet incident au brin 10



## Suppression de cellules

- De façon générale : suppression d'une i-cellule ( $i < n$ ) incidente au brin b

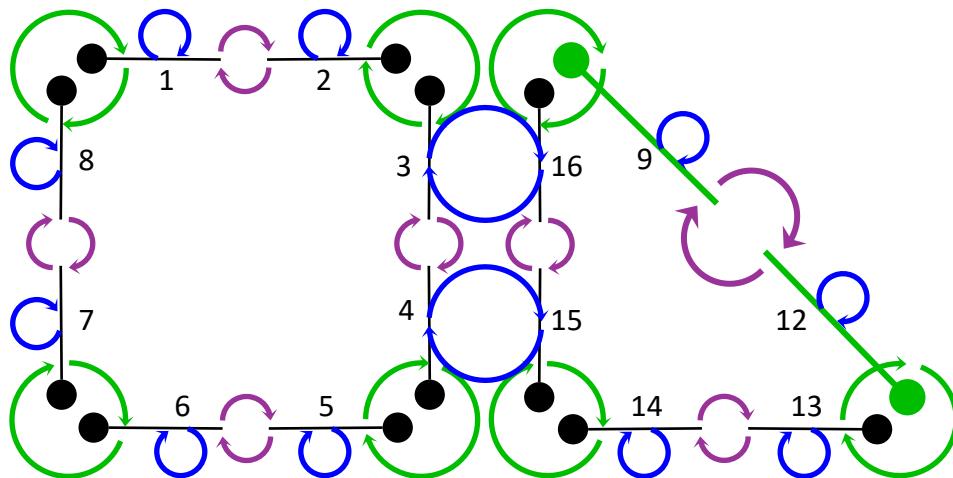
- Orbite à supprimer :  $C = \langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle(b)$
  - Vérifier la cohérence résultante

$$\forall b' \in C, \alpha_{i+1}(\alpha_{i+2}(b')) = \alpha_{i+2}(\alpha_{i+1}(b'))$$

- Si valide, supprimer C et recoller  $B^S = \alpha_i(C) - C$  tel que:

$$\forall b' \in B^S, \alpha_i(b') \leftarrow \alpha_i(\alpha_{i+1}(\alpha_i(b')))$$

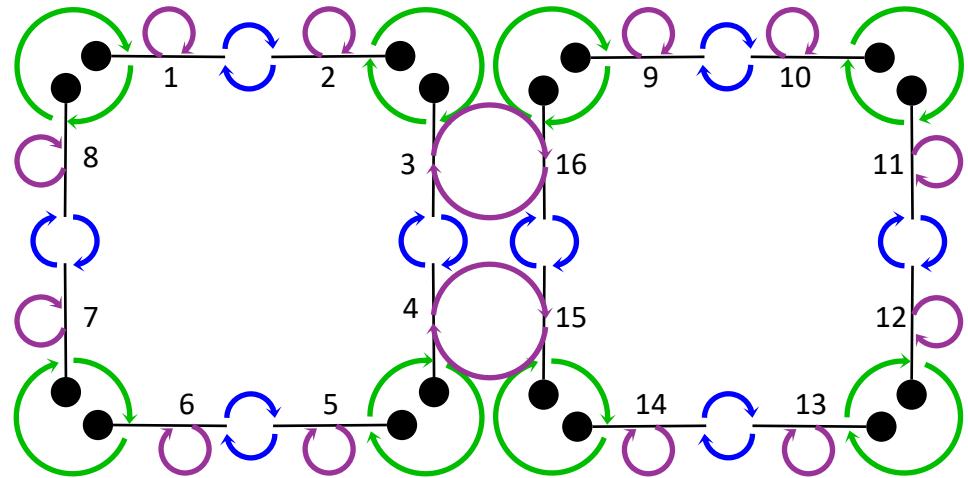
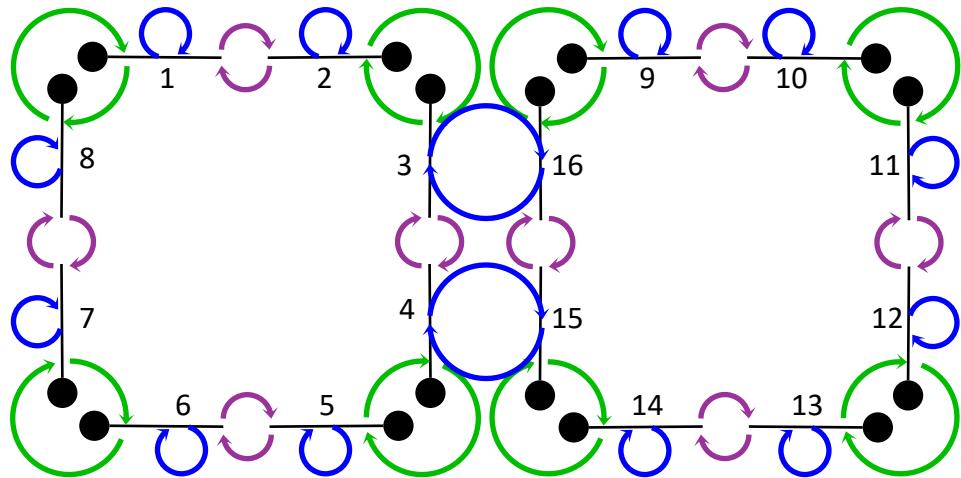
## **Exercice : suppression du sommet incident au brin 10**



# Dualisation

- Préserver tous les brins et renverser les relations  $\alpha_i$

$$\forall i \in \{0..n\}, \forall b \in B, \alpha_i b \leftarrow \alpha_{n-i}(b)$$

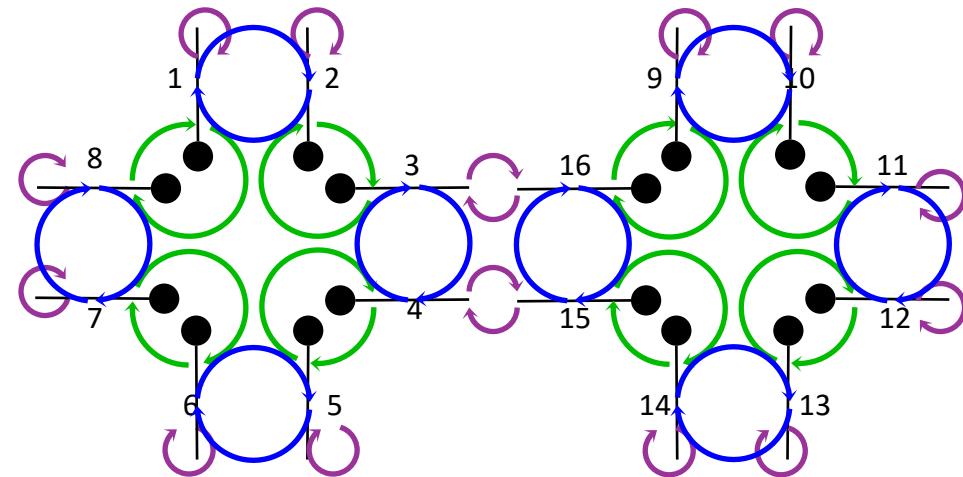
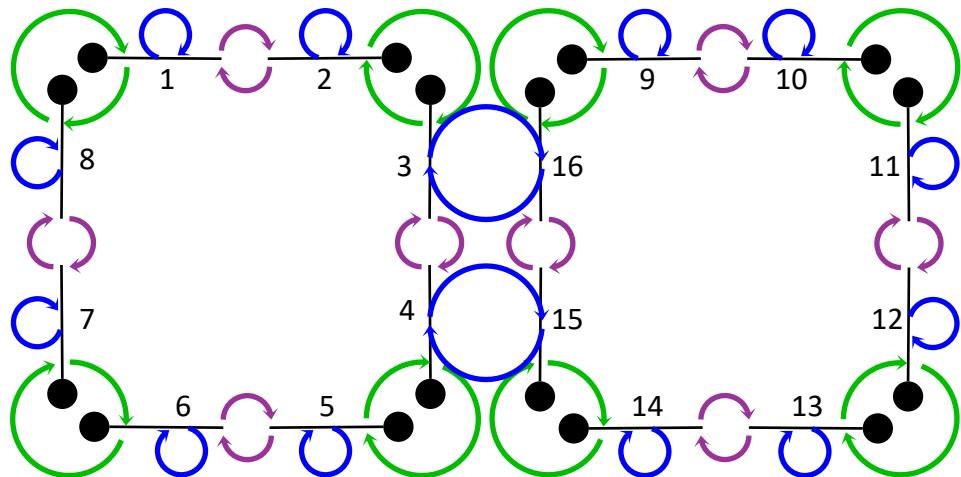


- Changement des orbites décrivant les sommets  $\langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle$

# Dualisation

- Préserver tous les brins et renverser les relations  $\alpha_i$

$$\forall i \in \{0..n\}, \forall b \in B, \alpha_i b \leftarrow \alpha_{n-i}(b)$$

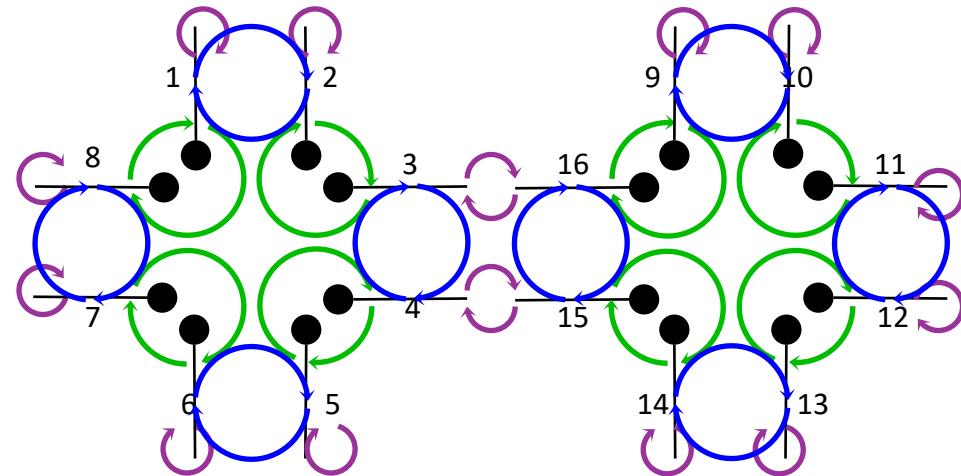
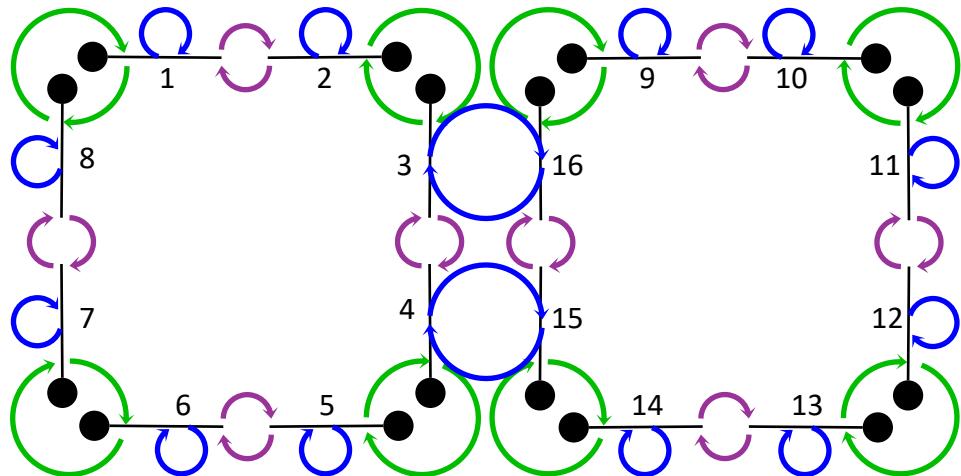


- Changement des orbites décrivant les sommets  $\langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle$
- Nécessairement, changement du plongement géométrique

# Dualisation

- Préserver tous les brins et renverser les relations  $\alpha_i$

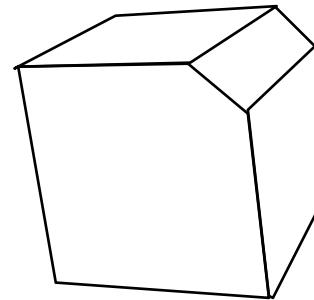
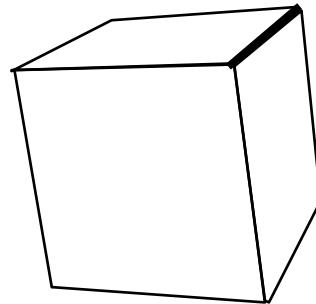
$$\forall i \in \{0..n\}, \forall b \in B, \alpha_i b \leftarrow \alpha_{n-i}(b)$$



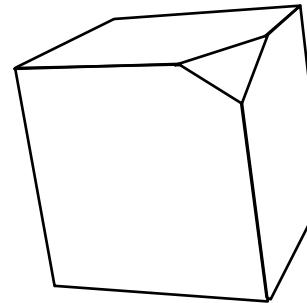
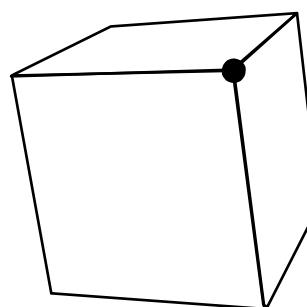
- Changement des orbites décrivant les sommets  $\langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle$
- Nécessairement, changement du plongement géométrique
- Valide uniquement si  $\alpha_n$  sans point fixe...

# Chanfreinage

- Les opérations de chanfreinage consistent en l'ajout de faces :
  - En remplacement d'une arête (a) ;
  - En remplacement d'un sommet (b).



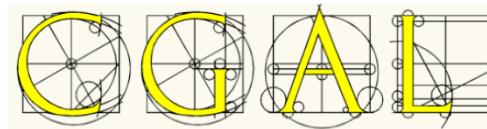
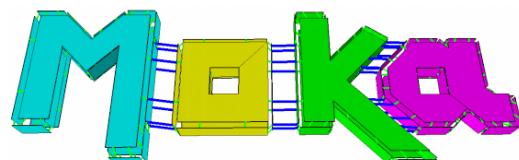
(a) Chanfreinage d'une arête.



(b) Chanfreinage d'un sommet.

# Conclusion

- **Cartes généralisées :**
  - Un des modèles B-Rep les plus intuitifs et concis en termes de définition
  - Générique, pouvant représenter des objets en n-dimensions
  - Implémentation relativement simple, quoique gourmande en mémoire
- Manipulation de subdivisions volumiques complexes:
  - Avec une grande variété et flexibilité d'opérations sur la topologie (extension, fusion, division, ...)
  - En ayant un traitement indépendant de la géométrie
- Implémentations dans plusieurs librairies open-source



# Références

## Références sur les cartes généralisées :

- P. Lienhardt. *Topological models for boundary representation: a comparison with n-dimensional generalized maps.* Computer-Aided Design, Volume 23, Issue 1 Jan./Feb. 1991.
- P. Lienhardt. *N-dimensional generalized topological maps. and cellular quasi-manifolds.* Journal of Computational Geometry and Applications , 1994.
- P. Lienhardt. *Aspects in Topology-Based Geometric Modeling.* Proceedings of the 7th International Workshop on Discrete Geometry for Computer Imagery, 1997.
- J.F. Dufourd. *An obj3 functional specifications for the boundary representation.* ACM Symposium on Solid Modeling Foundation and CAD/CAM Applications, 1991
- Y. Bertrand, J.F. Dufourd, J. Françon, and P. Lienhardt. *Algebraic specifications and developments in geometric modeling.* Avril 1993. TAPSOFT'93, Orsay.

# Références

## Références sur les "autres" structures combinatoires

- [AFF85] S. Ansaldi, L. de Floriani, B. Falcidieno, *Geometric Modeling of Solid Objects by Using a Face Adjacency Graph Representation*, Computer Graphics, 19, 3, 131-139, 1985.
- [Baum75] B. Baumgart, *A polyhedron Representation for Computer Vision*, AFPIIS Nat. Conf. Proc., 44, 589-596, 1975.
- [Bris89] E. Brisson, *Representing Geometric Structures in D Dimensions :Topology and Order*. Actes de 5<sup>th</sup> A.C.M. Symposium on Computational Geometry, Saarbrücken, 218-227, juin 1989.
- [CR91] G. Crocker, W. Reinke. *An editable Non-Manifold Boundary Representation*. Computer Graphics and Applications, 11, 2, 1991
- [DL87] D. Dobkin, M. Laszlo. *Primitives for the Manipulation of Three-Dimensional Subdivisions*. In Third ACM Symposium on Computational Geometry. Juin 1987

# Références

## Références sur les "autres" structures combinatoires (suite)

- [GS85] Guibas, L. and Stolfi, J., *Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams*, ACT Transactions On Graphics, 4(2), April, 1985.
- [LL90] Y. Luo, G. Lukacs, *A boundary representation for form features and non-manifold solid objects*. In Proceedings of Symposium on Solid Modeling and Applications'1991.
- [Sobh89] C. Sobhanpanah, *Extension of a boundary representation technique for the description of N dimensional polytopes*. Computers & Graphics 13(1): 17-23, 1989
- [Tutt84] W. Tuttle, *Graph theory*, Encyclopedia of Mathematics and its Applications, Addison Wesley, 1984
- [Weil85] K. Weiler, *Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments*, IEEE Computer Graphics and Applications, vol. 5, no. 1, pp. 21-40, Jan., 1985
- [Weil86] K. Weiler, *The radial edge structure: a topological representation for non-manifold geometric boundary modeling*, IFIP WG5.2 Working Conference on Geometric Modeling for CAD Applications, Rensselaerville, NY, 12-14, May 1986