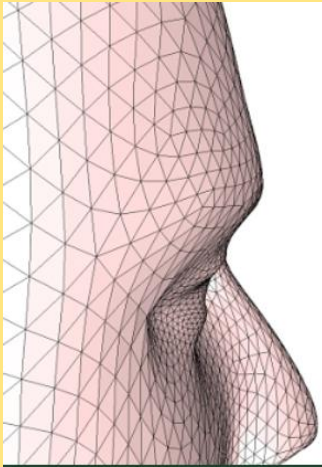


Orientations de cartes,  
dessins et compression

Gilles Schaeffer

LIX, CNRS/École Polytechnique  
ACI MD GéoComp

# Maillages, triangulations et cartes



Combinatoire

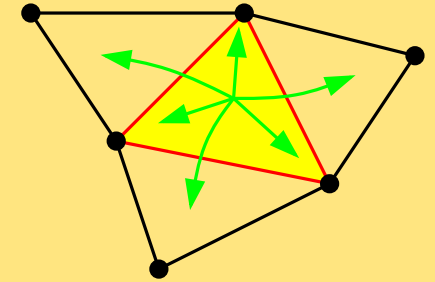
Géométrie

## Maillage triangulaire

```
struct triangle{  
    triangle *t1, *t2, *t3;  
    vertex *v1, *v2, *v3;  
}
```

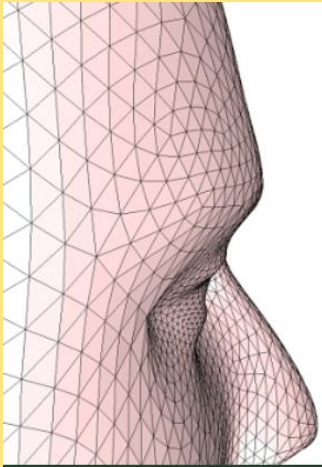
```
struct vertex{  
    triangle * root;  
    int label;  
}
```

```
struct point{  
    float x;  
    float y;  
}  
point vertex_geometry[max_label];
```



(Tous les maillages présents sur ces transparents sont pris sur les pages du projet Geometrica.)

# Maillages, triangulations et cartes



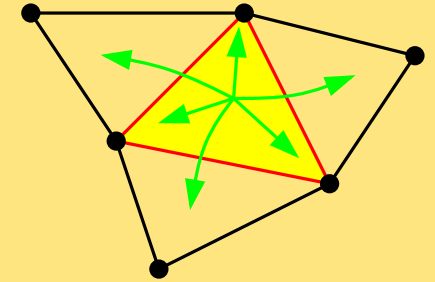
Combinatoire

Géométrie

Maillage triangulaire

```
struct triangle{  
    triangle *t1, *t2, *t3;  
    vertex *v1, *v2, *v3;  
}
```

```
struct vertex{  
    triangle * root;  
    int label;  
}
```

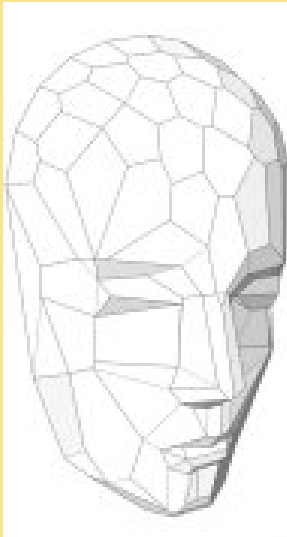


triangulation combinatoire

```
struct point{  
    float x;  
    float y;  
}  
point vertex_geometry[max_label];
```

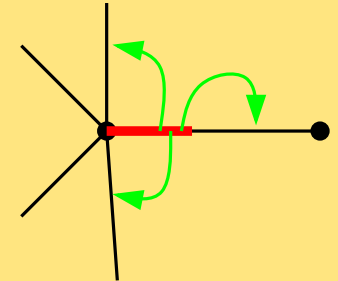
(Tous les maillages présents sur ces transparents sont pris sur les pages du projet Geometrica.)

# Maillages, triangulations et cartes



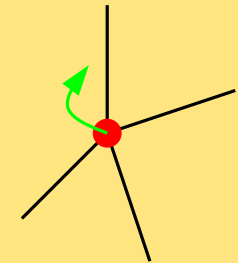
Combinatoire

```
struct edge{  
    edge * oppo;  
    edge * next;  
    edge * prev;  
    vertex * origin;  
}
```



Géométrie

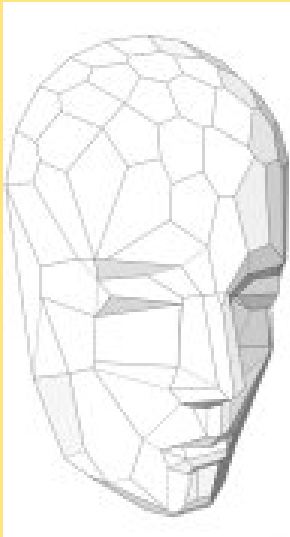
```
struct vertex{  
    edge * root;  
    int label;  
}
```



Maillage polyedral

```
struct point{  
    float x;  
    float y;  
}  
point vertex_geometry[max_label];
```

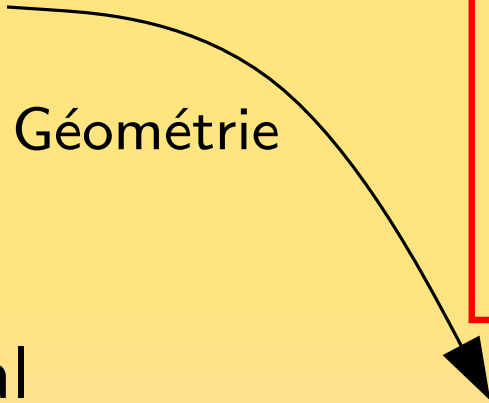
# Maillages, triangulations et cartes



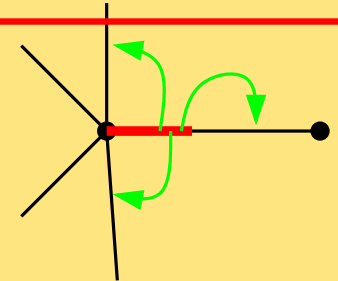
Combinatoire



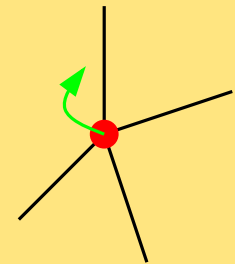
Géométrie



```
struct edge{  
    edge * oppo;  
    edge * next;  
    edge * prev;  
    vertex * origin;  
}
```



```
struct vertex{  
    edge * root;  
    int label;  
}
```



Carte 3-connexe

Maillage polyedral

```
struct point{  
    float x;  
    float y;  
}  
point vertex_geometry[max_label];
```

# Triangulations et cartes

## La combinatoire des cartes

- problèmes de coloration, polynôme de Tutte, énumération, lien avec la physique statistique, etc.

# Triangulations et cartes

## La combinatoire des cartes

- problèmes de coloration, polynôme de Tutte, énumération, lien avec la physique statistique, etc.

## Dessin de cartes

- redessiner une carte plane en plaçant les sommets sur une grille aussi petite que possible, avec arêtes de direction arbitraire ou avec arêtes orthogonales, etc.

# Triangulations et cartes

## La combinatoire des cartes

- problèmes de coloration, polynôme de Tutte, énumération, lien avec la physique statistique, etc.

## Dessin de cartes

- redessiner une carte plane en plaçant les sommets sur une grille aussi petite que possible, avec arêtes de direction arbitraire ou avec arêtes orthogonales, etc.

## Compression, codage, représentation compacte

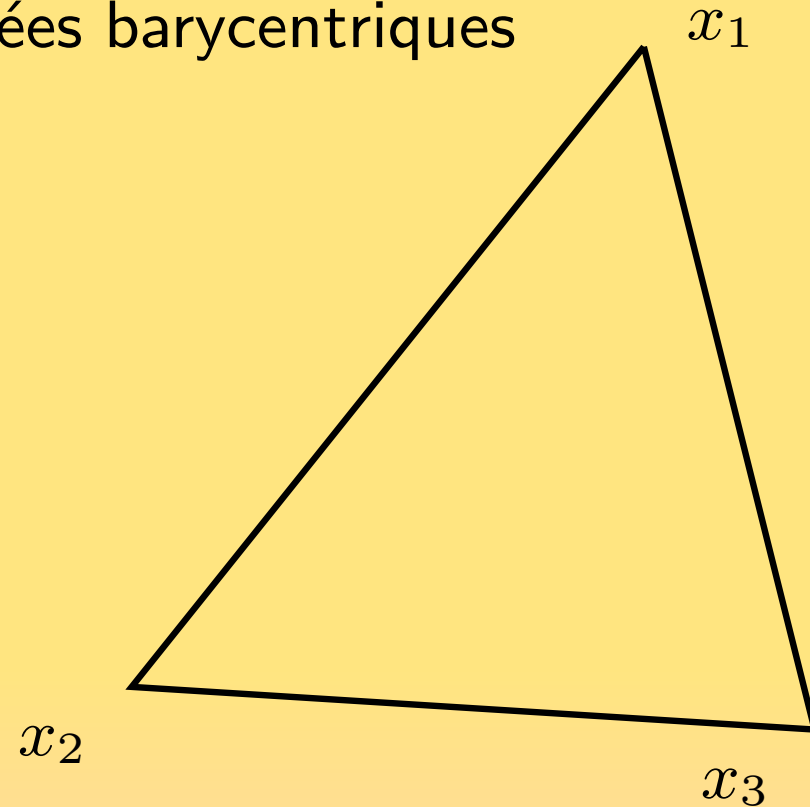
- Minimiser l'espace mémoire occupé par une carte.
- Traiter des requêtes en temps constant.  
Exemple de requête : deux sommets sont-ils voisins ?



Dessin de triangulations et forêts de Schnyder

# L'algorithme de dessin de triangulations de Schnyder

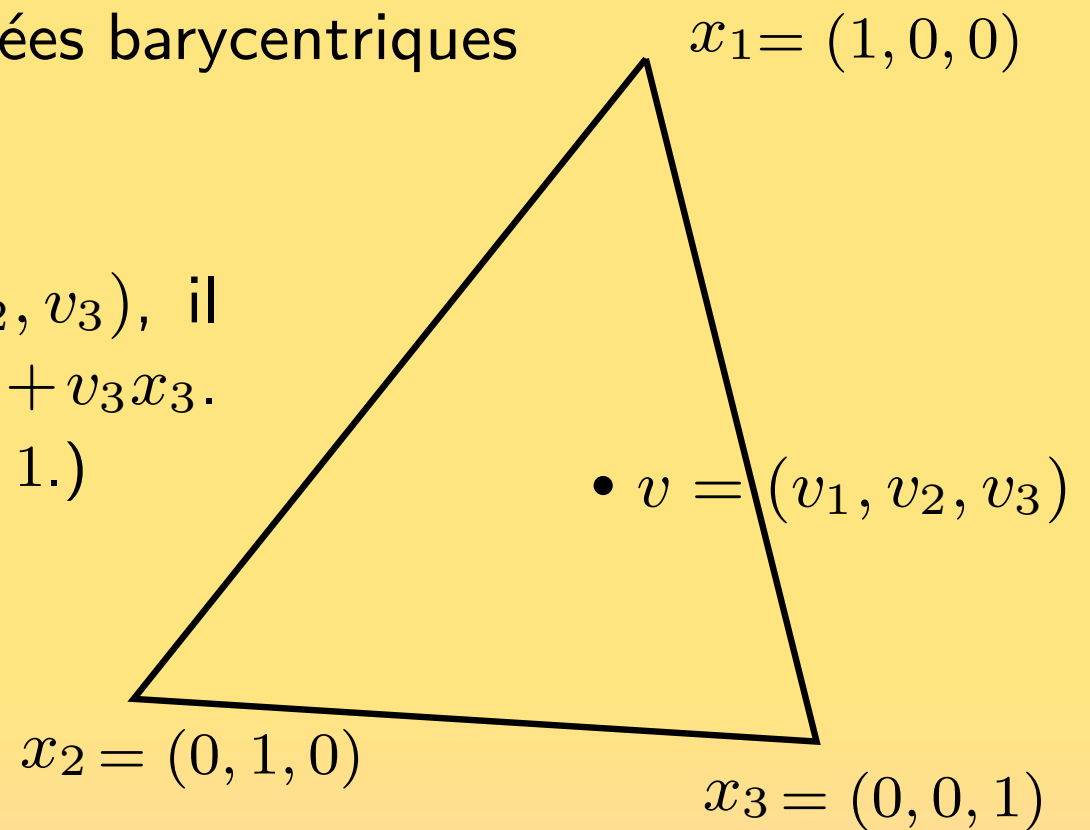
On place les 3 points  $x_1$ ,  $x_2$  et  $x_3$  arbitrairement et on veut calculer des coordonnées barycentriques pour les points intérieurs.



# L'algorithme de dessin de triangulations de Schnyder

On place les 3 points  $x_1$ ,  $x_2$  et  $x_3$  arbitrairement et on veut calculer des coordonnées barycentriques pour les points intérieurs.

Si  $v$  a pour coordonnées  $(v_1, v_2, v_3)$ , il est placé en position  $v_1x_1 + v_2x_2 + v_3x_3$ .  
(On suppose  $v_i \in (0, 1)$  et  $\sum v_i = 1$ .)

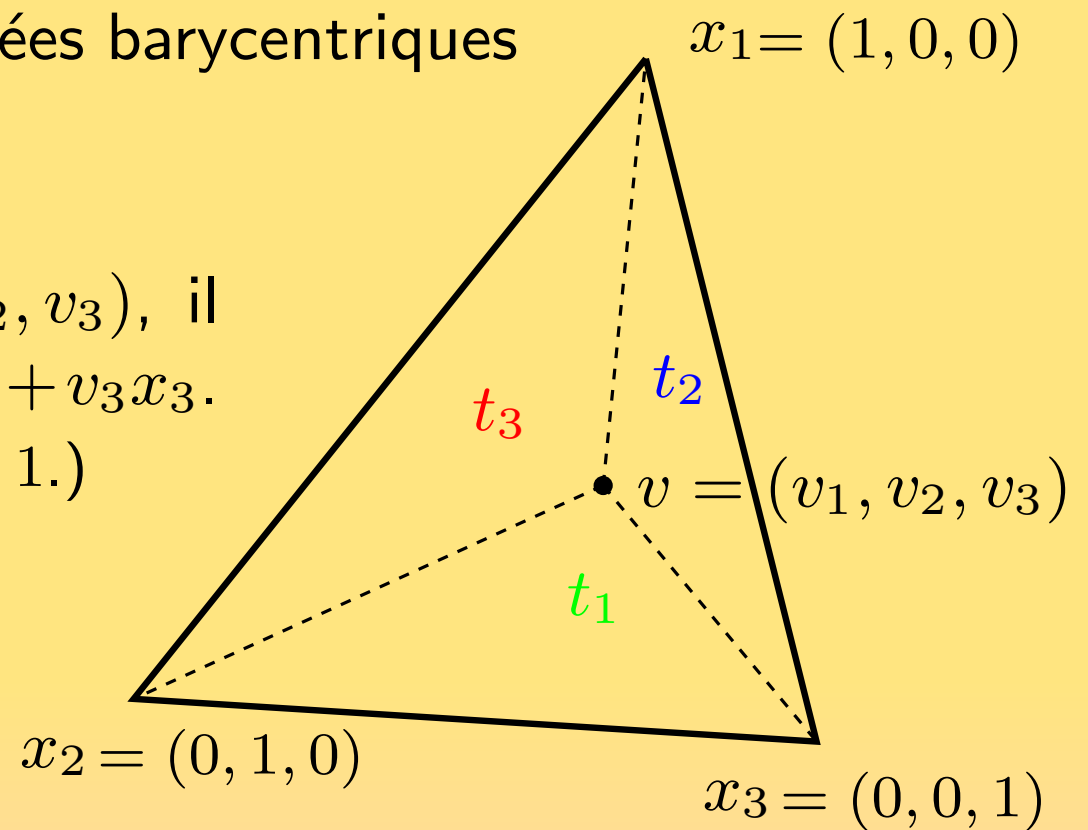


# L'algorithme de dessin de triangulations de Schnyder

On place les 3 points  $x_1$ ,  $x_2$  et  $x_3$  arbitrairement et on veut calculer des coordonnées barycentriques pour les points intérieurs.

Si  $v$  a pour coordonnées  $(v_1, v_2, v_3)$ , il est placé en position  $v_1x_1 + v_2x_2 + v_3x_3$ .  
(On suppose  $v_i \in (0, 1)$  et  $\sum v_i = 1$ .)

La coordonnée  $v_i$  correspond à l'aire du triangle  $t_i$ .

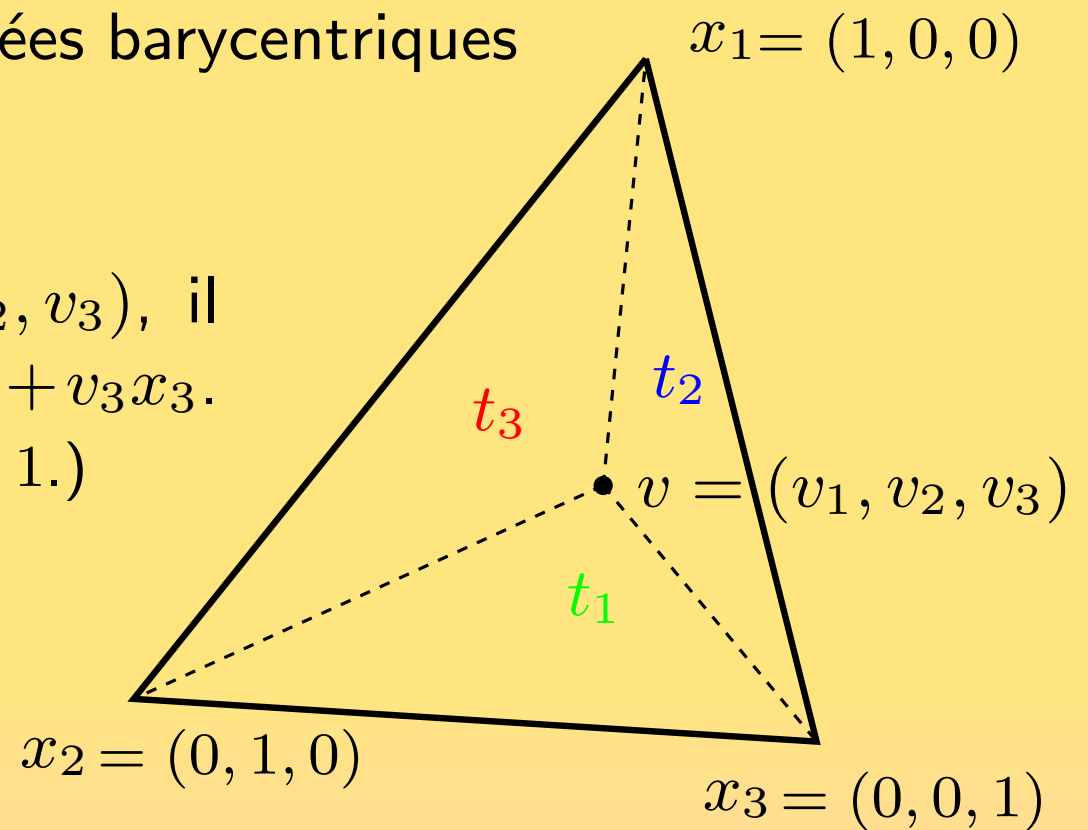


# L'algorithme de dessin de triangulations de Schnyder

On place les 3 points  $x_1$ ,  $x_2$  et  $x_3$  arbitrairement et on veut calculer des coordonnées barycentriques pour les points intérieurs.

Si  $v$  a pour coordonnées  $(v_1, v_2, v_3)$ , il est placé en position  $v_1x_1 + v_2x_2 + v_3x_3$ .  
(On suppose  $v_i \in (0, 1)$  et  $\sum v_i = 1$ .)

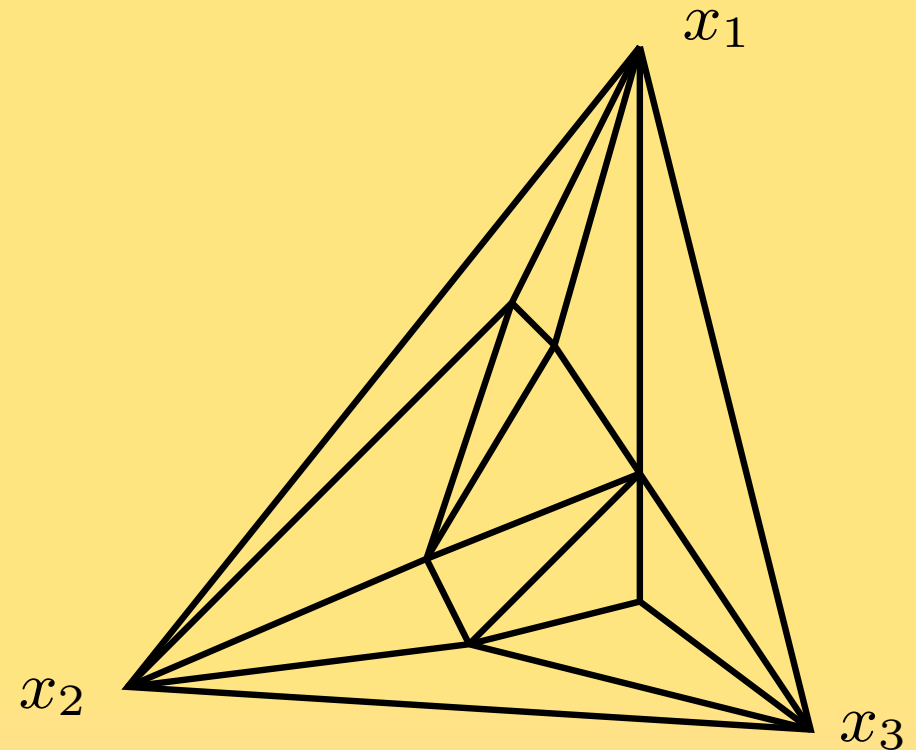
La coordonnée  $v_i$  correspond à l'aire du triangle  $t_i$ .



On veut un équivalent combinatoire de ces triangles...

# Triangulations et forêts de Schnyder

Soit  $T$  une triangulation de bord  $\{x_1, x_2, x_3\}$ .



$I = \{\text{sommets internes}\}$ .

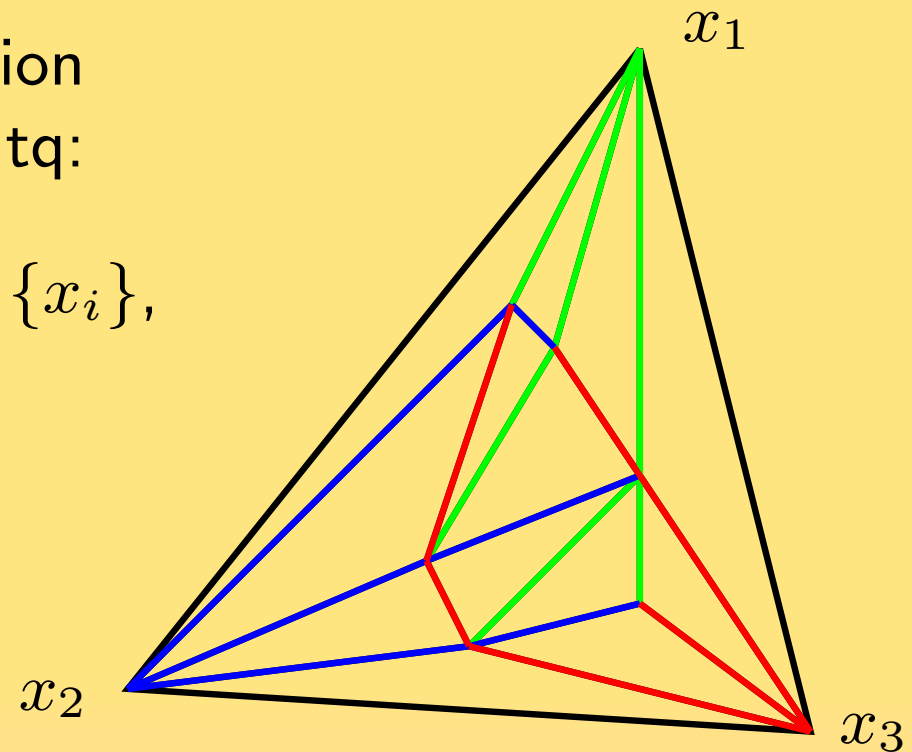
(ici  $|I| = 6$ )

# Triangulations et forêts de Schnyder

Soit  $T$  une triangulation de bord  $\{x_1, x_2, x_3\}$ .

Une forêt de Schnyder est une partition  $T_1, T_2, T_3$  des arêtes internes de  $T$  tq:

i)  $T_i$  est un arbre couvrant de  $I \cup \{x_i\}$ ,



$I = \{\text{sommets internes}\}$ .

(ici  $|I| = 6$ )

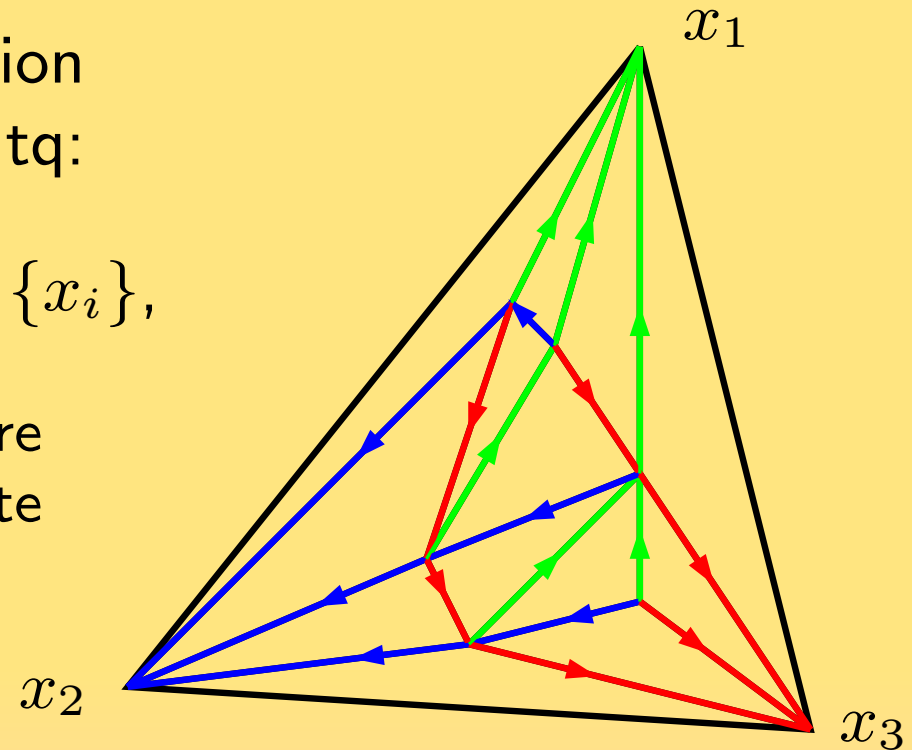
# Triangulations et forêts de Schnyder

Soit  $T$  une triangulation de bord  $\{x_1, x_2, x_3\}$ .

Une forêt de Schnyder est une partition  $T_1, T_2, T_3$  des arêtes internes de  $T$  tq:

i)  $T_i$  est un arbre couvrant de  $I \cup \{x_i\}$ ,

Si on oriente les arêtes de chaque arbre vers sa racine, chaque sommet a 1 arête sortante de chaque couleur.



$I = \{\text{sommets internes}\}$ .

(ici  $|I| = 6$ )



# Triangulations et forêts de Schnyder

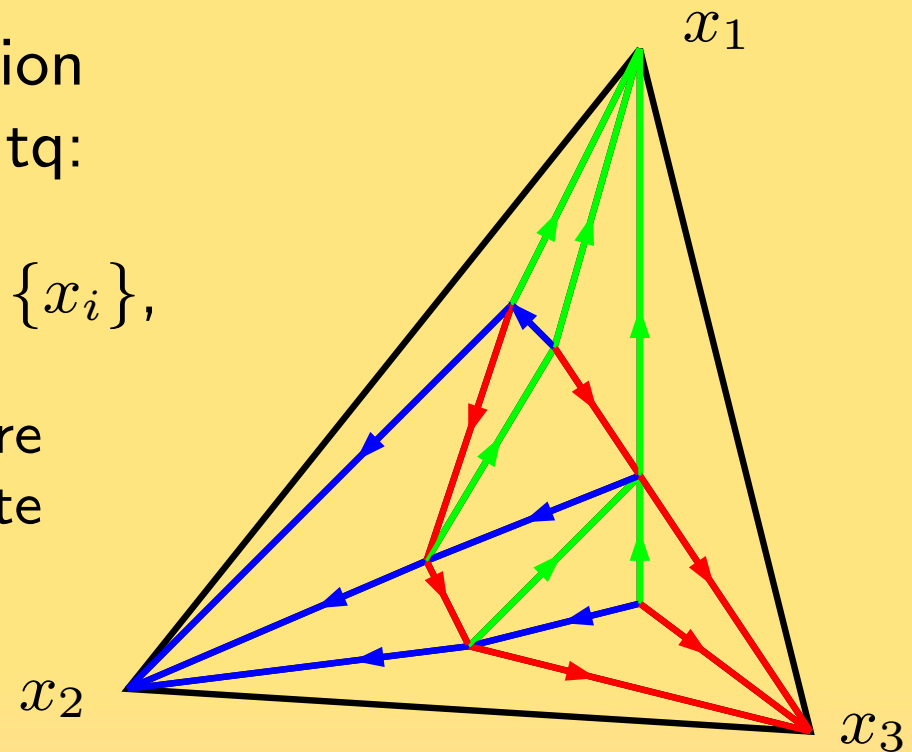
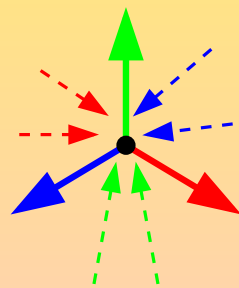
Soit  $T$  une triangulation de bord  $\{x_1, x_2, x_3\}$ .

Une forêt de Schnyder est une partition  $T_1, T_2, T_3$  des arêtes internes de  $T$  tq:

i)  $T_i$  est un arbre couvrant de  $I \cup \{x_i\}$ ,

Si on oriente les arêtes de chaque arbre vers sa racine, chaque sommet a 1 arête sortante de chaque couleur.

ii) Les couleurs doivent respecter la condition de Schnyder.



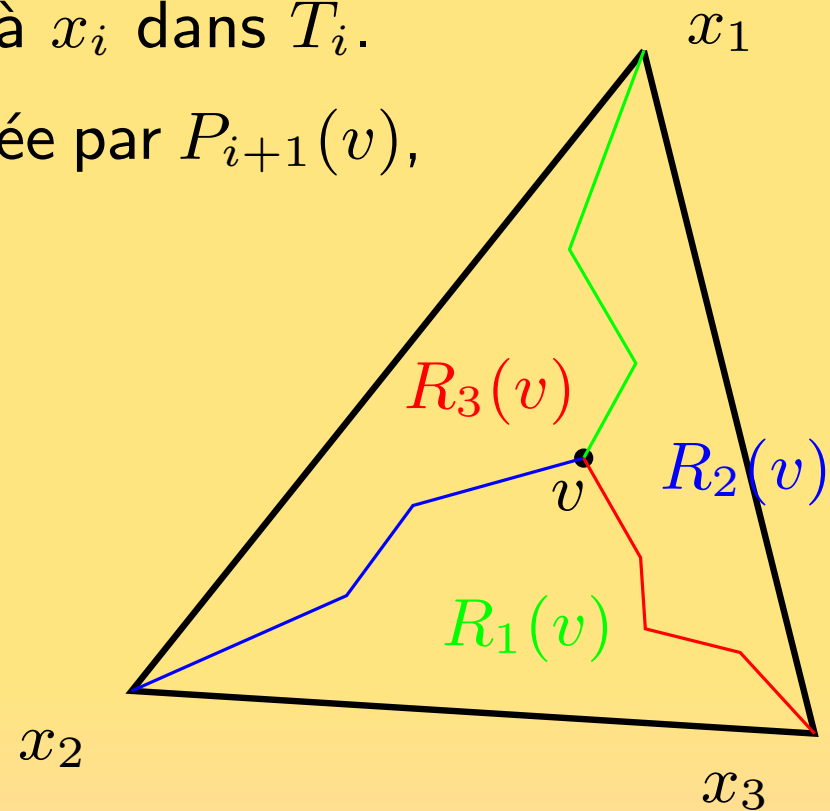
$I = \{\text{sommets internes}\}$ .

(ici  $|I| = 6$ )

# L'algorithme de dessin de triangulations de Schnyder

On utilise une forêt de Schnyder:

- Soit  $P_i(v)$  le chemin de  $v$  à  $x_i$  dans  $T_i$ .
- Soit  $R_i(v)$  la région délimitée par  $P_{i+1}(v)$ ,  $P_{i+2}(v)$  et  $(x_{i+1}, x_{i+2})$ .



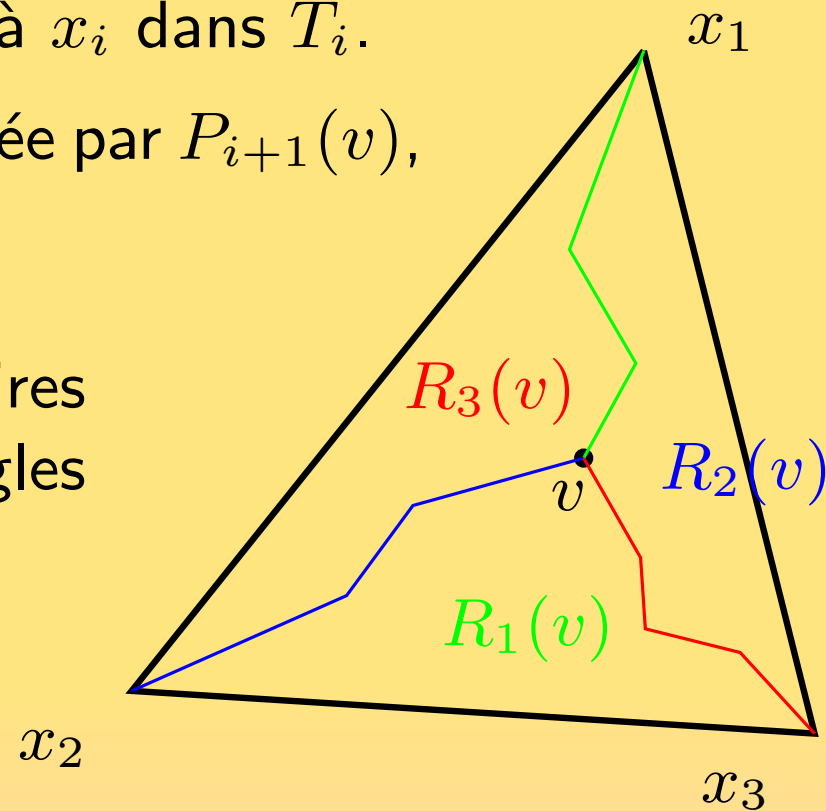
# L'algorithme de dessin de triangulations de Schnyder

On utilise une forêt de Schnyder:

- Soit  $P_i(v)$  le chemin de  $v$  à  $x_i$  dans  $T_i$ .
- Soit  $R_i(v)$  la région délimitée par  $P_{i+1}(v)$ ,  $P_{i+2}(v)$  et  $(x_{i+1}, x_{i+2})$ .

L'équivalent combinatoire des aires est donné par le nombre de triangles inclus dans chaque région:

$$v_i = \frac{|R_i(v)|}{|T|}$$



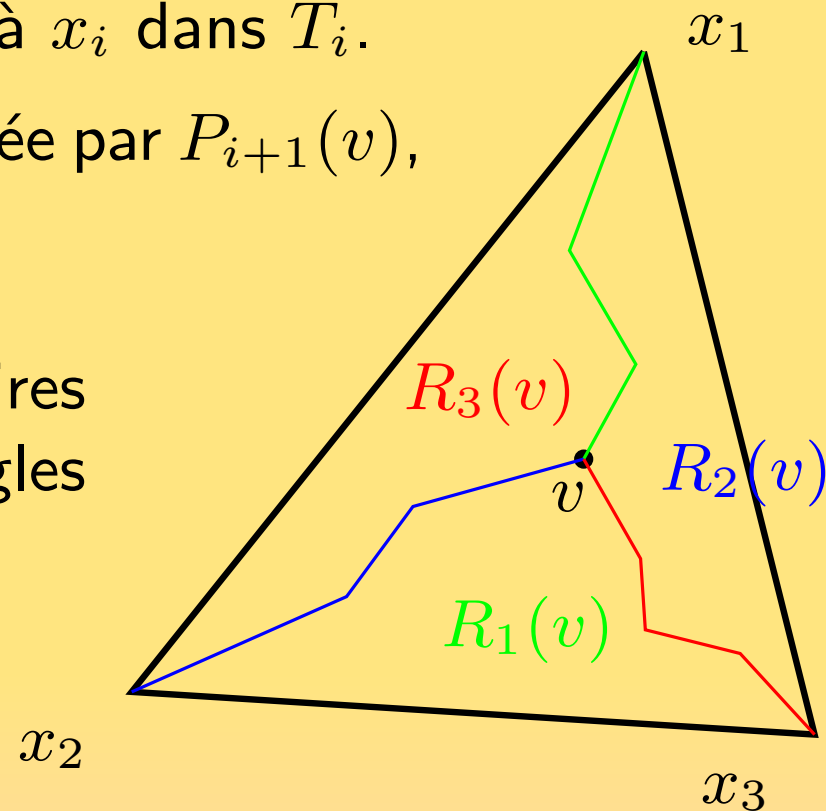
# L'algorithme de dessin de triangulations de Schnyder

On utilise une forêt de Schnyder:

- Soit  $P_i(v)$  le chemin de  $v$  à  $x_i$  dans  $T_i$ .
- Soit  $R_i(v)$  la région délimitée par  $P_{i+1}(v)$ ,  $P_{i+2}(v)$  et  $(x_{i+1}, x_{i+2})$ .

L'équivalent combinatoire des aires est donné par le nombre de triangles inclus dans chaque région:

$$v_i = \frac{|R_i(v)|}{|T|}$$



**Théorème. (Schnyder)** Le dessin de  $T$  avec arêtes rectilignes et chaque sommet  $v$  de coordonnées barycentriques  $(v_1, v_2, v_3)$  est planaire, quel que soit le placement de  $x_1, x_2, x_3$  (non alignés).

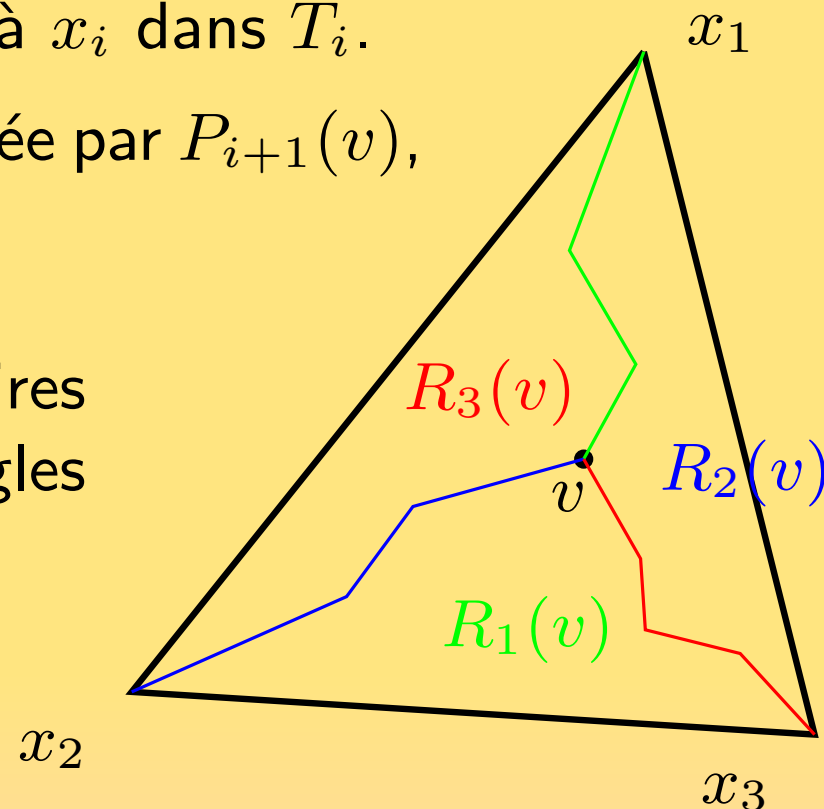
# L'algorithme de dessin de triangulations de Schnyder

On utilise une forêt de Schnyder:

- Soit  $P_i(v)$  le chemin de  $v$  à  $x_i$  dans  $T_i$ .
- Soit  $R_i(v)$  la région délimitée par  $P_{i+1}(v)$ ,  $P_{i+2}(v)$  et  $(x_{i+1}, x_{i+2})$ .

L'équivalent combinatoire des aires est donné par le nombre de triangles inclus dans chaque région:

$$v_i = \frac{|R_i(v)|}{|T|}$$



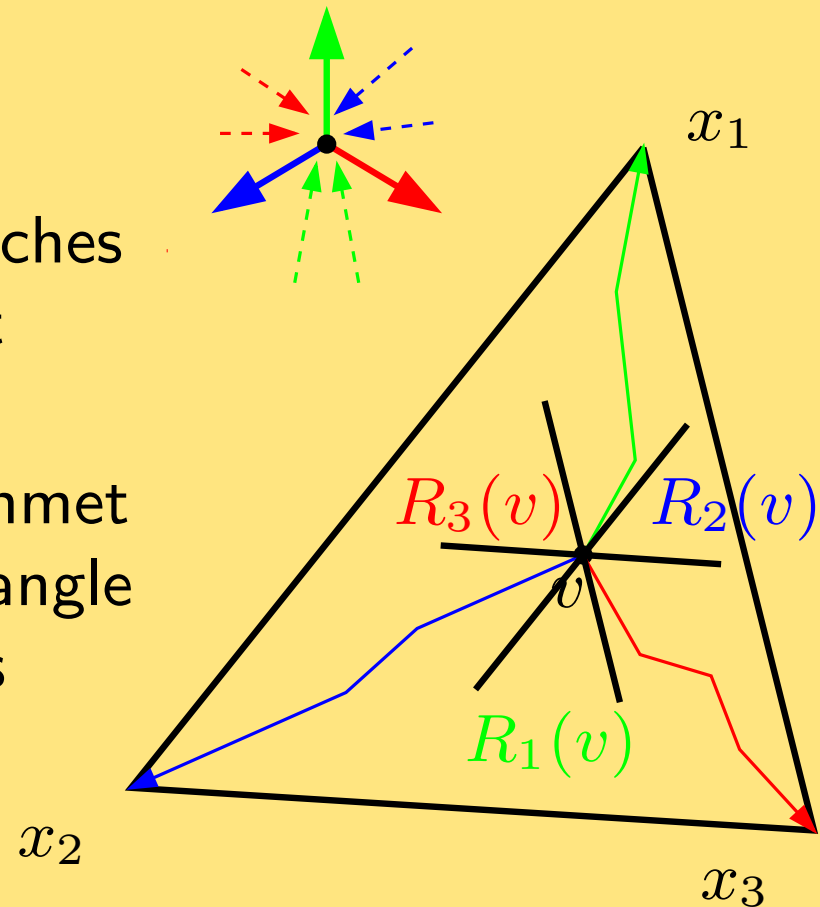
Pour une triangulation à  $n$  sommets,  $|T| = 2n - 5$ .

En prenant  $x_1 = (2n - 5, 0)$ ,  $x_2 = (0, 0)$  et  $x_3 = (0, 2n - 5)$ , on obtient un dessin sur une grille  $(2n - 5) \times (2n - 5)$ .

Idée de la preuve...

Il suffit de montrer que les branches sont orientées géométriquement comme on l'espère.

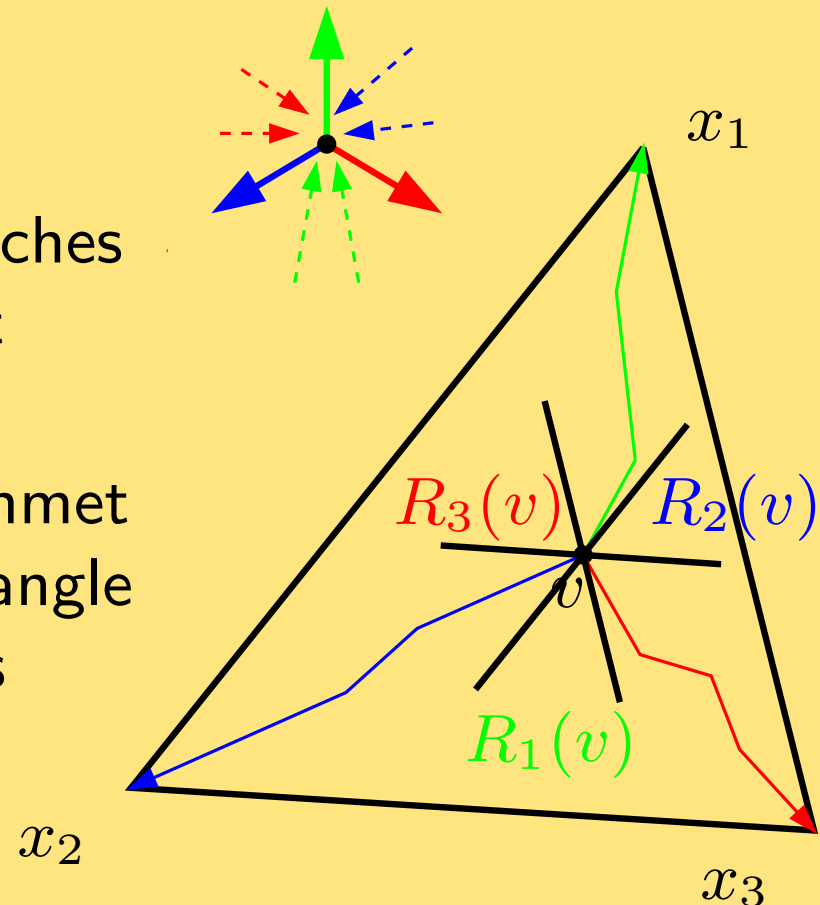
**Lemme.** Au voisinage d'un sommet les parallèles aux 3 côtés du triangle  $(x_1, x_2, x_3)$  séparent les 6 types d'arêtes.



Idée de la preuve...

Il suffit de montrer que les branches sont orientées géométriquement comme on l'espère.

**Lemme.** Au voisinage d'un sommet les parallèles aux 3 côtés du triangle  $(x_1, x_2, x_3)$  séparent les 6 types d'arêtes.



Ceci permet de montrer que les régions  $R_i(v)$  du nouveau dessin contiennent les mêmes sommets que les  $R_i(v)$  du dessin initial.

Si une arête croise l'une des 3 arêtes issues de  $v$  elle intersecte 2 régions différentes dans le nouveau dessin et donc aussi dans l'ancien. Ceci contredit la planarité du dessin initial.

□

L'algorithme de dessin de Schnyder donne une première raison de s'intéresser aux forêts de Schnyder.



L'algorithme de dessin de Schnyder donne une première raison de s'intéresser aux forêts de Schnyder.

Une autre raison : le test d'adjacence entre sommets...

**Exercice.** utiliser une forêt de Schnyder pour décrire une structure de donnée `planar_triangulation` enrichie permettant de répondre à cette requête en temps constant.

L'algorithme de dessin de Schnyder donne une première raison de s'intéresser aux forêts de Schnyder.

Une autre raison : le test d'adjacence entre sommets...

**Exercice.** utiliser une forêt de Schnyder pour décrire une structure de donnée `planar_triangulation` enrichie permettant de répondre à cette requête en temps constant.

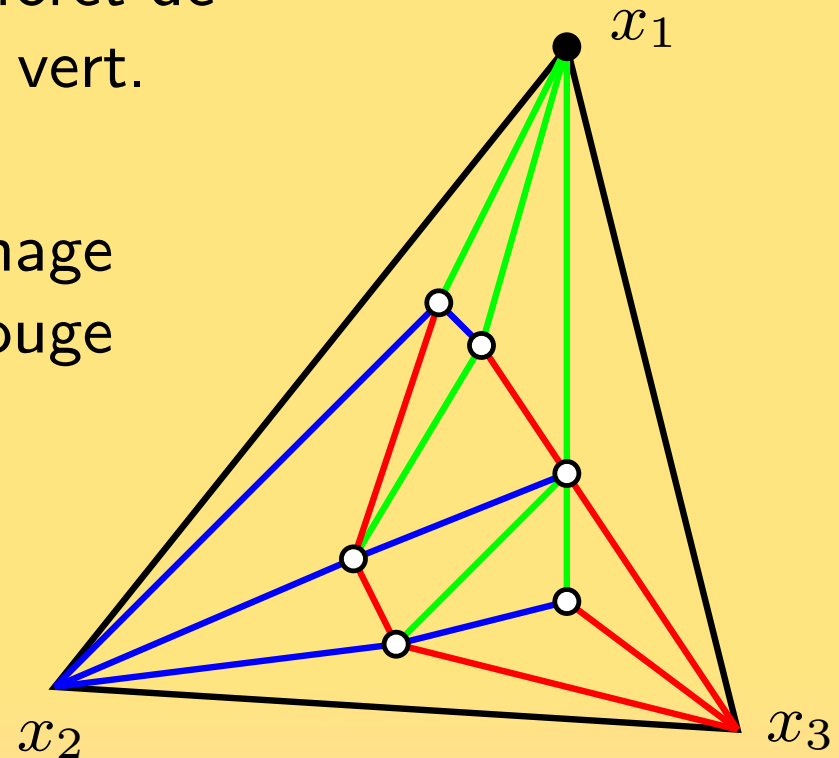
Ceci conduit à poser la question de l'existence de forêts de Schnyder.

## Forêts de Schnyder et ordres canoniques

Considérons l'épluchage d'une forêt de Schnyder par le haut de l'arbre vert.

**Lemme.** On peut faire l'épluchage en gardant des arbres bleu et rouge couvrants.

**Corollaire.** On peut donc garder toujours un bord simple (sans point d'articulation).



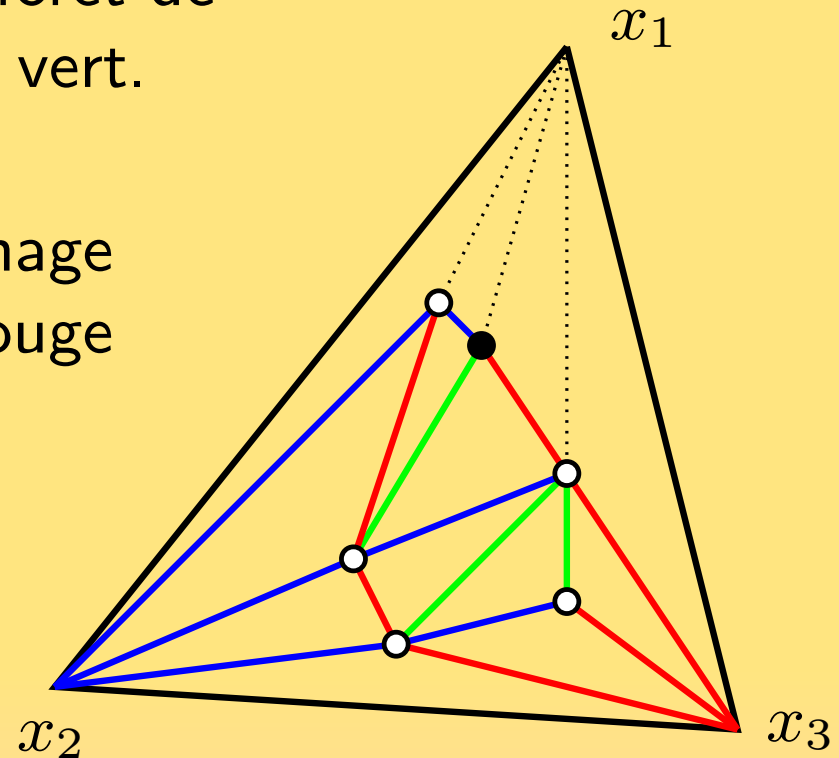
**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

Considérons l'épluchage d'une forêt de Schnyder par le haut de l'arbre vert.

**Lemme.** On peut faire l'épluchage en gardant des arbres bleu et rouge couvrants.

**Corollaire.** On peut donc garder toujours un bord simple (sans point d'articulation).



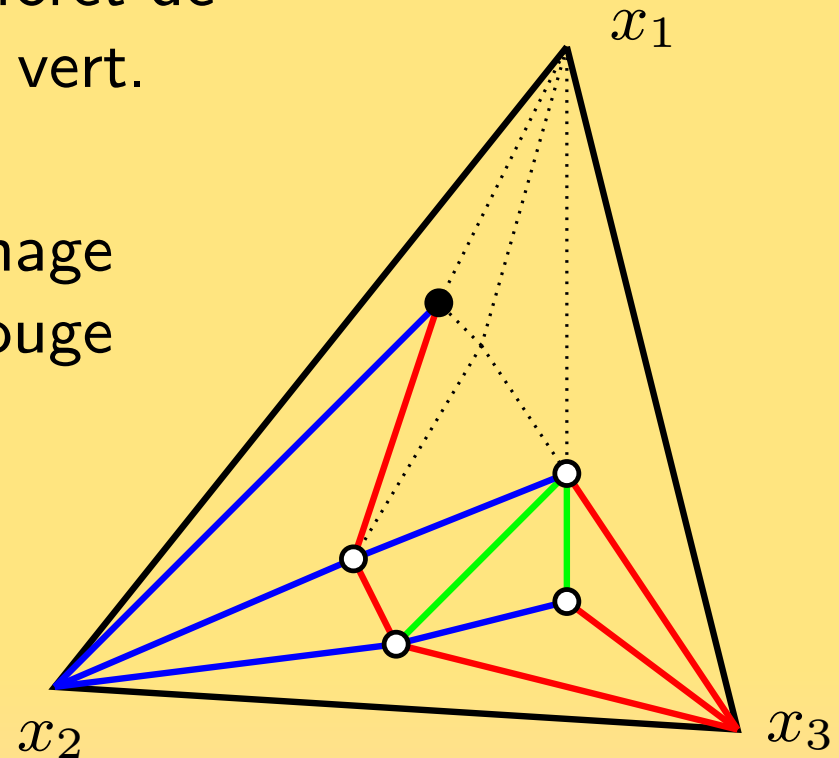
**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

Considérons l'épluchage d'une forêt de Schnyder par le haut de l'arbre vert.

**Lemme.** On peut faire l'épluchage en gardant des arbres bleu et rouge couvrants.

**Corollaire.** On peut donc garder toujours un bord simple (sans point d'articulation).



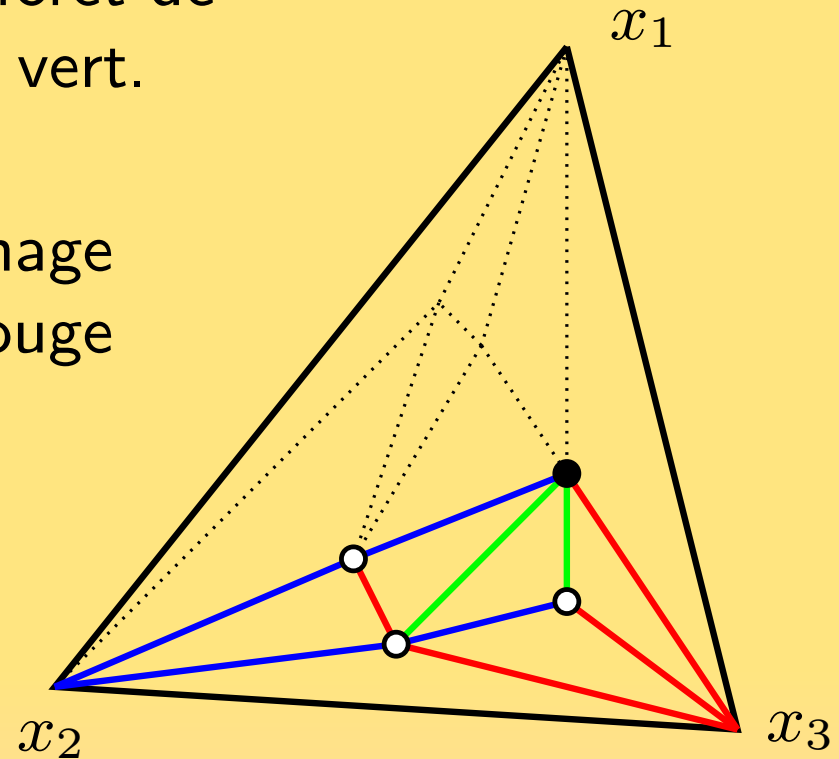
**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

Considérons l'épluchage d'une forêt de Schnyder par le haut de l'arbre vert.

**Lemme.** On peut faire l'épluchage en gardant des arbres bleu et rouge couvrants.

**Corollaire.** On peut donc garder toujours un bord simple (sans point d'articulation).



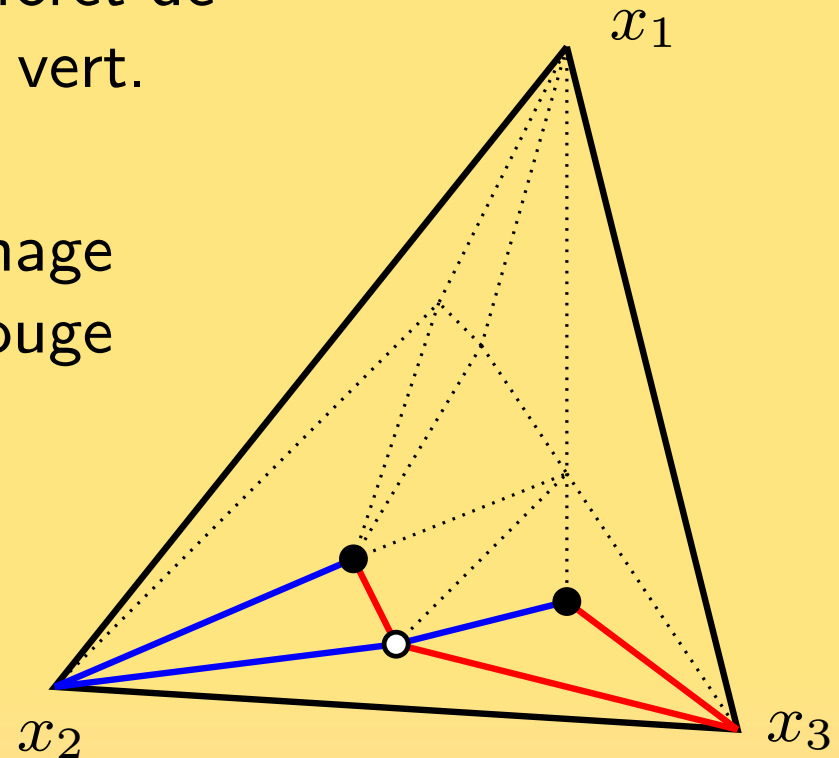
**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

Considérons l'épluchage d'une forêt de Schnyder par le haut de l'arbre vert.

**Lemme.** On peut faire l'épluchage en gardant des arbres bleu et rouge couvrants.

**Corollaire.** On peut donc garder toujours un bord simple (sans point d'articulation).



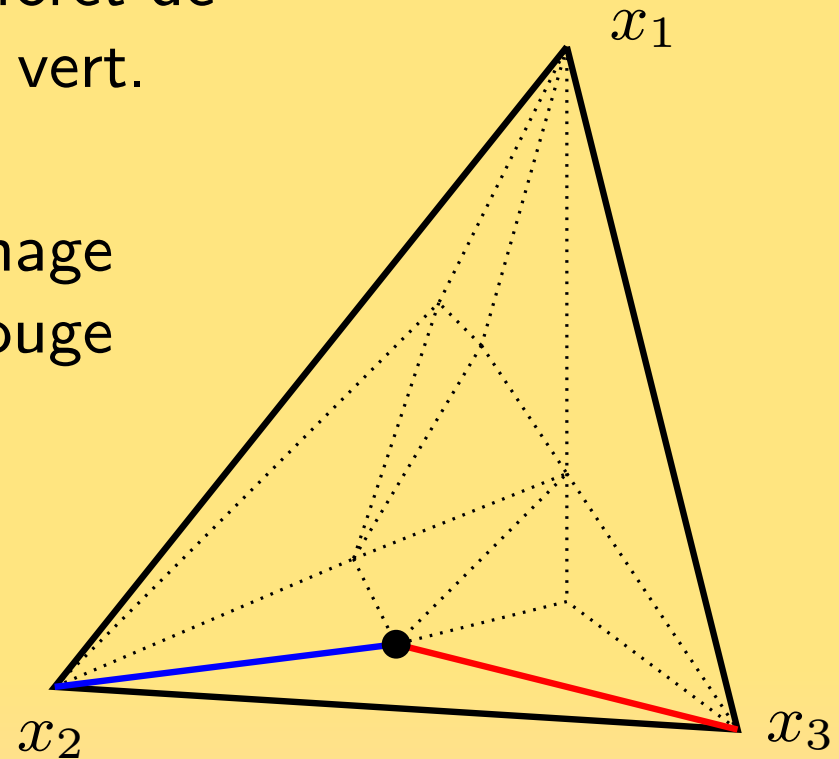
**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

Considérons l'épluchage d'une forêt de Schnyder par le haut de l'arbre vert.

**Lemme.** On peut faire l'épluchage en gardant des arbres bleu et rouge couvrants.

**Corollaire.** On peut donc garder toujours un bord simple (sans point d'articulation).



**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

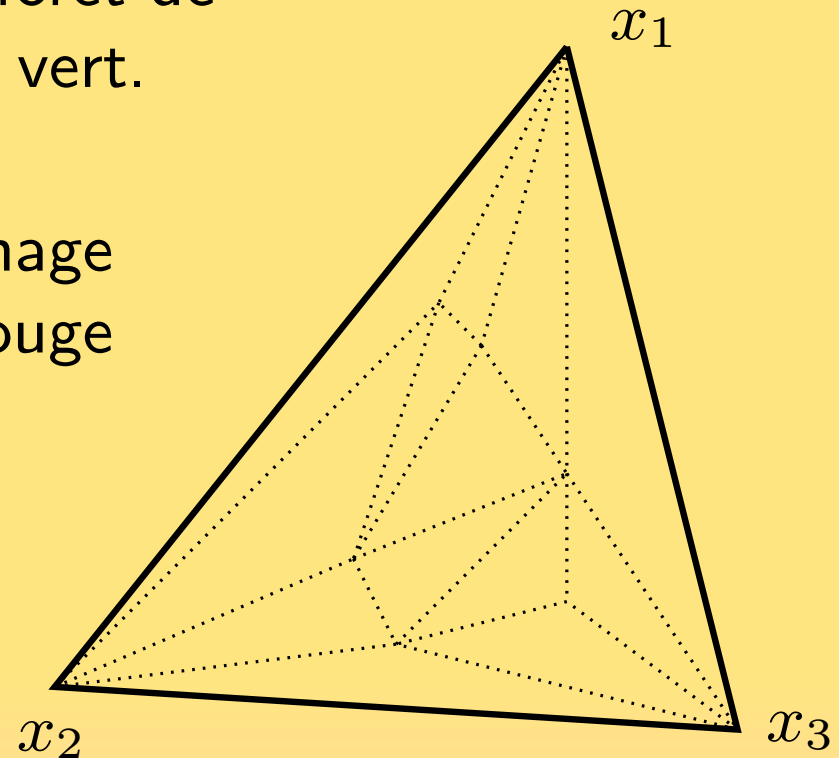


## Forêts de Schnyder et ordres canoniques

Considérons l'épluchage d'une forêt de Schnyder par le haut de l'arbre vert.

**Lemme.** On peut faire l'épluchage en gardant des arbres bleu et rouge couvrants.

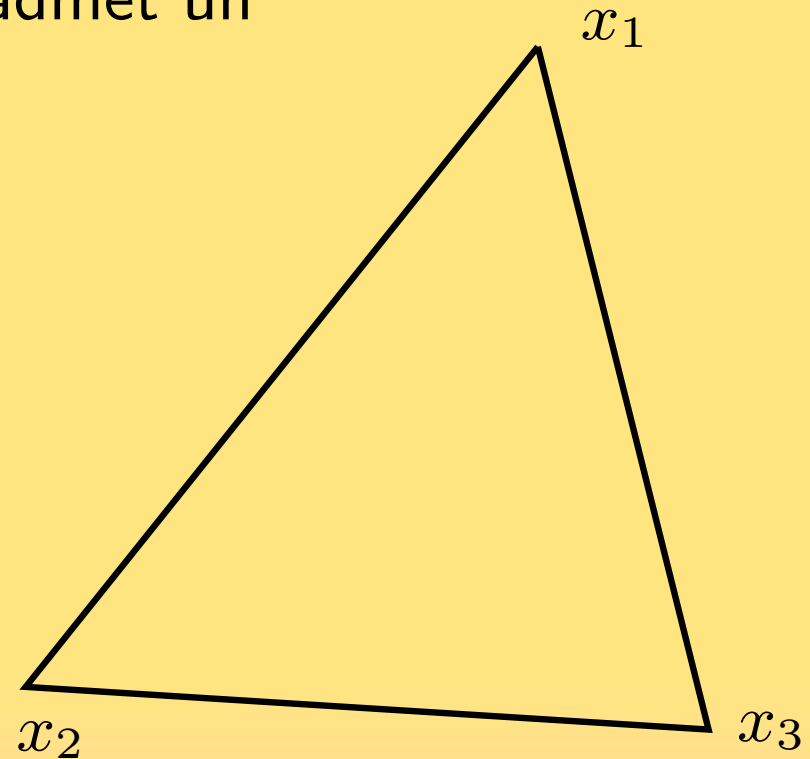
**Corollaire.** On peut donc garder toujours un bord simple (sans point d'articulation).



**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

**Théorème.** Toute triangulation admet un ordre canonique.

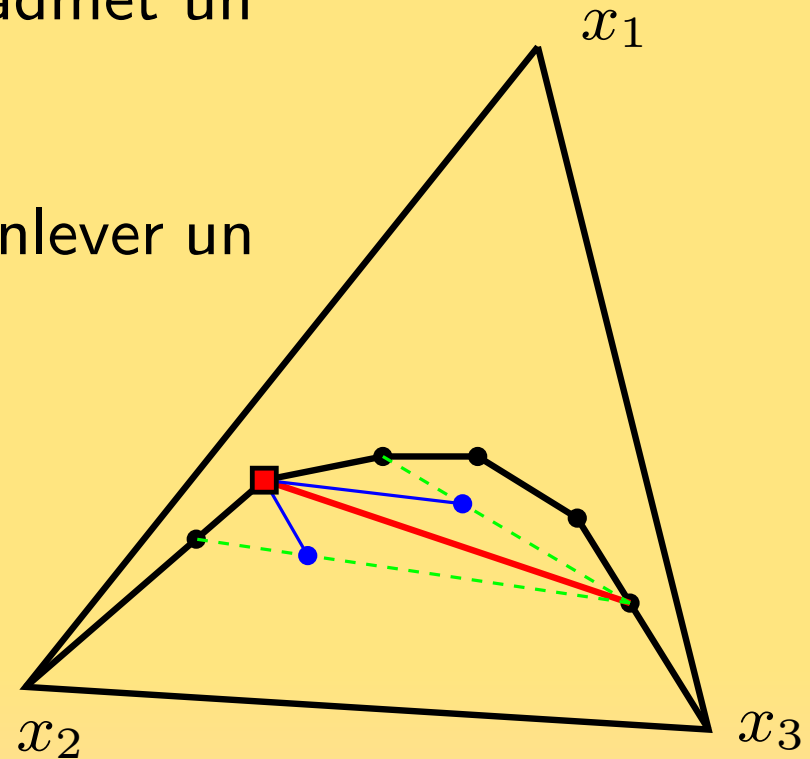


**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

**Théorème.** Toute triangulation admet un ordre canonique.

À chaque étape il faut éviter d'enlever un sommet qui forme une corde.

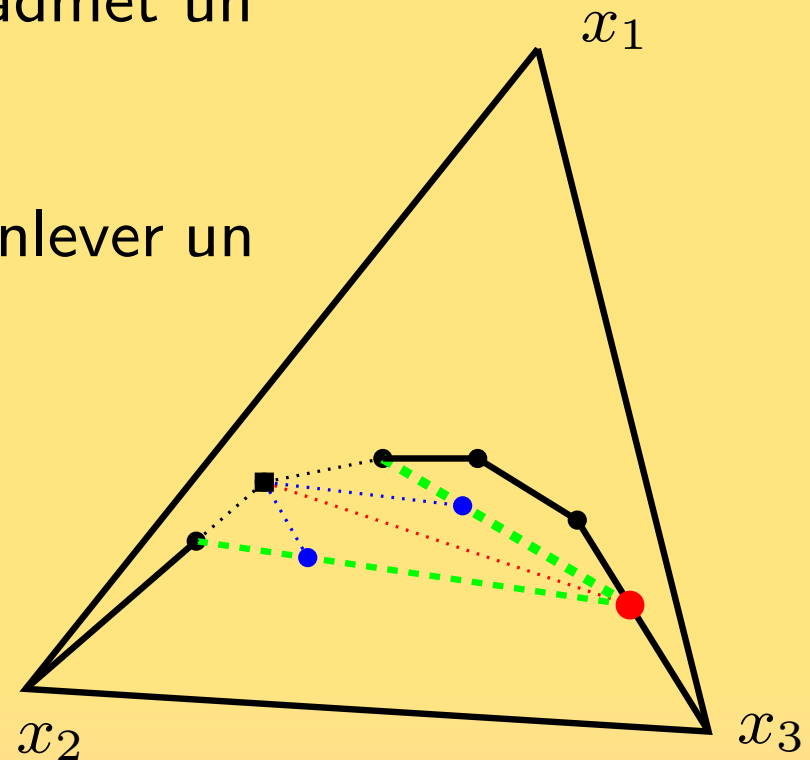


**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

**Théorème.** Toute triangulation admet un ordre canonique.

À chaque étape il faut éviter d'enlever un sommet qui forme une corde.



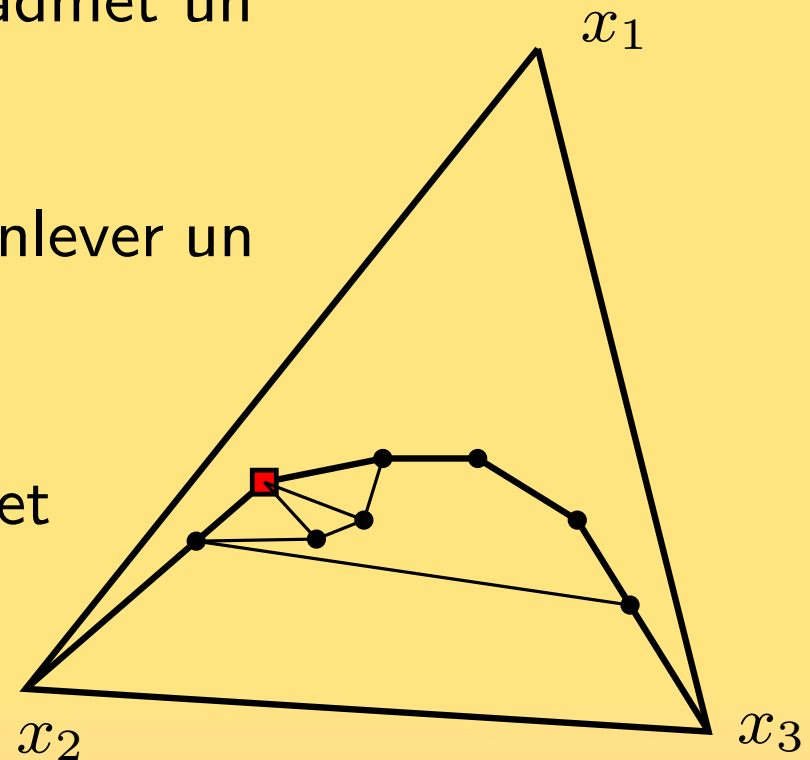
**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

## Forêts de Schnyder et ordres canoniques

**Théorème.** Toute triangulation admet un ordre canonique.

À chaque étape il faut éviter d'enlever un sommet qui forme une corde.

Mais il existe toujours un sommet non bloqué.



**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

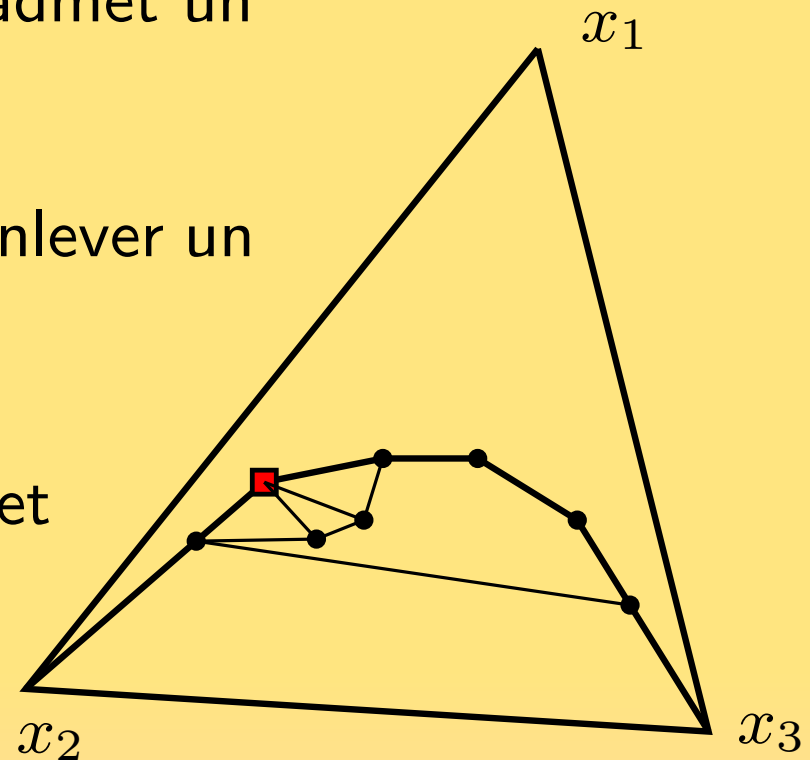
## Forêts de Schnyder et ordres canoniques

**Théorème.** Toute triangulation admet un ordre canonique.

À chaque étape il faut éviter d'enlever un sommet qui forme une corde.

Mais il existe toujours un sommet non bloqué.

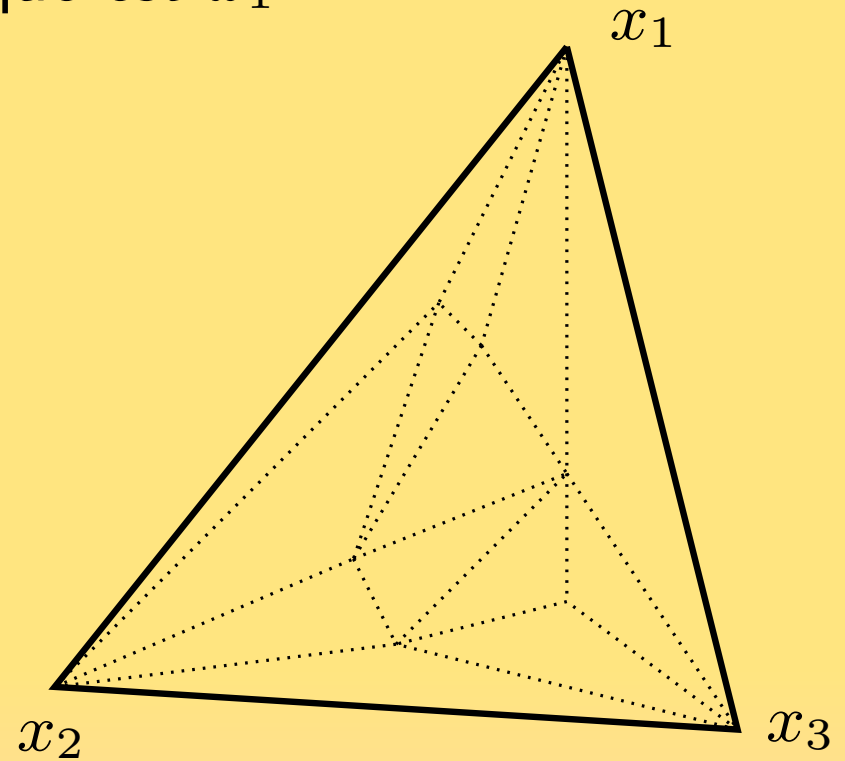
⇒ l'algorithme glouton suffit !



**Définition.** Un ordre  $\{x_1, v_1, \dots, v_n, x_2, x_3\}$  est **canonique** si  $T \setminus \{x_1, v_1, \dots, v_k\}$  a un bord simple pour tout  $k$ .

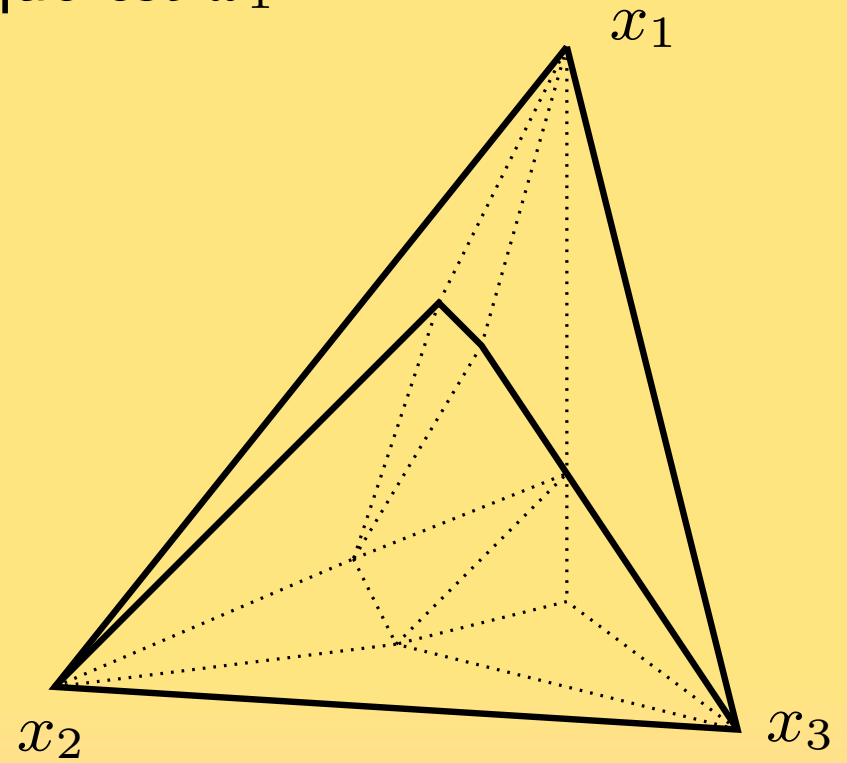
# Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .



# Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .

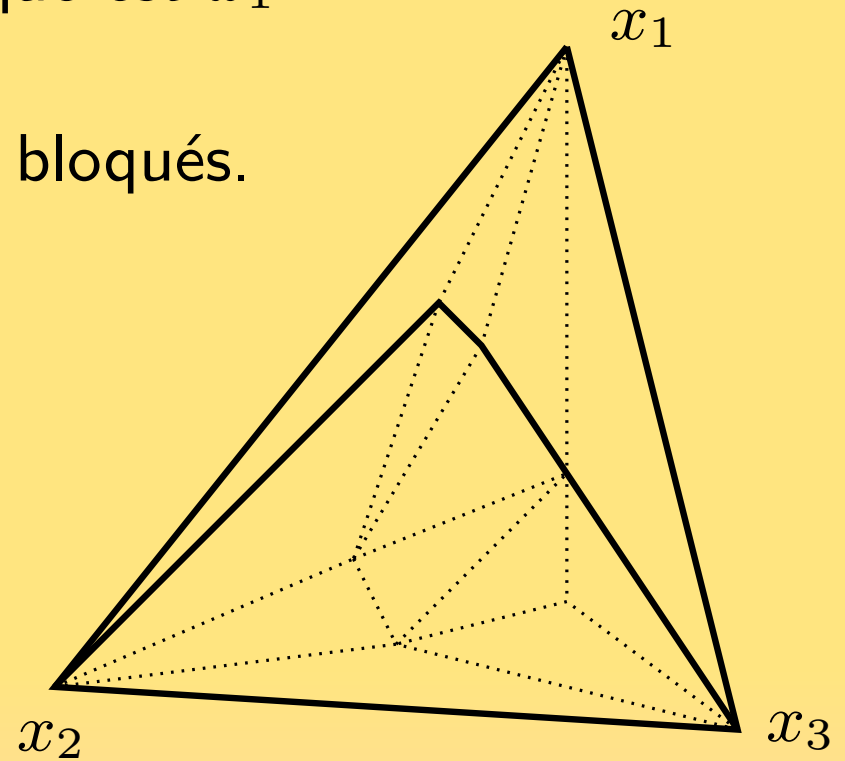




## Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .

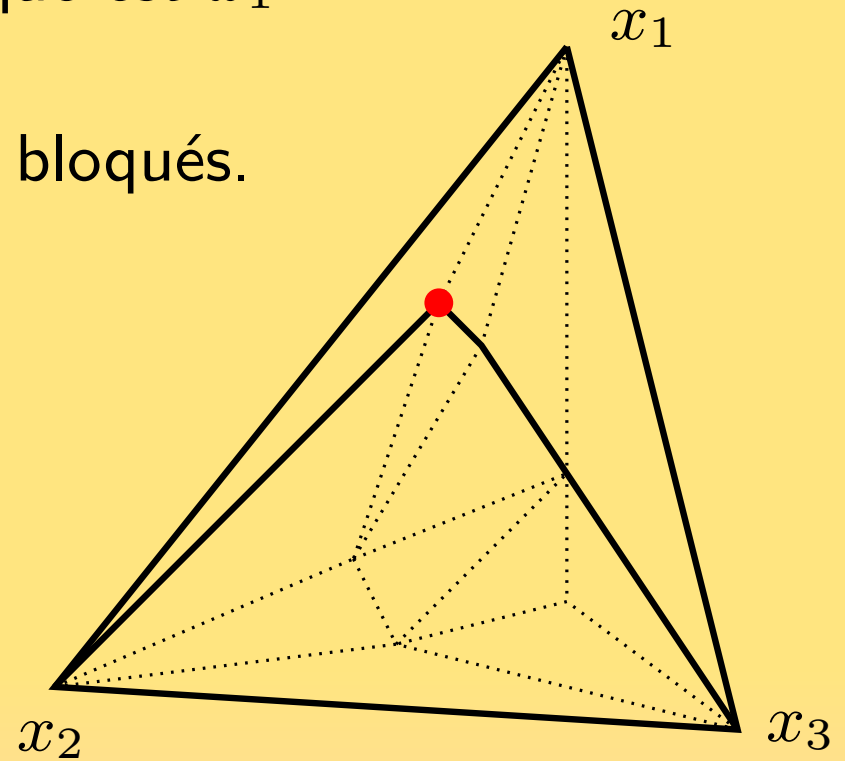
On choisit  $x_1$  parmi les 3 non bloqués.



## Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .

On choisit  $x_1$  parmi les 3 non bloqués.

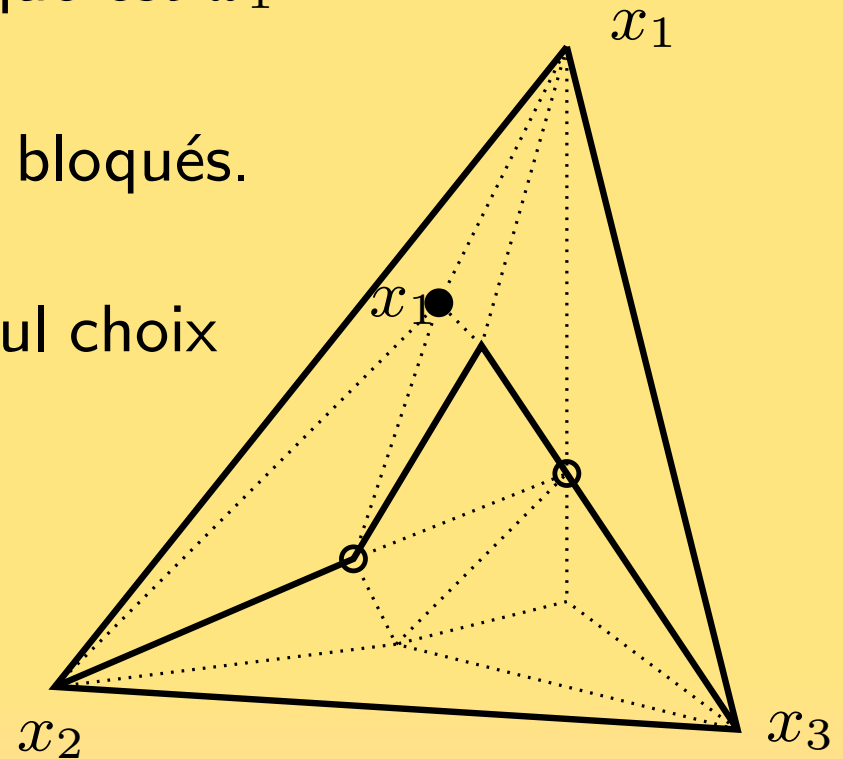


## Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .

On choisit  $x_1$  parmi les 3 non bloqués.

Il y a 2 sommets bloqués : 1 seul choix pour  $x_2$ .

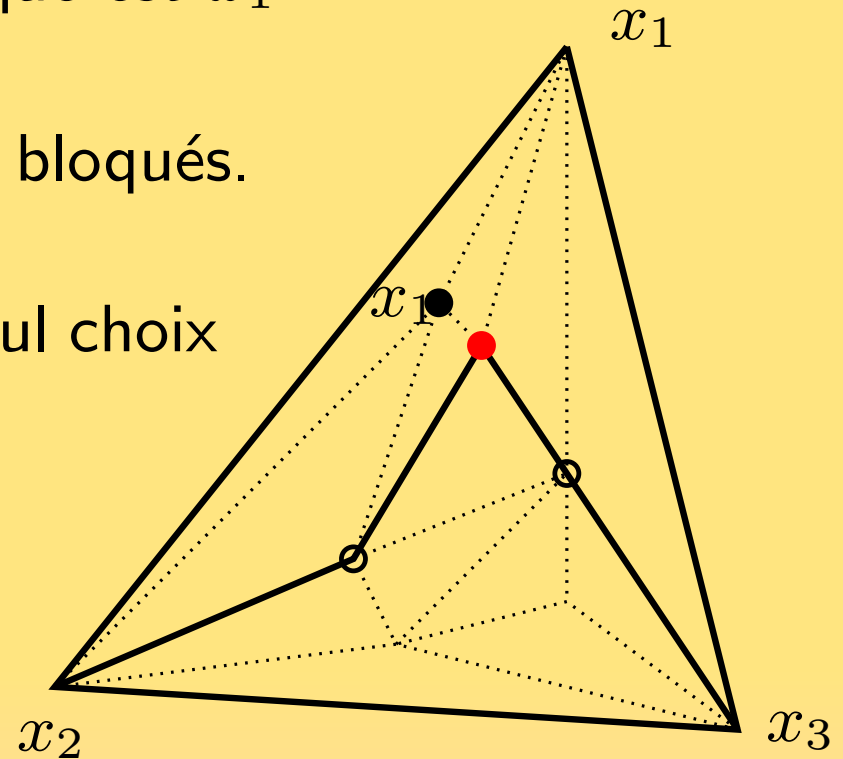


## Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .

On choisit  $x_1$  parmi les 3 non bloqués.

Il y a 2 sommets bloqués : 1 seul choix pour  $x_2$ .

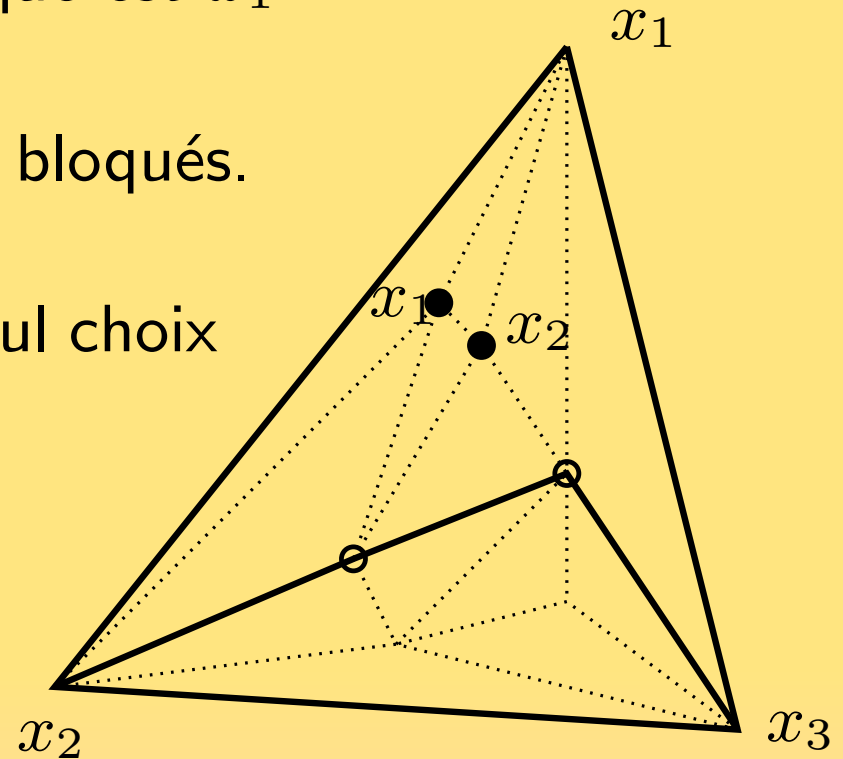


## Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .

On choisit  $x_1$  parmi les 3 non bloqués.

Il y a 2 sommets bloqués : 1 seul choix pour  $x_2$ .



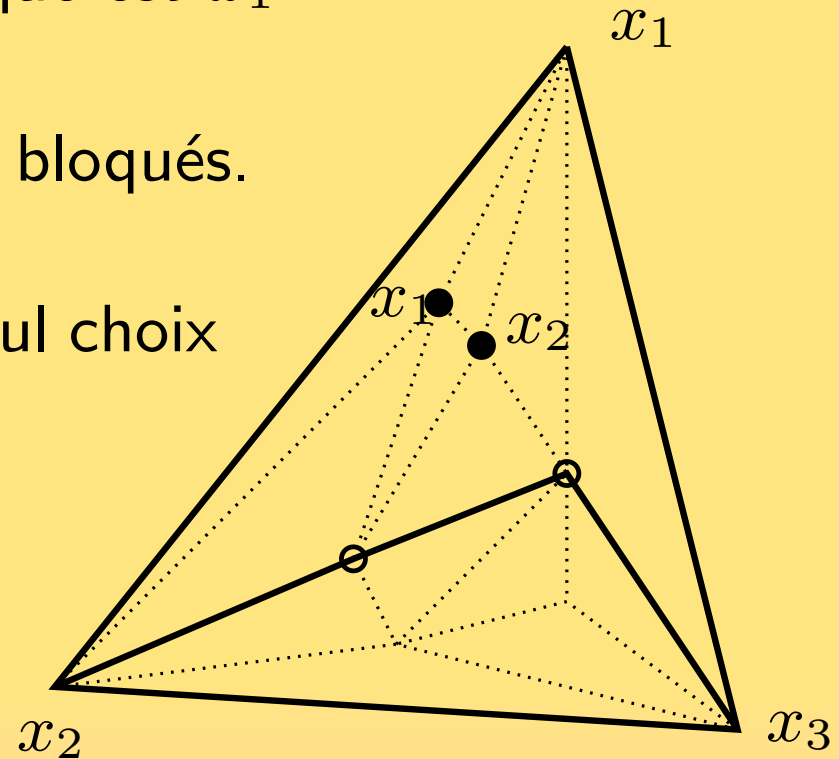
## Construction par l'algorithme glouton

Le premier dans l'ordre canonique est  $x_1$ .

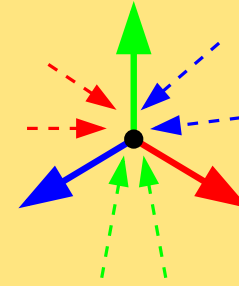
On choisit  $x_1$  parmi les 3 non bloqués.

Il y a 2 sommets bloqués : 1 seul choix pour  $x_2$ .

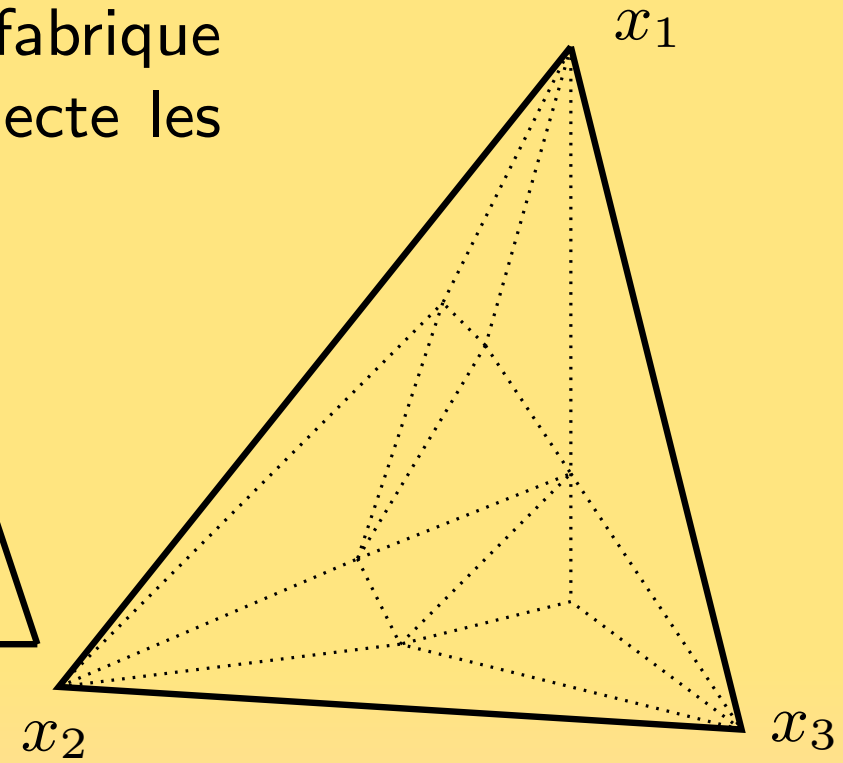
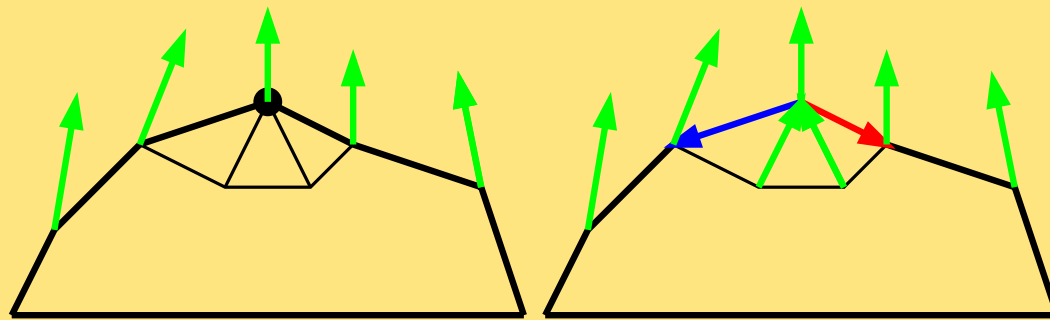
Les 2 sommets ne sont plus bloqués, on peut choisir l'un des 2 et continuer...



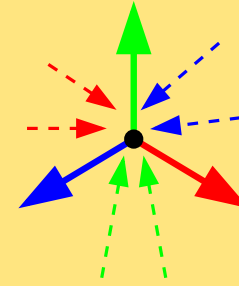
# Construction par l'algorithme glouton



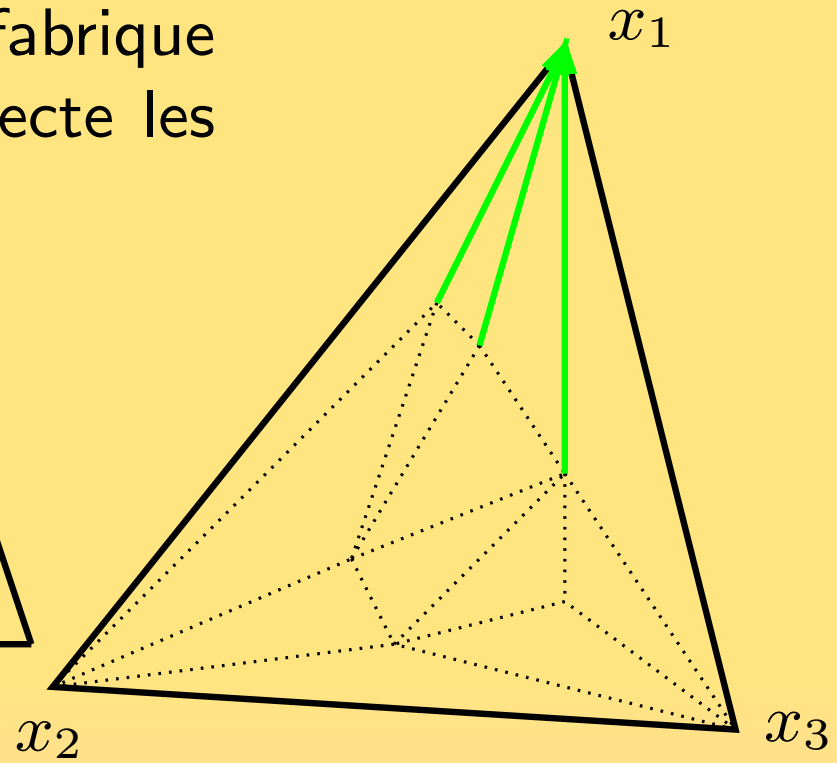
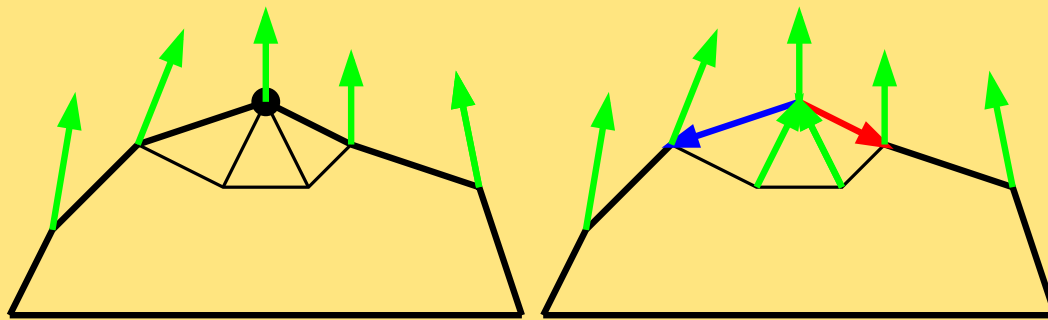
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



Construction par l'algorithme glouton

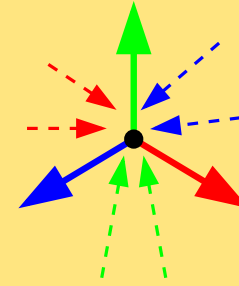


En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.

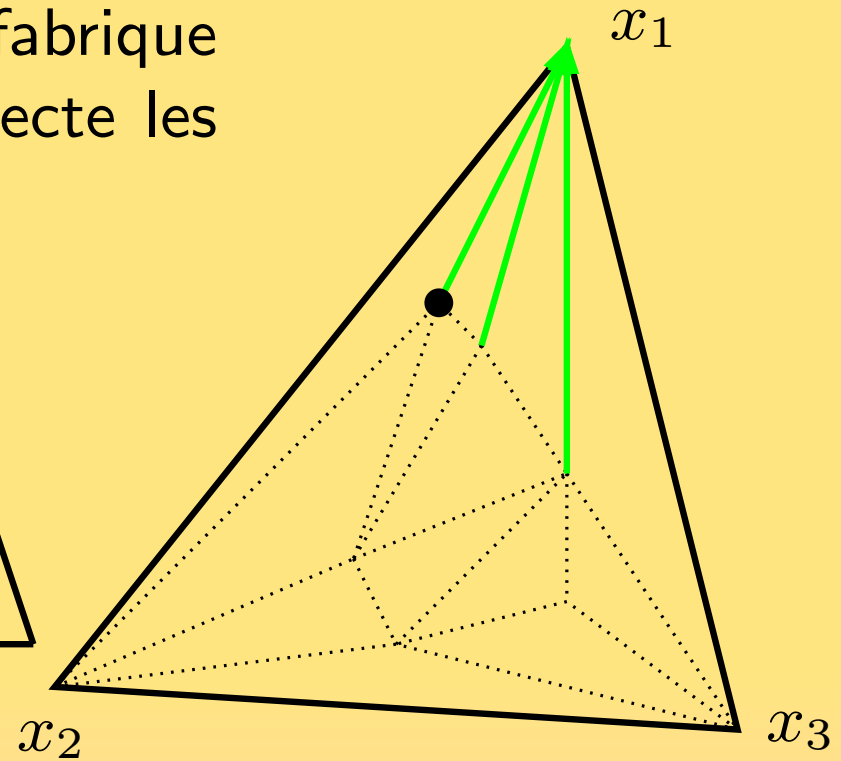
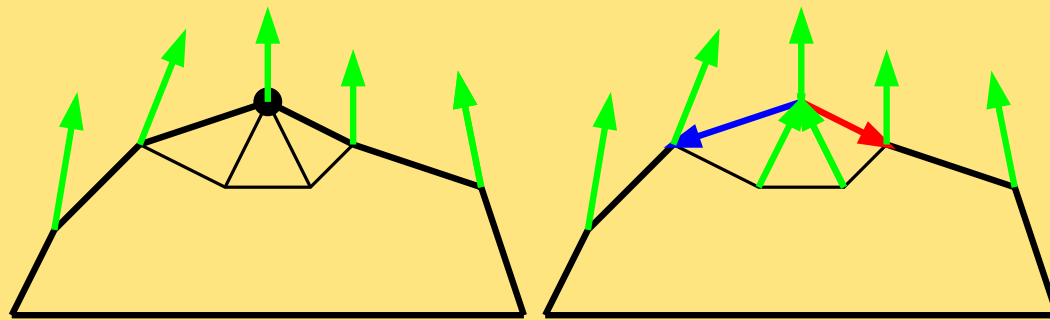




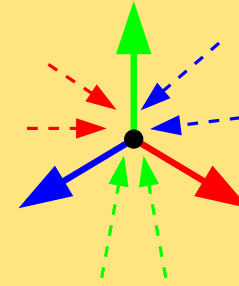
Construction par l'algorithme glouton



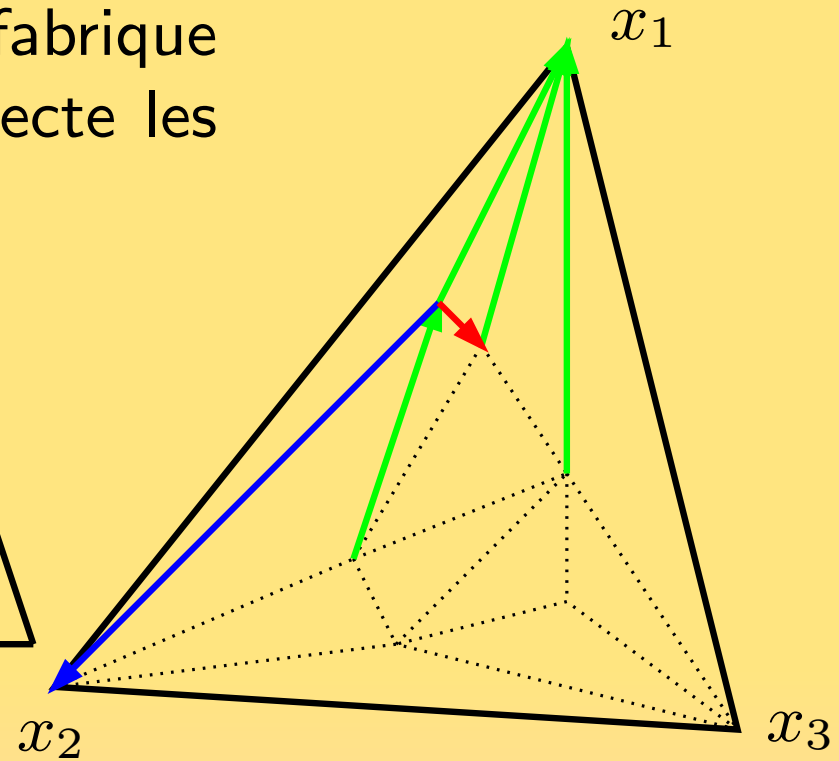
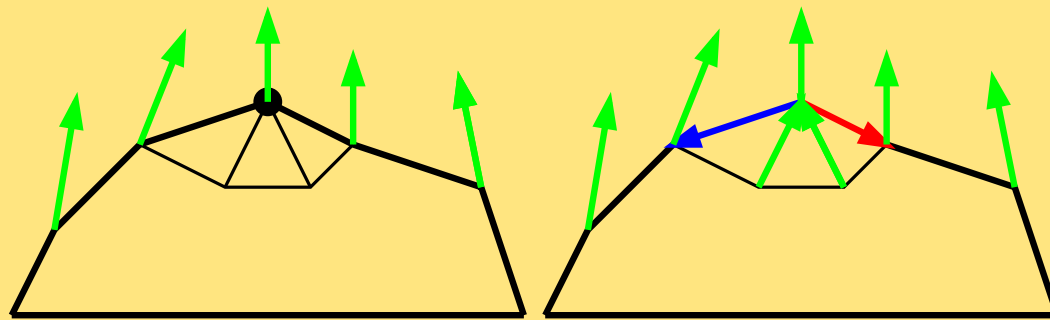
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



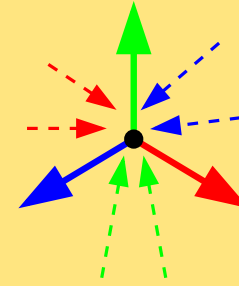
# Construction par l'algorithme glouton



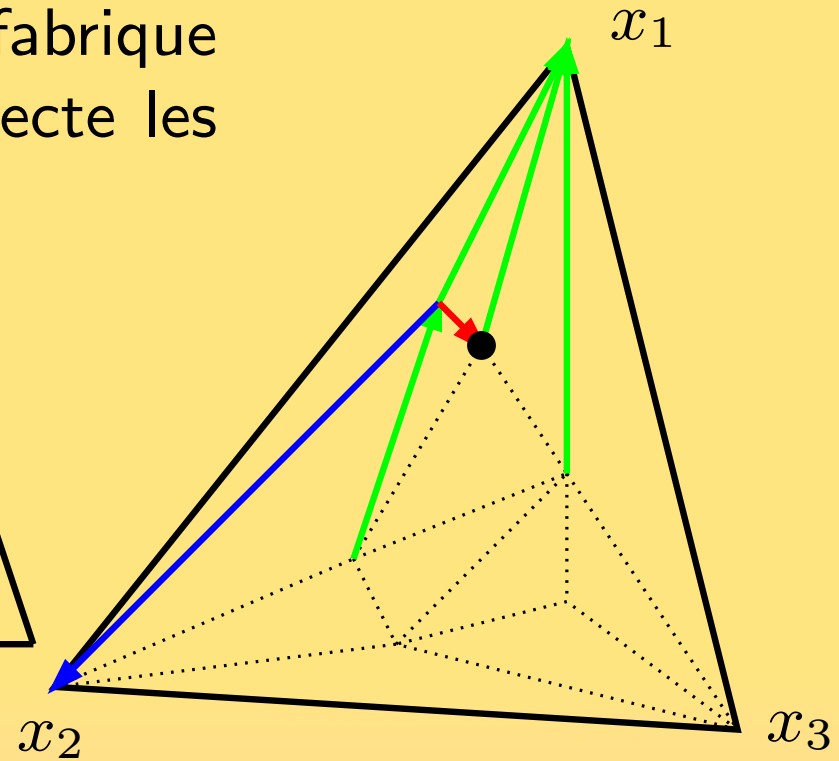
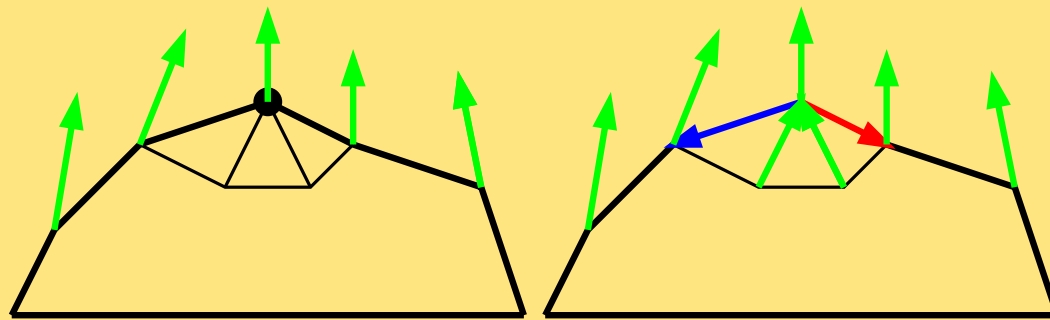
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



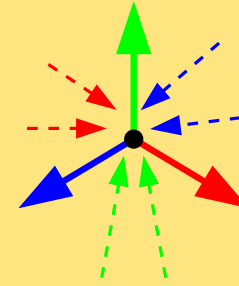
# Construction par l'algorithme glouton



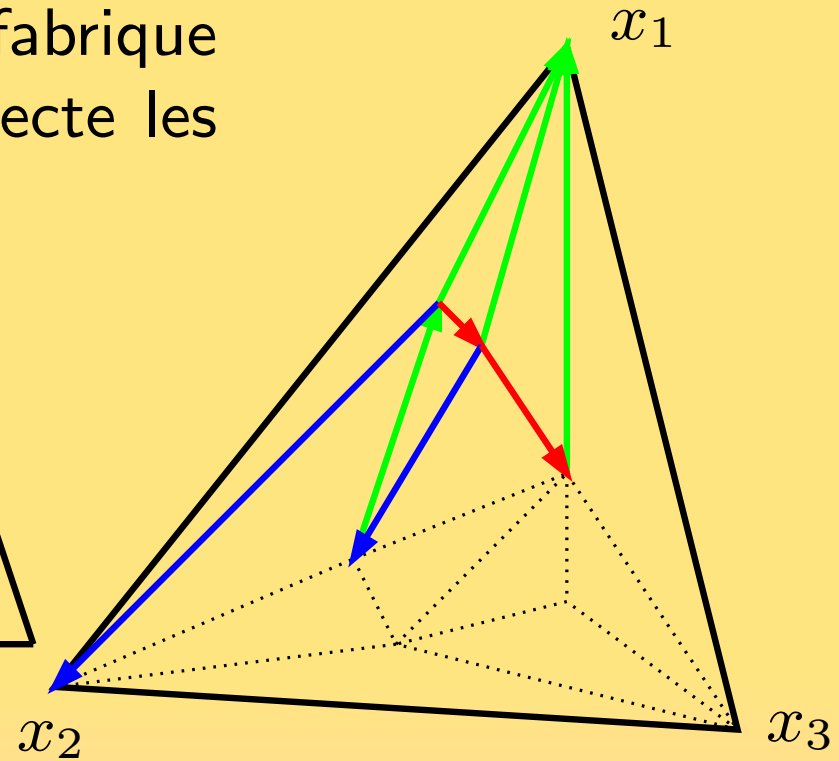
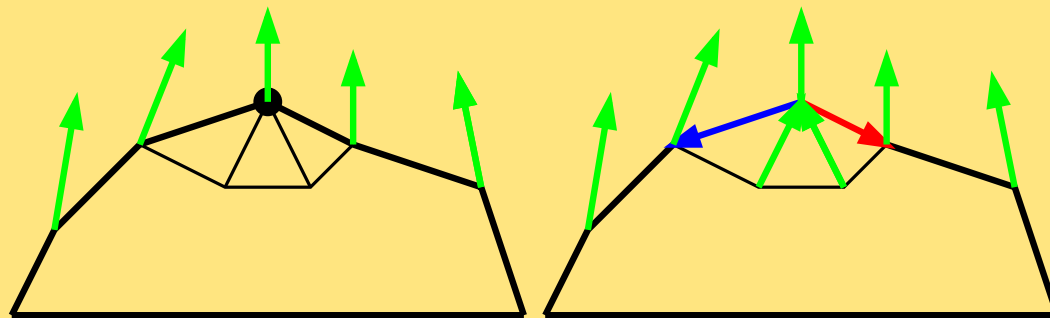
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



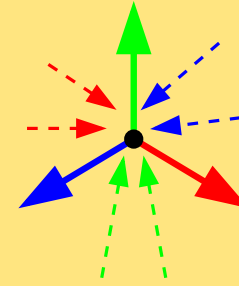
# Construction par l'algorithme glouton



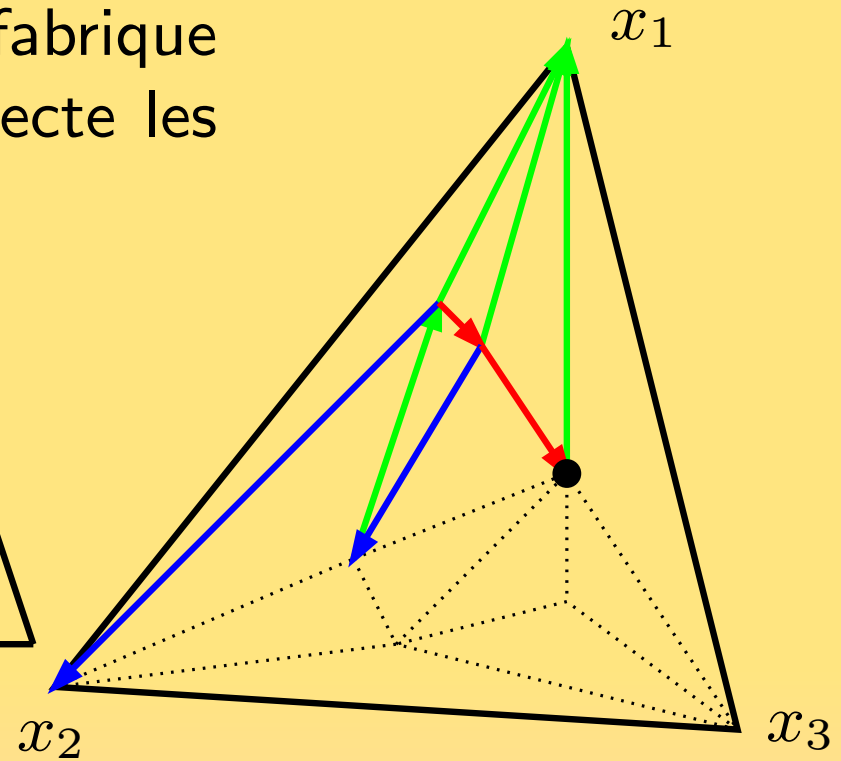
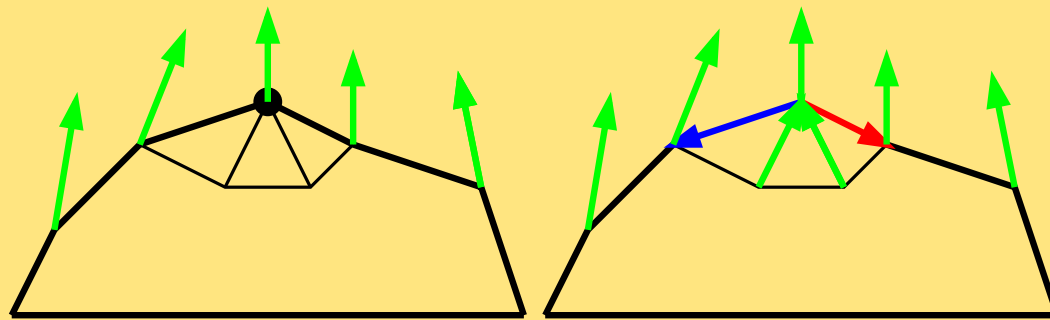
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



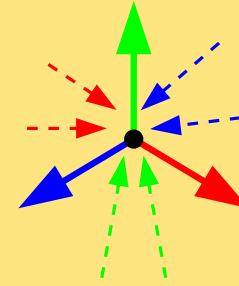
# Construction par l'algorithme glouton



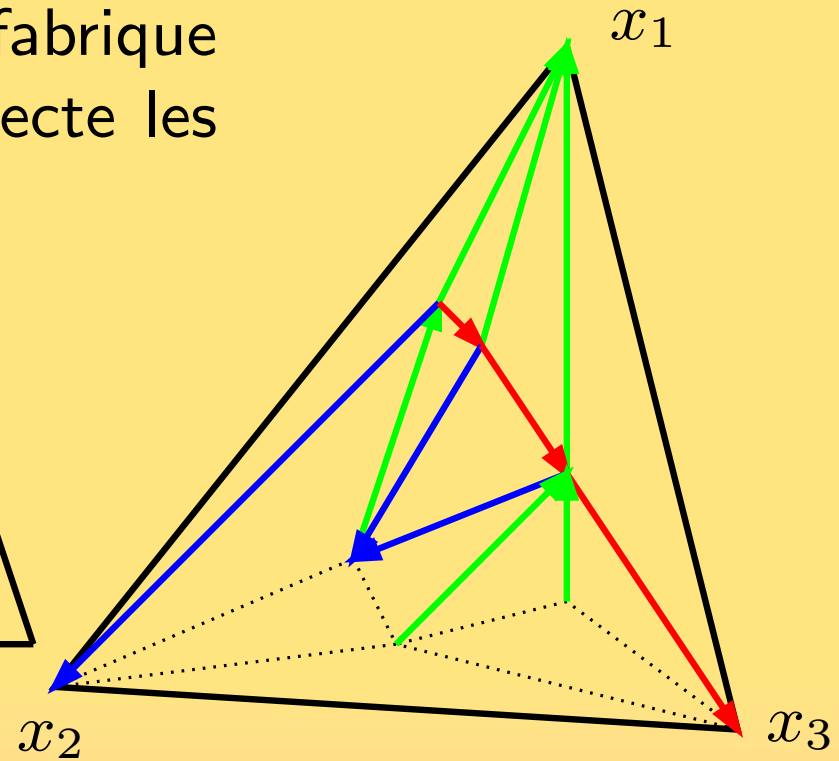
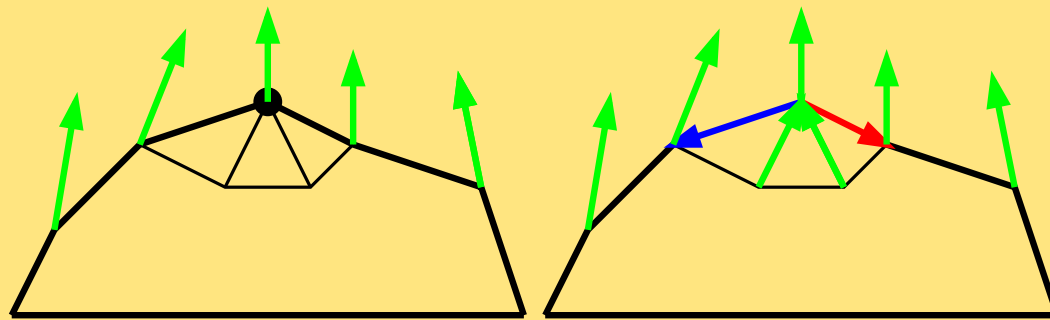
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



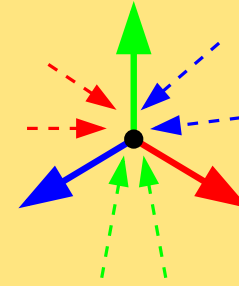
# Construction par l'algorithme glouton



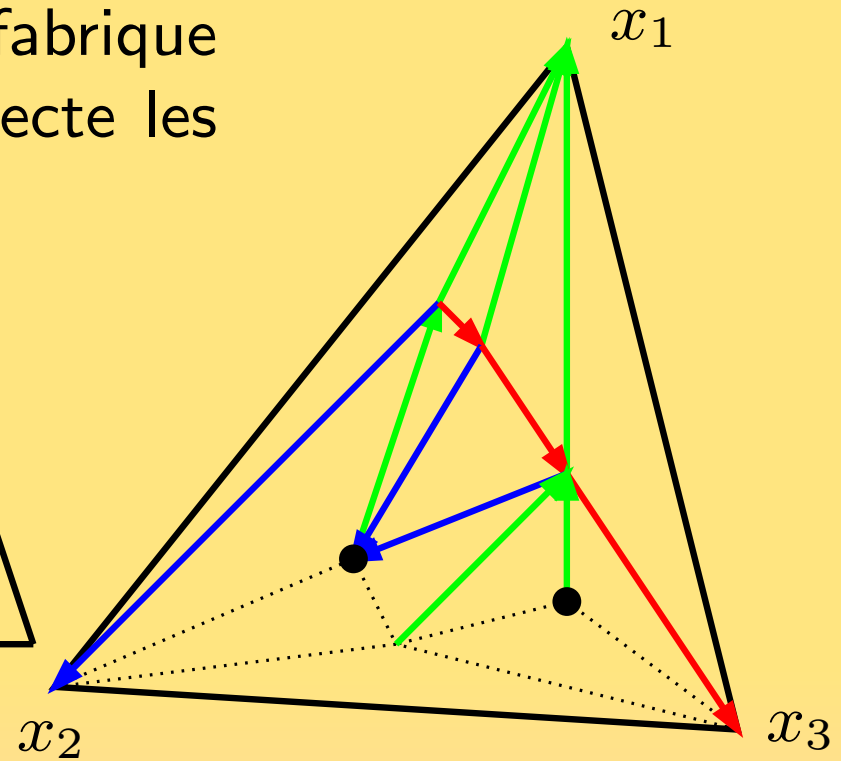
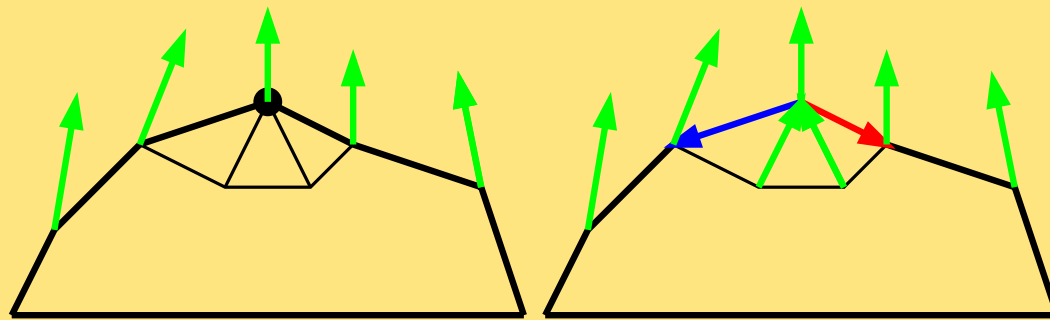
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



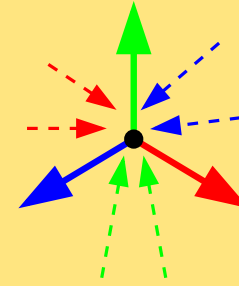
# Construction par l'algorithme glouton



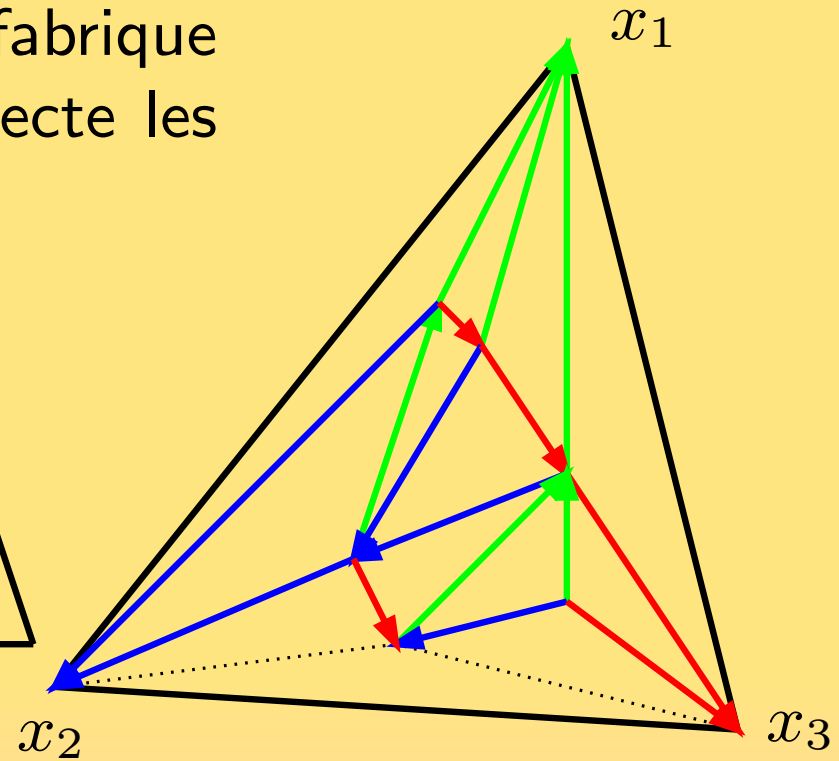
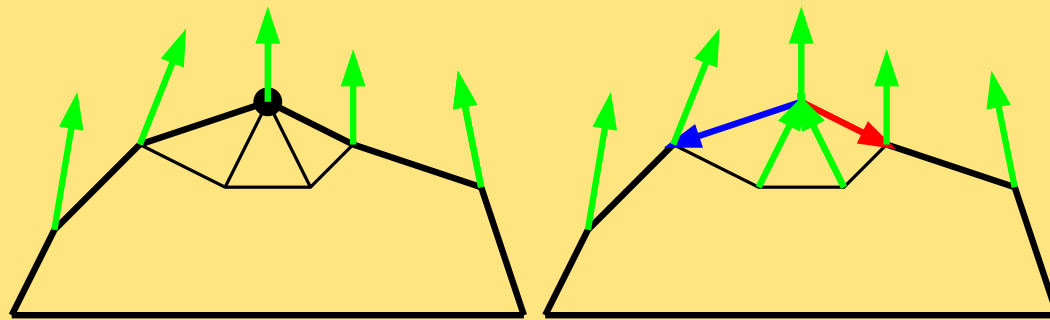
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



# Construction par l'algorithme glouton

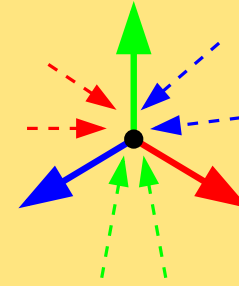


En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.

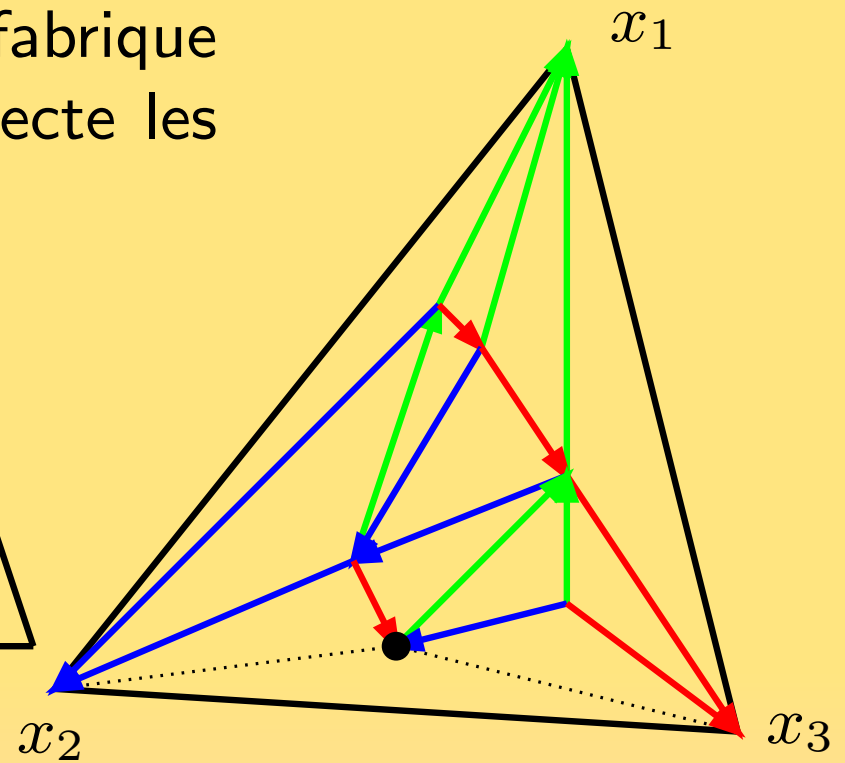
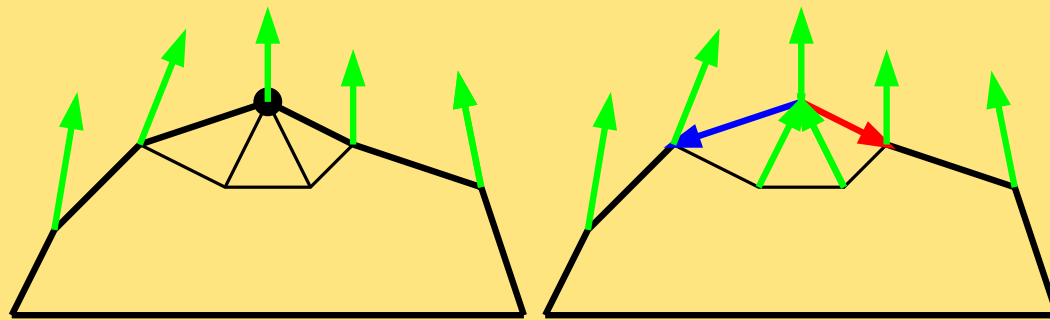




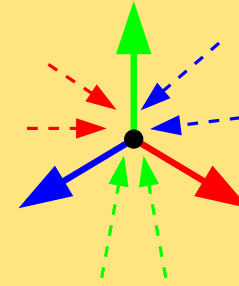
# Construction par l'algorithme glouton



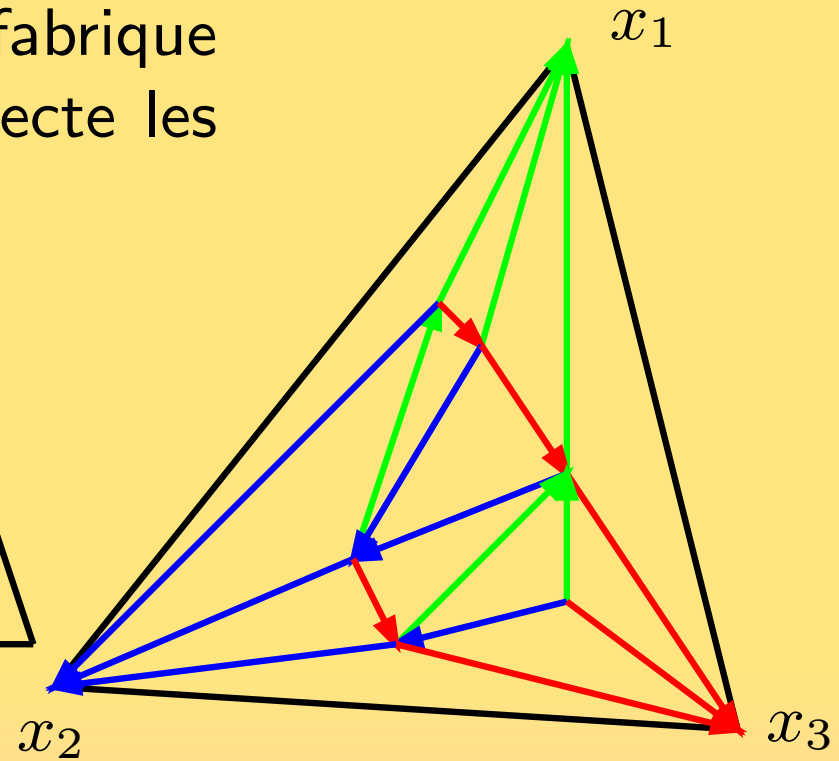
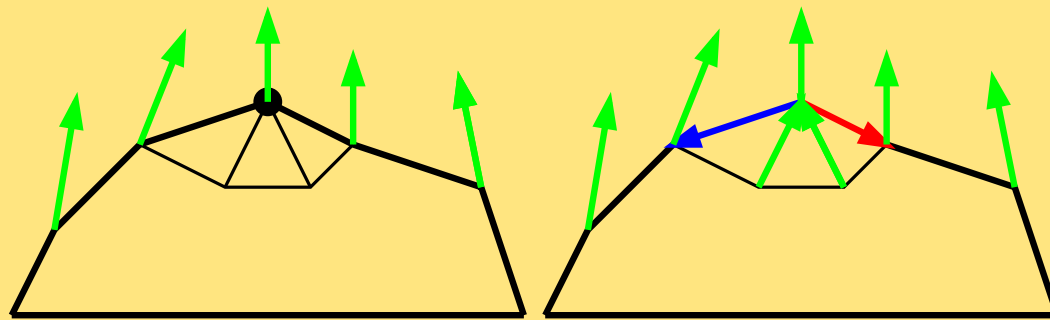
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



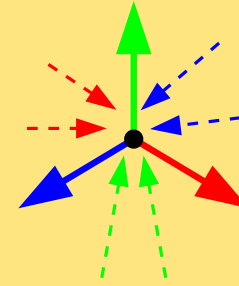
# Construction par l'algorithme glouton



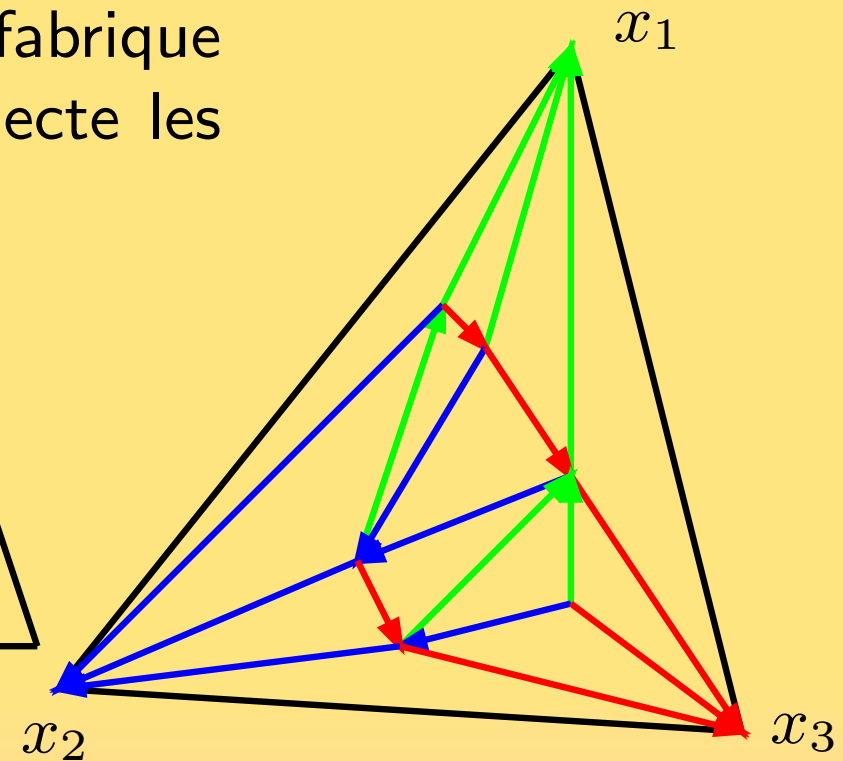
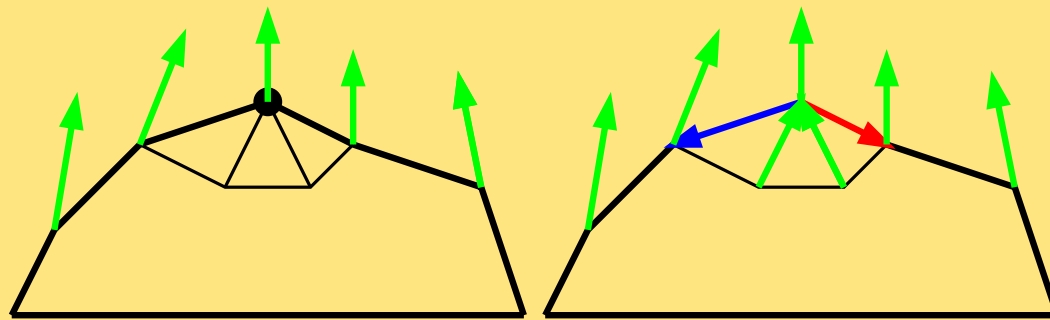
En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



Construction par l'algorithme glouton

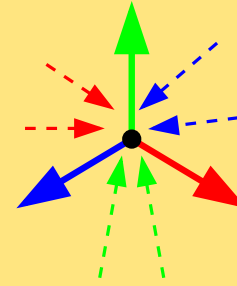


En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.

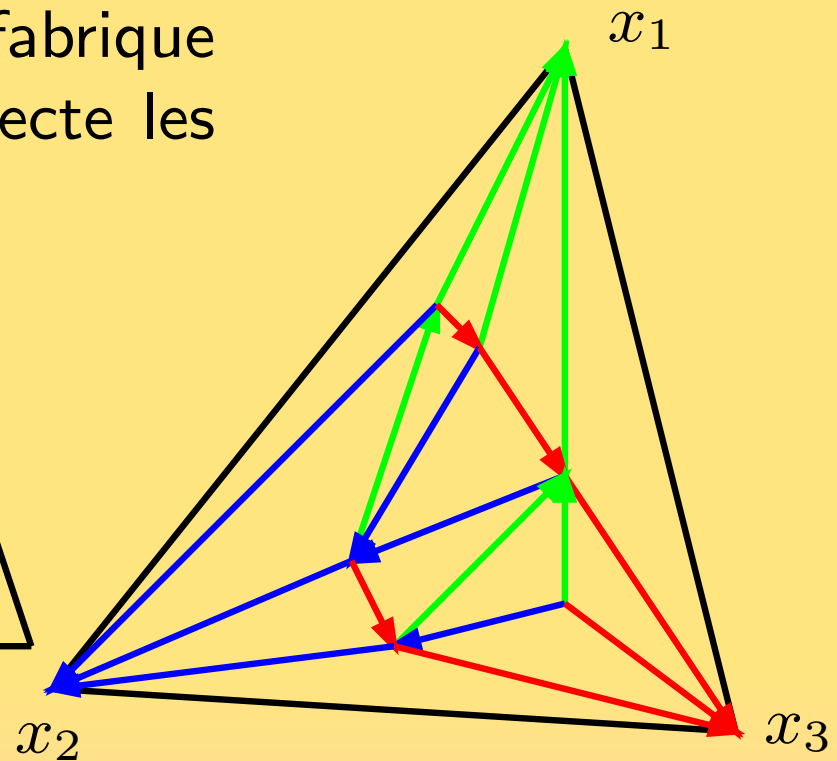
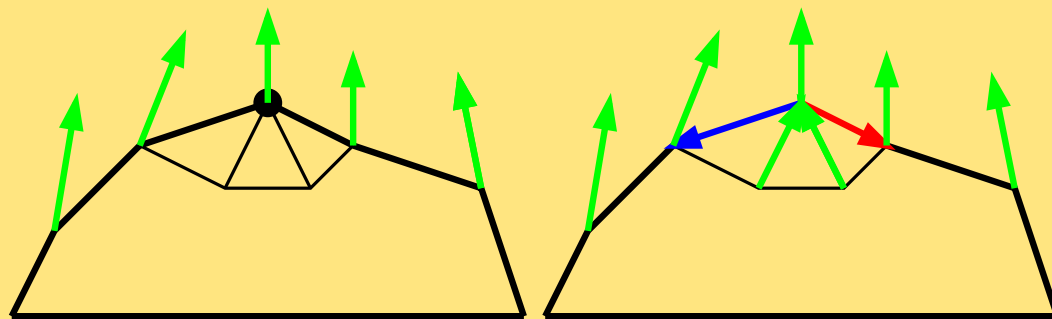


**Lemme.** Les arêtes de chaque couleur forment un arbre.

Construction par l'algorithme glouton



En remettant les couleurs, on fabrique une coloration orientée qui respecte les conditions locales de Schnyder.



**Lemme.** Les arêtes de chaque couleur forment un arbre.

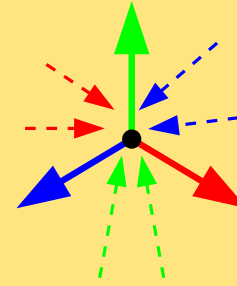
**Corollaire.** Les orientations canoniques sont en bijection avec les forêts de Schnyder.

On a donc montré que toute triangulation admet une décomposition en forêt de Schnyder.

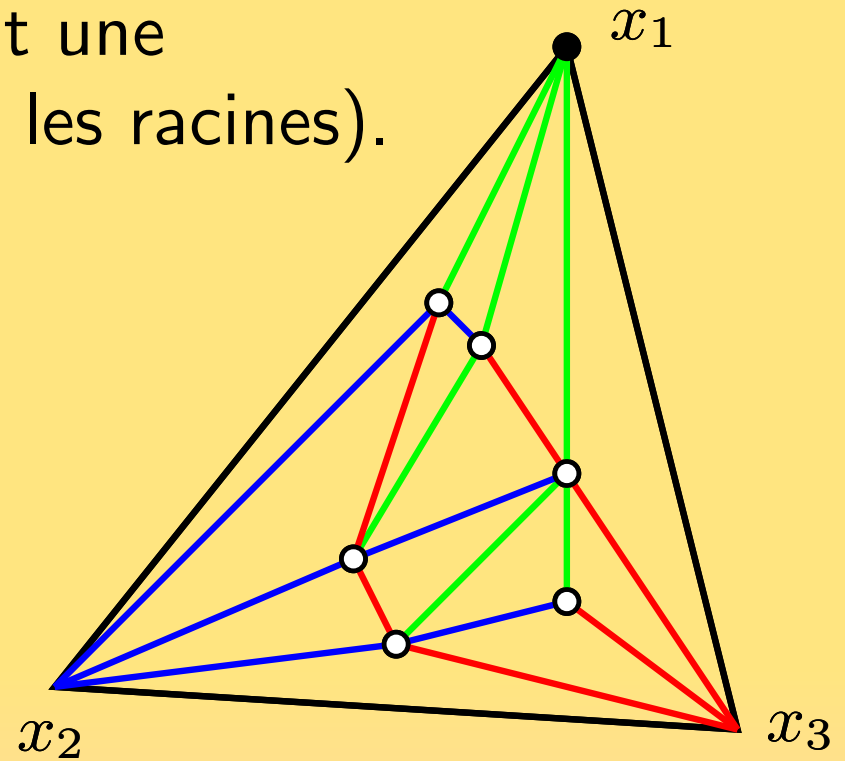
En général il y a plusieurs décompositions possibles.

Pour finir cette première partie, on voudrait comparer ces différentes décompositions.

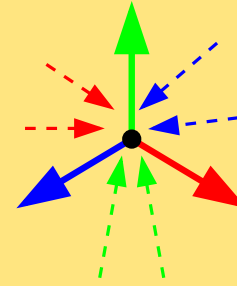
# 3-orientations d'une triangulation



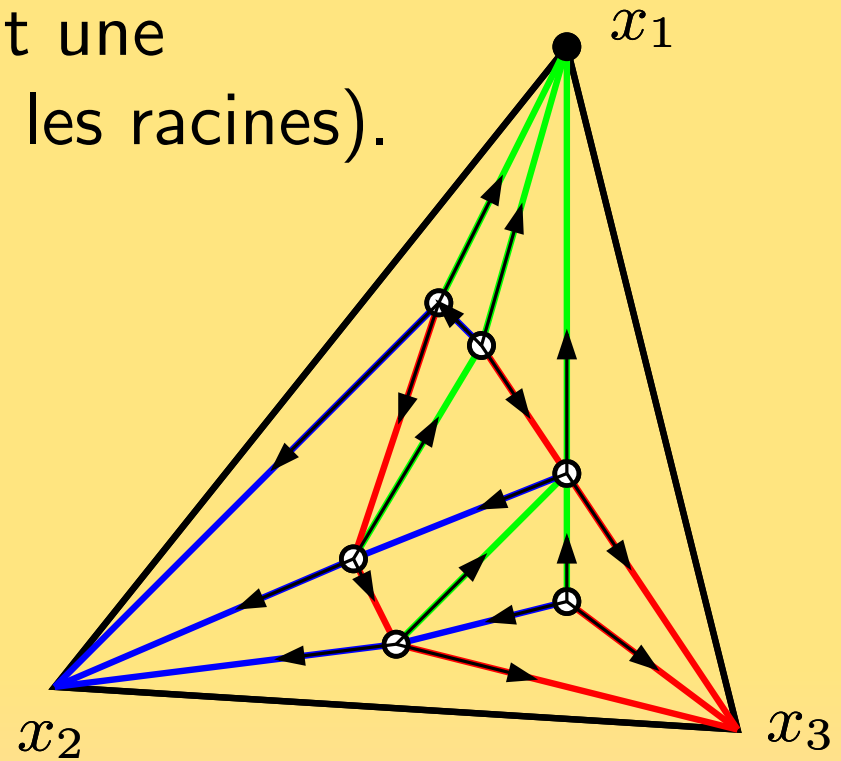
La partition en 3 arbres induit une 3-orientation des arêtes (vers les racines).



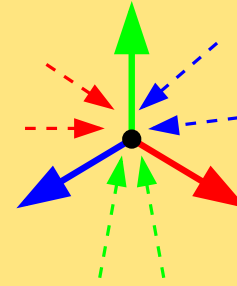
# 3-orientations d'une triangulation



La partition en 3 arbres induit une 3-orientation des arêtes (vers les racines).



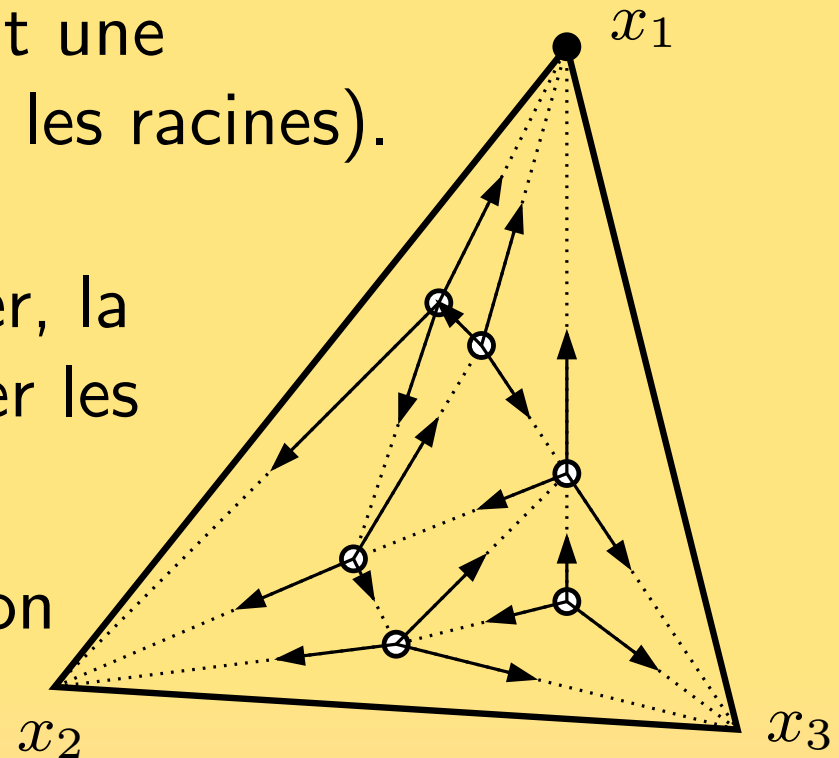
## 3-orientations d'une triangulation



La partition en 3 arbres induit une 3-orientation des arêtes (vers les racines).

Au vu de la règle de Schnyder, la 3-orientation suffit à retrouver les couleurs.

**Lemme.** L'algorithme glouton de coloriage ne donne jamais de contradiction.

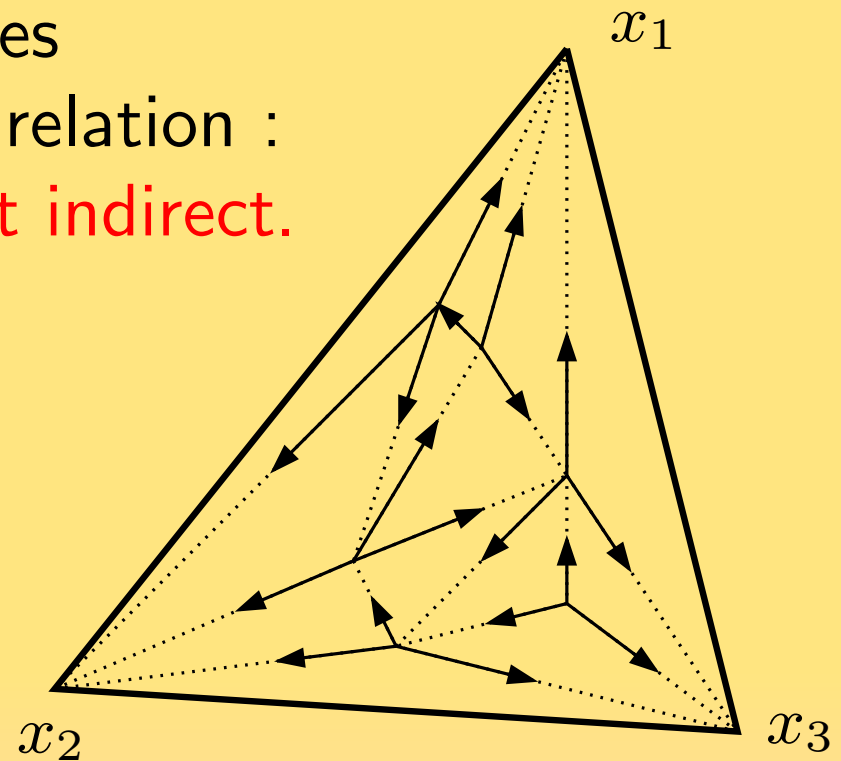


**Théorème.** Les 3-orientations sont en bijection avec les forêts de Schnyder.



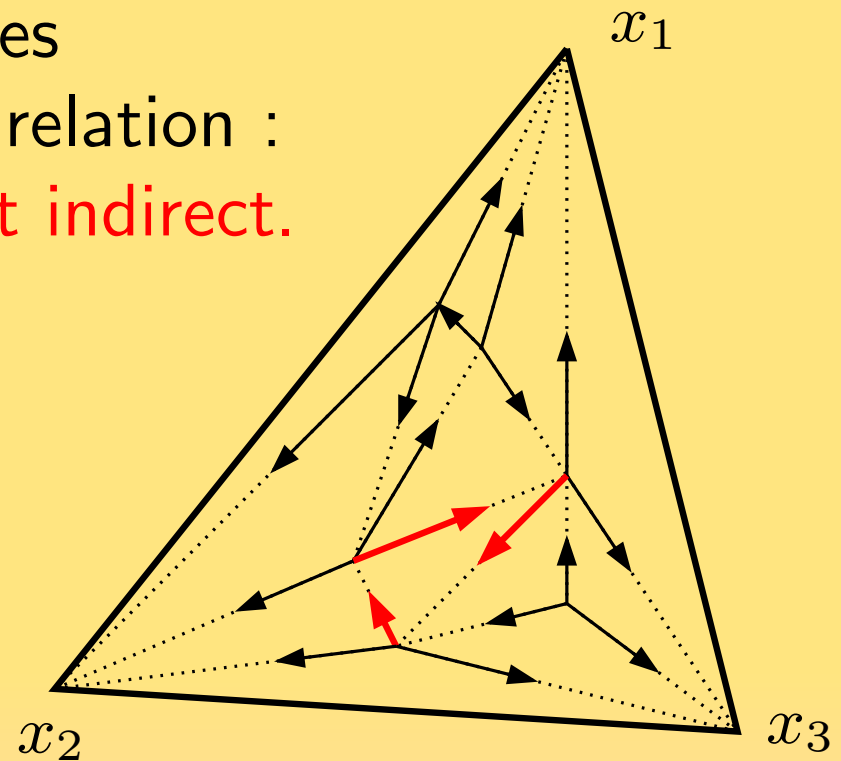
## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**



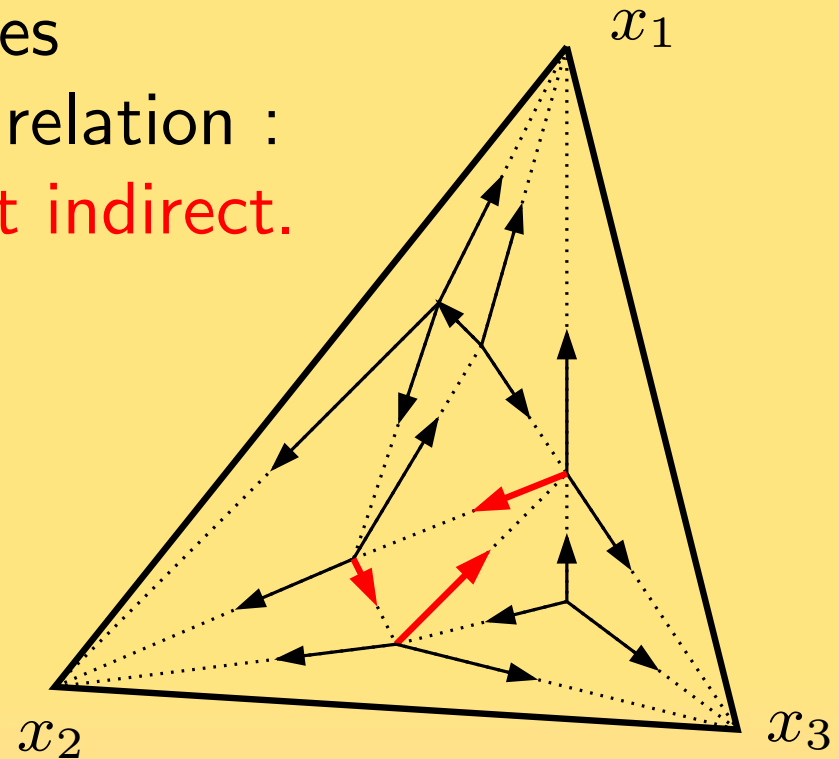
## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**



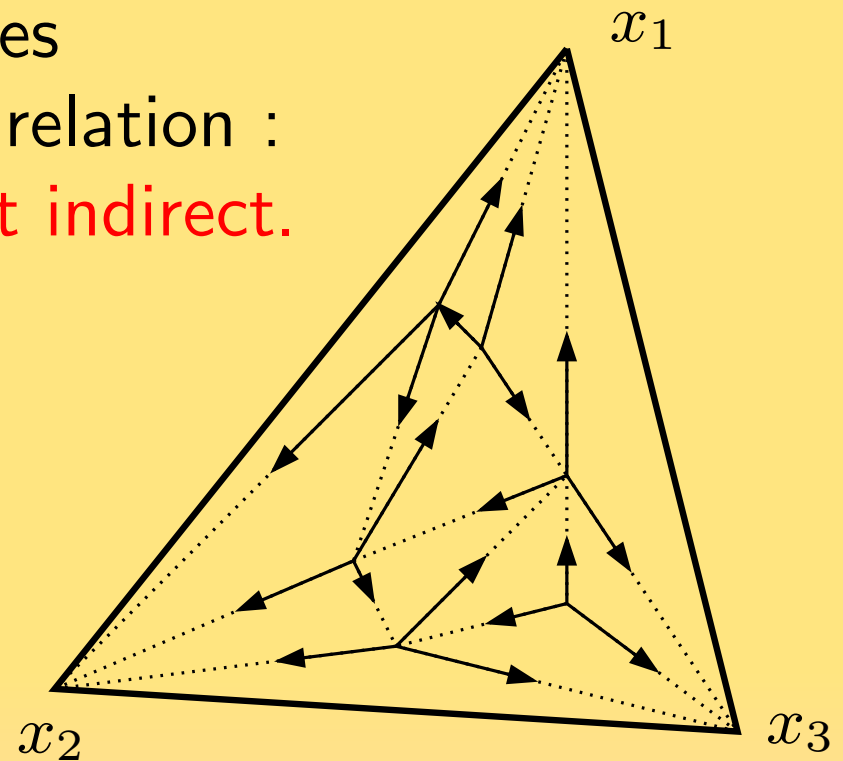
## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**



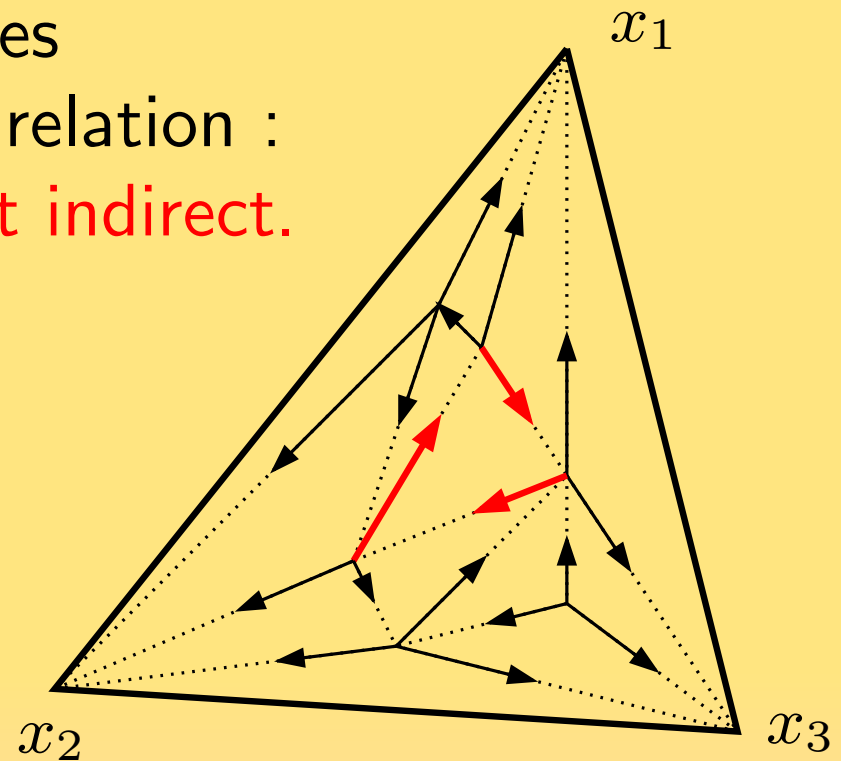
## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**



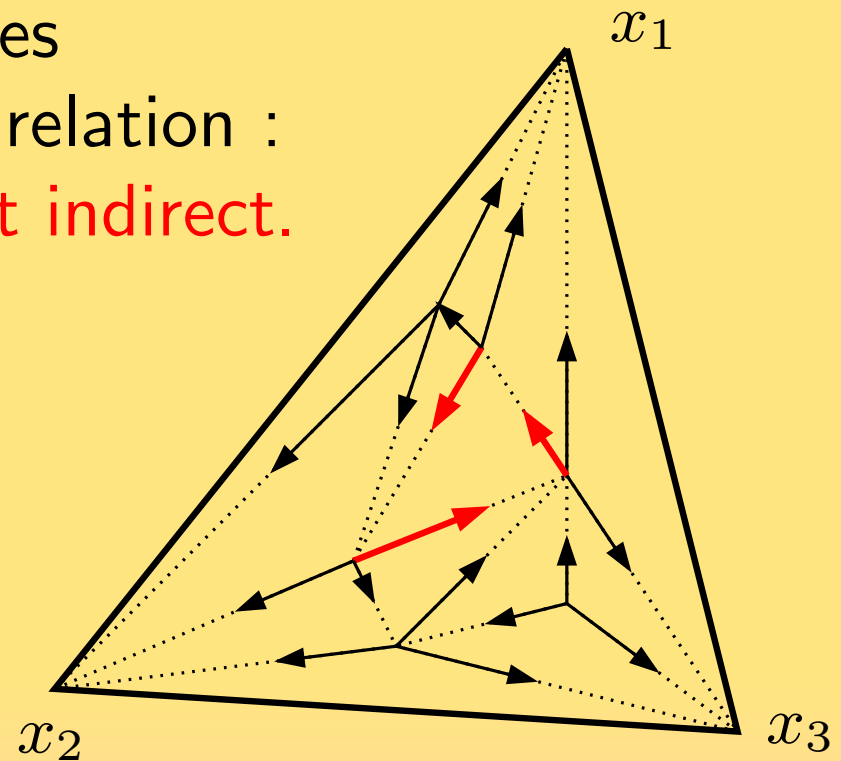
## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**



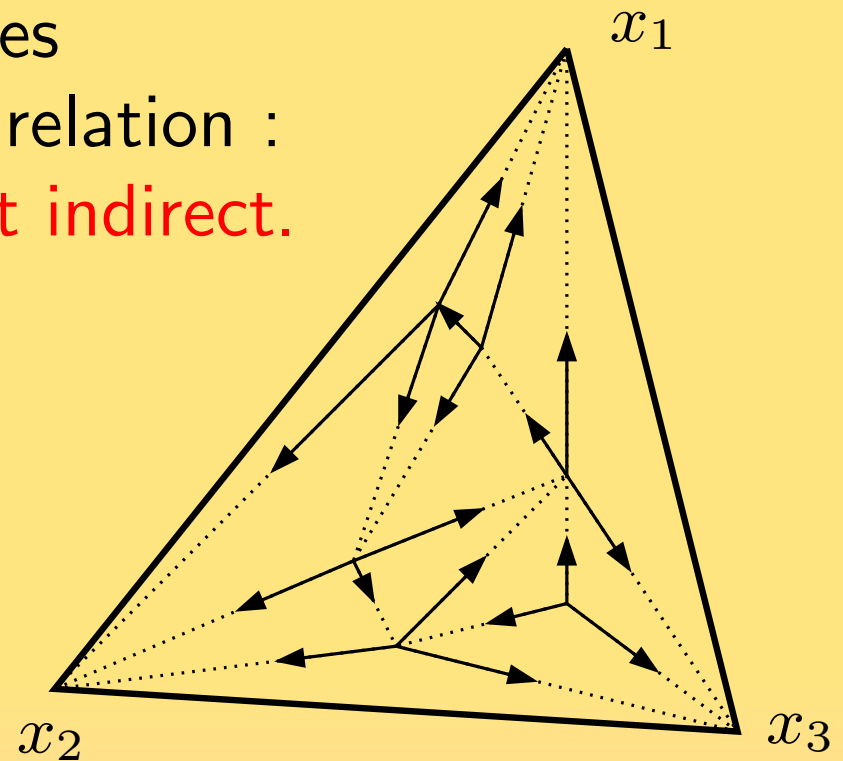
## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**



## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**

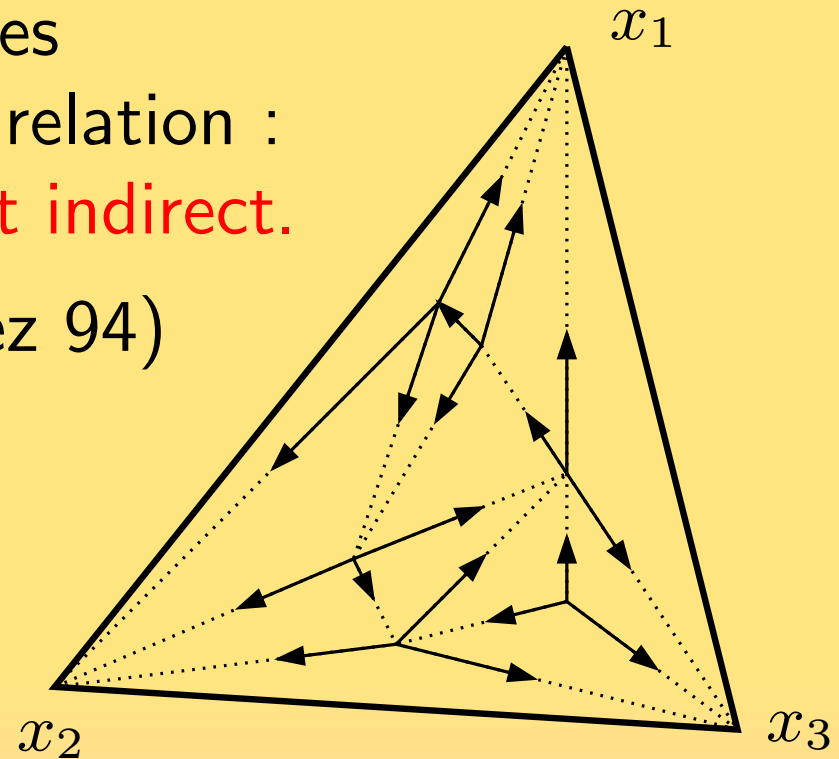


## 3-orientations d'une triangulation

On définit un ordre partiel sur les triangulations par clôture de la relation :  
**retourner les flèches d'un circuit indirect.**

**Théorème.** (Ossona de Mendez 94)

Le retournement de circuit définit un treillis, dont la relation de couverture est le retournement d'un triangle orienté indirect.



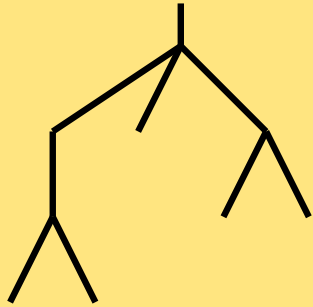
**Corollaire.** Il y a une unique orientation sans circuit indirect.



# Entropie et codage

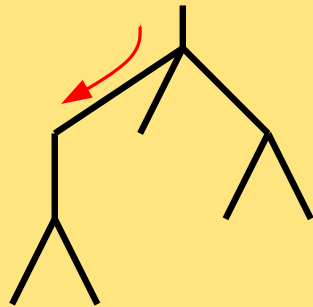
# Entropie et codage

Le code de contour d'un arbre plan (arbre ordonné).



# Entropie et codage

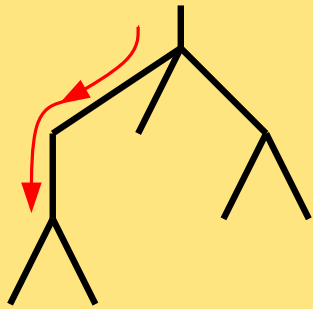
Le code de contour d'un arbre plan (arbre ordonné).



1

# Entropie et codage

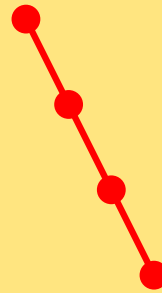
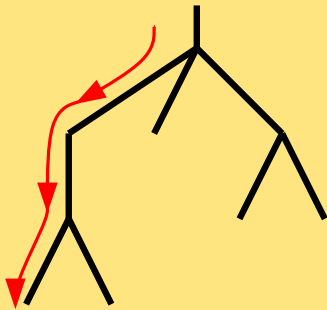
Le code de contour d'un arbre plan (arbre ordonné).



11

# Entropie et codage

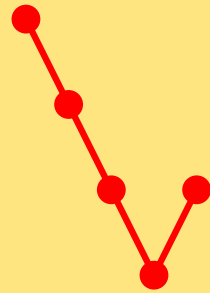
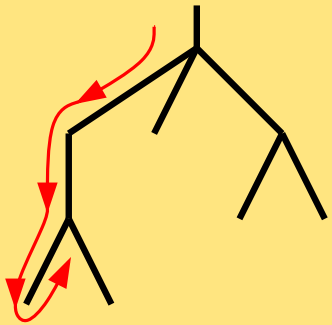
Le code de contour d'un arbre plan (arbre ordonné).



111

# Entropie et codage

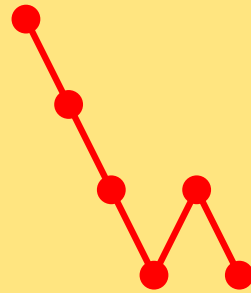
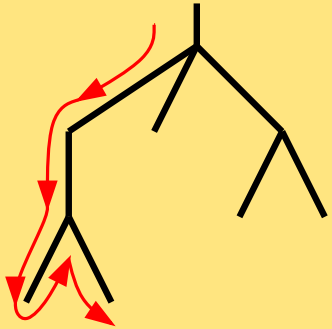
Le code de contour d'un arbre plan (arbre ordonné).



1110

# Entropie et codage

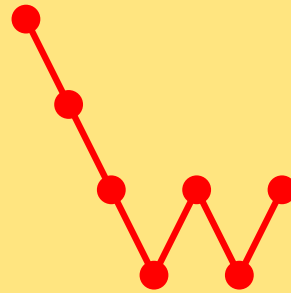
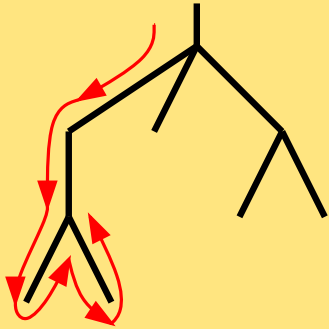
Le code de contour d'un arbre plan (arbre ordonné).



11101

# Entropie et codage

Le code de contour d'un arbre plan (arbre ordonné).

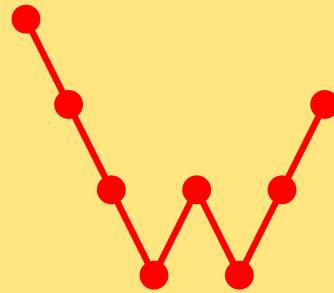
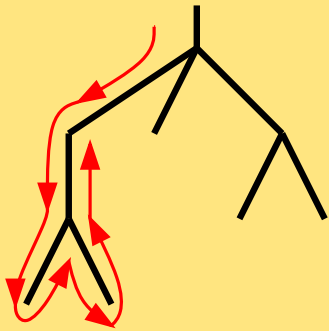


111010



# Entropie et codage

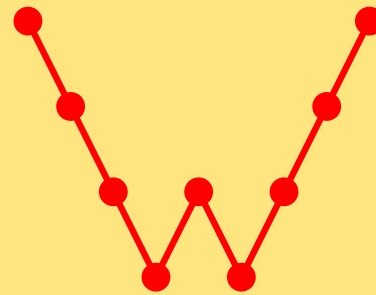
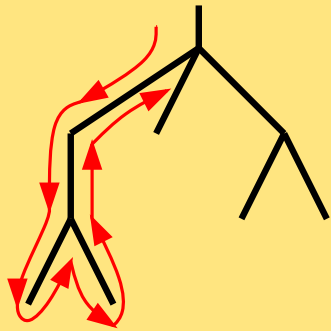
Le code de contour d'un arbre plan (arbre ordonné).



1110100

# Entropie et codage

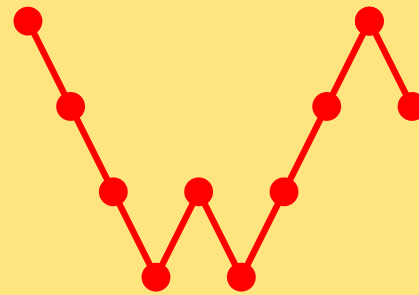
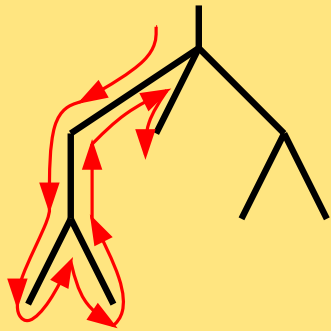
Le code de contour d'un arbre plan (arbre ordonné).



11101000

# Entropie et codage

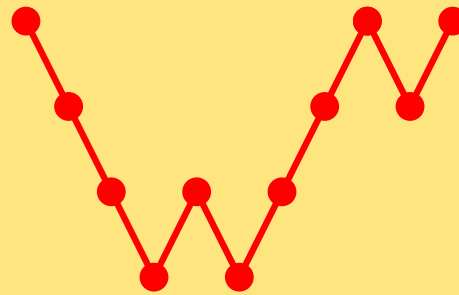
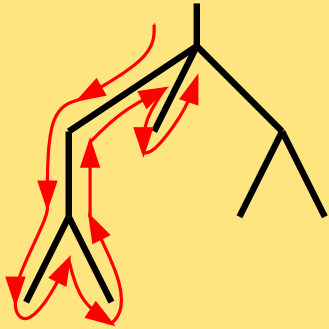
Le code de contour d'un arbre plan (arbre ordonné).



111010001

# Entropie et codage

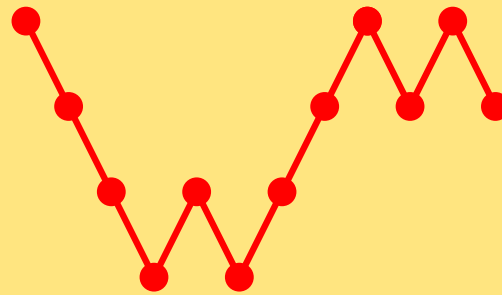
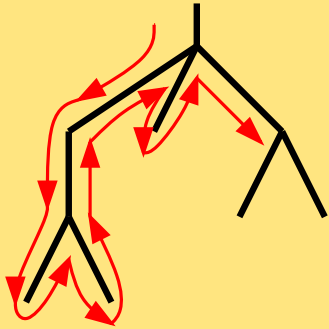
Le code de contour d'un arbre plan (arbre ordonné).



1110100010

# Entropie et codage

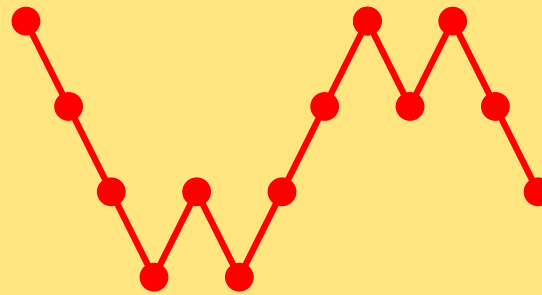
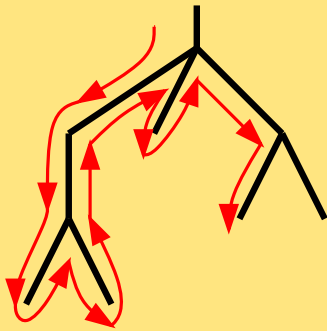
Le code de contour d'un arbre plan (arbre ordonné).



11101000101

# Entropie et codage

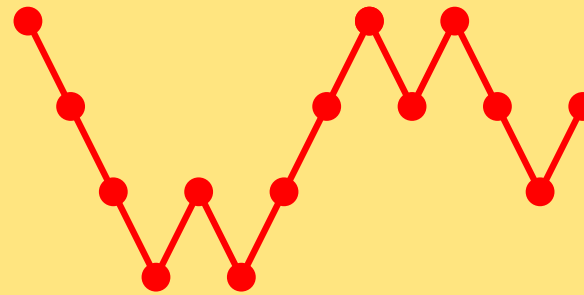
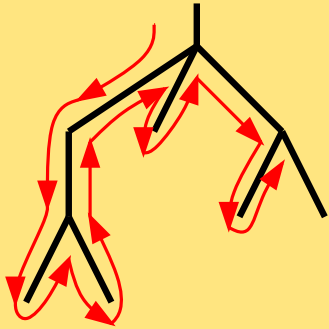
Le code de contour d'un arbre plan (arbre ordonné).



111010001011

# Entropie et codage

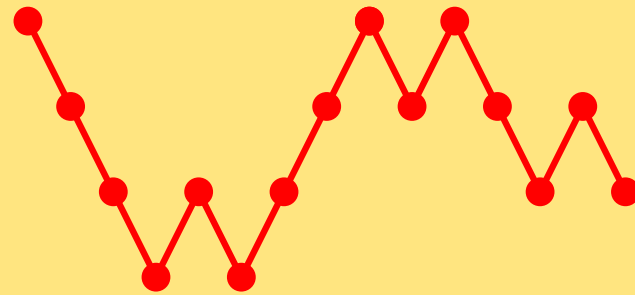
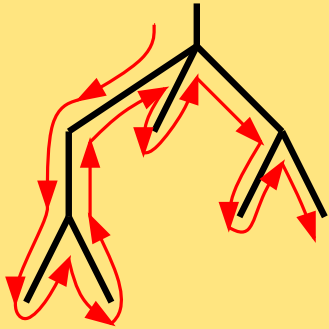
Le code de contour d'un arbre plan (arbre ordonné).



1110100010110

# Entropie et codage

Le code de contour d'un arbre plan (arbre ordonné).



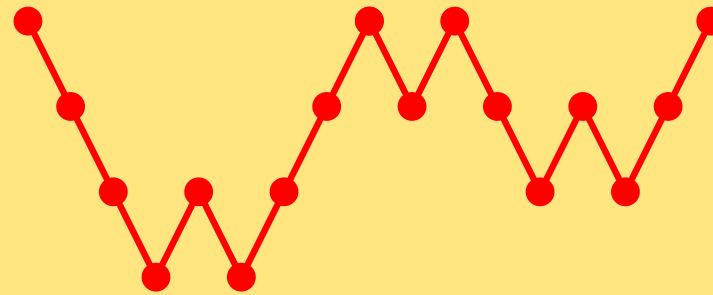
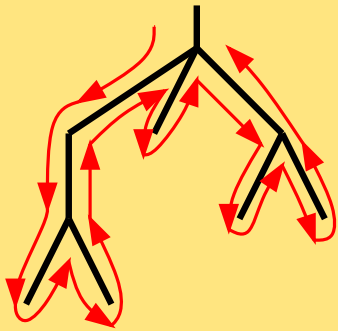
11101000101101





# Entropie et codage

Le code de contour d'un arbre plan (arbre ordonné).

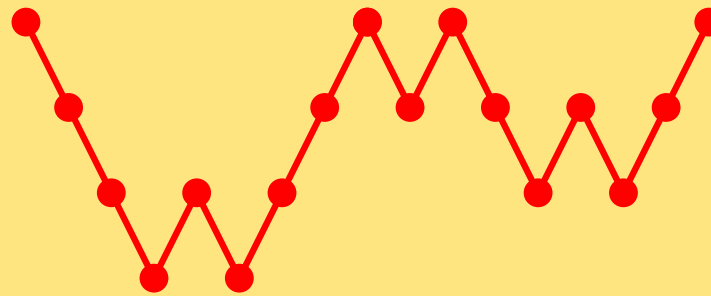
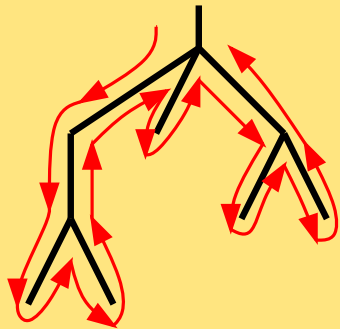


1110100010110100

$\Rightarrow 2n$  bits pour coder un arbre avec  $n$  arêtes.

# Entropie et codage

Le code de contour d'un arbre plan (arbre ordonné).



1110100010110100

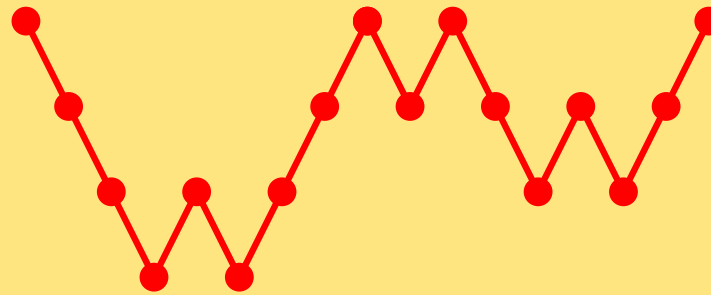
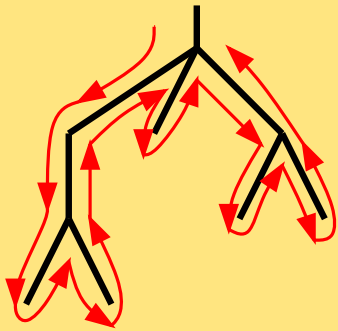
$\Rightarrow 2n$  bits pour coder un arbre avec  $n$  arêtes.

Or le nombre d'arbres avec  $n$  arêtes est  $|\mathcal{A}_n| = \frac{1}{n+1} \binom{2n}{n} \asymp 4^n$ .

$\Rightarrow$  il faut au moins  $\log_2 |\mathcal{A}_n| = 2n + o(n)$  bits

# Entropie et codage

Le code de contour d'un arbre plan (arbre ordonné).



1110100010110100

$\Rightarrow 2n$  bits pour coder un arbre avec  $n$  arêtes.

Or le nombre d'arbres avec  $n$  arêtes est  $|\mathcal{A}_n| = \frac{1}{n+1} \binom{2n}{n} \asymp 4^n$ .

$\Rightarrow$  il faut au moins  $\log_2 |\mathcal{A}_n| = 2n + o(n)$  bits

Ce codage est "asymptotiquement optimal".

**Définition.** L'entropie d'une classe de structure  $\mathcal{C}_n$  est

$$||\mathcal{C}_n|| = \log_2(1 + |\mathcal{C}_n|).$$

**Lemme.** Tout codage pour les objets de  $\mathcal{C}_n$  utilise au moins un mot de taille  $||\mathcal{C}_n||$ .

**Définition.** Un codage qui utilise asymptotiquement  $||\mathcal{C}_n|| \cdot (1 + o(1))$  bits est dit asymptotiquement optimal.

**Définition.** L'entropie d'une classe de structure  $\mathcal{C}_n$  est

$$||\mathcal{C}_n|| = \log_2(1 + |\mathcal{C}_n|).$$

**Lemme.** Tout codage pour les objets de  $\mathcal{C}_n$  utilise au moins un mot de taille  $||\mathcal{C}_n||$ .

**Définition.** Un codage qui utilise asymptotiquement  $||\mathcal{C}_n|| \cdot (1 + o(1))$  bits est dit asymptotiquement optimal.

Un codage peut utiliser moins de  $||\mathcal{C}_n||$  bits pour coder certains objets de  $\mathcal{C}_n$ .

Un codeur amélioré pour les arbres à 2 feuilles par nœuds.

**Idée:** Chaque feuille donne le code 10



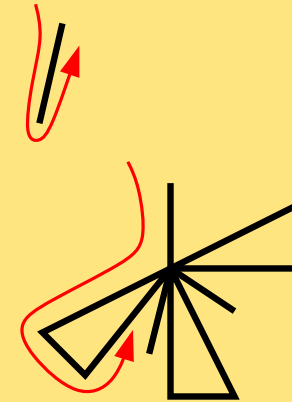
Un codeur amélioré pour les arbres à 2 feuilles par nœuds.

**Idée:** Chaque feuille donne le code 10

Le codage d'un nœud est donc:

$\dots 1w_110w_210w_30\dots$

avec  $w_i$  code bien formé du sous-arbre.





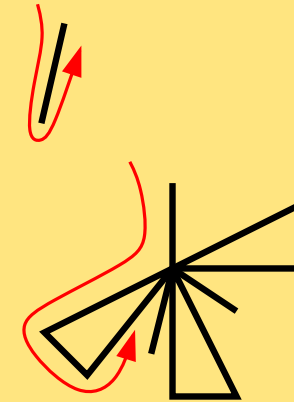
Un codeur amélioré pour les arbres à 2 feuilles par nœuds.

**Idée:** Chaque feuille donne le code 10

Le codage d'un nœud est donc:

$\dots 1w_1 10w_2 10w_3 0 \dots$

avec  $w_i$  code bien formé du sous-arbre.



On simplifie le codage en

$\dots 1w_1 0w_2 0w_3 0 \dots$

**Résultat.** Un arbre avec  $n$  nœuds et  $2n$  feuilles est codé par un mot de longueur  $4n$  avec  $n$  bit 1, au lieu de  $6n$  avec  $3n$  bit 1.

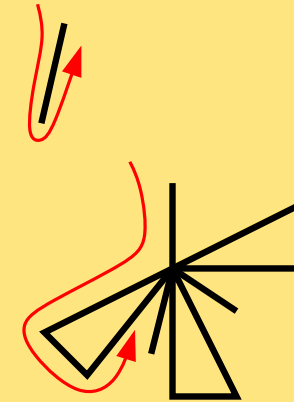
Un codeur amélioré pour les arbres à 2 feuilles par nœuds.

**Idée:** Chaque feuille donne le code 10

Le codage d'un nœud est donc:

$$\dots 1w_110w_210w_30\dots$$

avec  $w_i$  code bien formé du sous-arbre.



On simplifie le codage en

$$\dots 1w_10w_20w_30\dots$$

**Résultat.** Un arbre avec  $n$  nœuds et  $2n$  feuilles est codé par un mot de longueur  $4n$  avec  $n$  bit 1, au lieu de  $6n$  avec  $3n$  bit 1.

Plus précisément le nombre d'arbres ayant  $n$  nœuds portant chacun 2 feuilles est  $\frac{2}{n-1} \binom{4n-3}{n-2}$ .

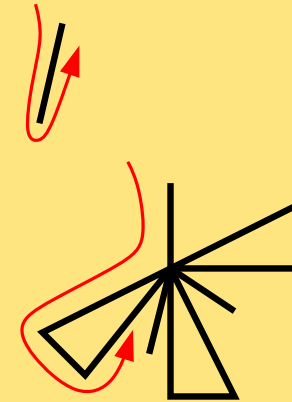
Un codeur amélioré pour les arbres à 2 feuilles par nœuds.

**Idée:** Chaque feuille donne le code 10

Le codage d'un nœud est donc:

$$\dots 1w_1 10w_2 10w_3 0 \dots$$

avec  $w_i$  code bien formé du sous-arbre.



On simplifie le codage en

$$\dots 1w_1 0w_2 0w_3 0 \dots$$

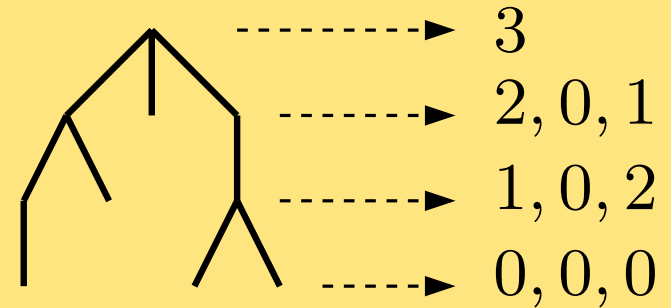
**Résultat.** Un arbre avec  $n$  nœuds et  $2n$  feuilles est codé par un mot de longueur  $4n$  avec  $n$  bit 1, au lieu de  $6n$  avec  $3n$  bit 1.

Après codage entropique le code est optimal pour les arbres à 2 feuilles par nœuds: en effet  $||\mathcal{A}_n^{(2)}|| = (\log_2 \frac{256}{27})n + o(n)$ .

Contrôle des degrés des sommets...

Le code par degré pour les arbres:

Lister les degrés des sommets par un parcours en largeur (ou autre).

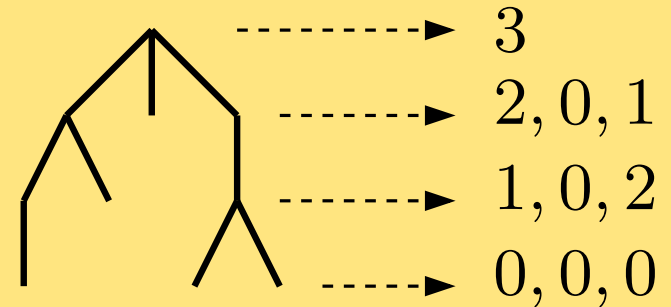


3,2,0,1,1,0,2,0,0,0

Contrôle des degrés des sommets...

Le code par degré pour les arbres:

Lister les degrés des sommets par un parcours en largeur (ou autre).



3,2,0,1,1,0,2,0,0,0

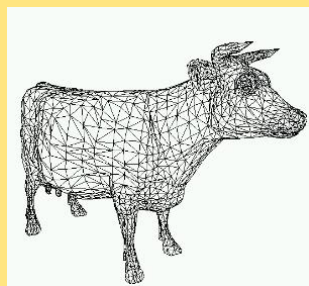
**Théorème.** Associé au codage entropique, le code par degré est optimal pour les arbres de fréquence des degrés fixées.

**Exemple.**

Les arbres binaires:  $n \times 2$  et  $(n + 1) \times 0 \Rightarrow 1$  bits/sommets.

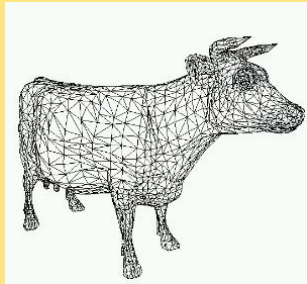
Les arbres ternaires:  $n \times 3$  et  $(2n + 1) \times 0 \Rightarrow \log_2 \frac{27}{4}$  bps.

# Codage et compression des maillages

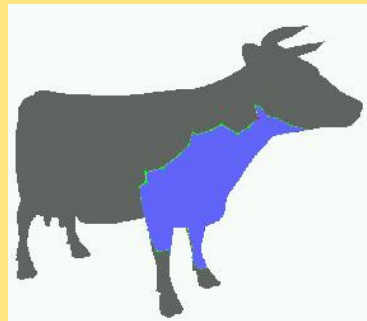


## Codage et compression des maillages

La majorité des codeurs de maillages de la littérature fonctionnent par conquête à partir d'une zone connue.

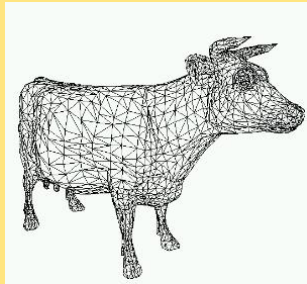


(la plus belle conquête de l'homme selon O.D.)

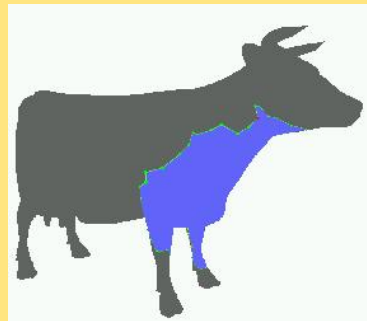


## Codage et compression des maillages

La majorité des codeurs de maillages de la littérature fonctionnent par conquête à partir d'une zone connue.



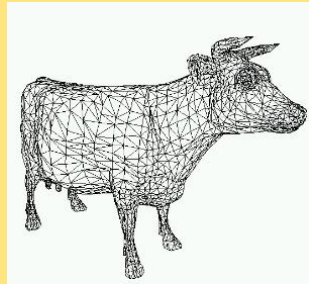
(la plus belle conquête de l'homme selon O.D.)



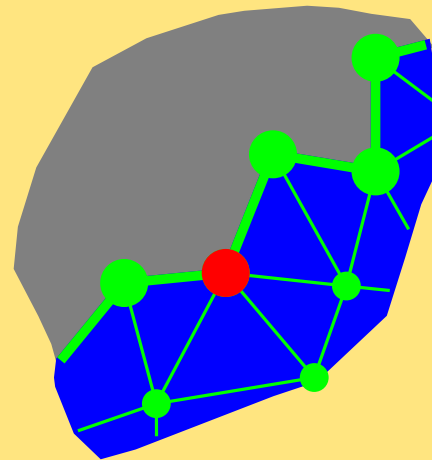
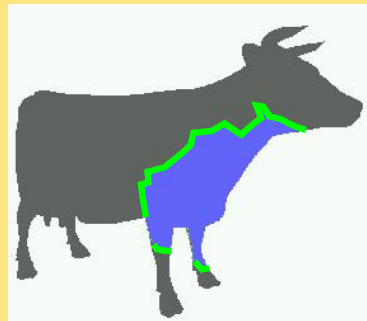


## Codage et compression des maillages

La majorité des codeurs de maillages de la littérature fonctionnent par conquête à partir d'une zone connue.



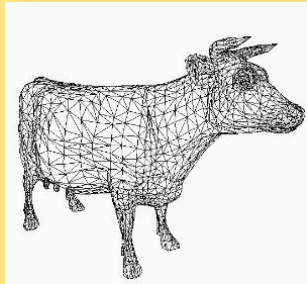
(la plus belle conquête de l'homme selon O.D.)



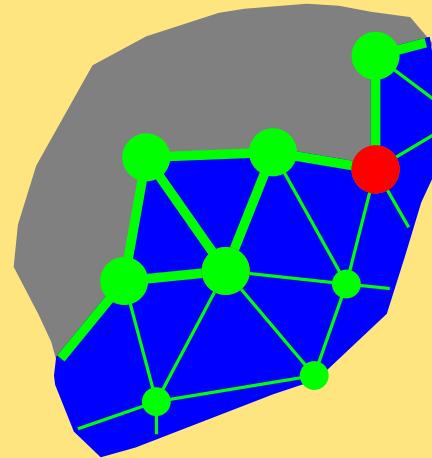
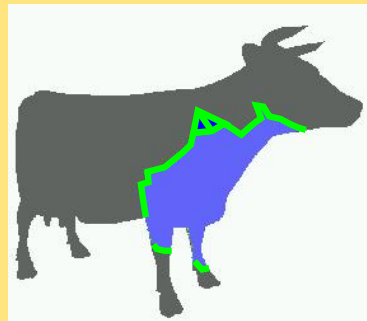
point base au bord

## Codage et compression des maillages

La majorité des codeurs de maillages de la littérature fonctionnent par conquête à partir d'une zone connue.



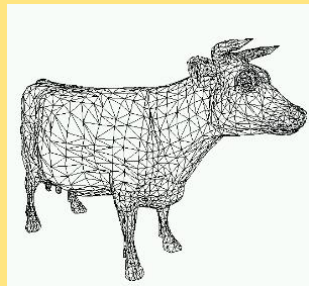
(la plus belle conquête de l'homme selon O.D.)



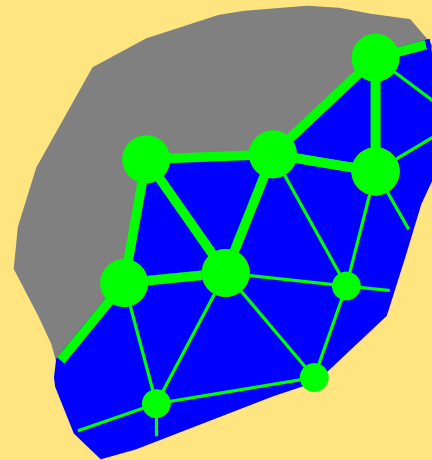
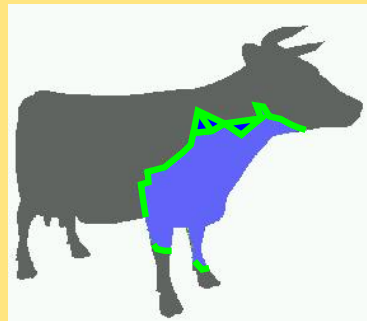
point base au bord  
conquete sommet

## Codage et compression des maillages

La majorité des codeurs de maillages de la littérature fonctionnent par conquête à partir d'une zone connue.



(la plus belle conquête de l'homme selon O.D.)



point base au bord  
conquete sommet  
conquête face

Chaque étape de la conquête donne lieu à l'écriture d'un symbole.  
Le choix du nouveau point base ne dépend que de la partie déjà codée de façon à permettre le décodage.

⇒ Edgebreaker, Touma-Gotsman

Edgebreaker ou Touma-Gotsman sont très efficaces en pratique.

typiquement 2 bits par sommets

Edgebreaker ou Touma-Gotsman sont très efficaces en pratique.

typiquement 2 bits par sommets

Pourtant...

**Théorème.** (Tutte 72)

Le nombre de triangulations planaires à  $n + 2$  sommets est

$$\frac{2(4n-3)!}{(3n-1)!n!} \asymp \left(\frac{256}{27}\right)^n$$

⇒ une entropie de  $\log_2 \frac{256}{27} \approx 3.25$  bps.

Edgebreaker ou Touma-Gotsman sont très efficaces en pratique.

typiquement 2 bits par sommets

Pourtant...

**Théorème.** (Tutte 72)

Le nombre de triangulations planaires à  $n + 2$  sommets est

$$\frac{2(4n-3)!}{(3n-1)!n!} \asymp \left(\frac{256}{27}\right)^n$$

$\Rightarrow$  une entropie de  $\log_2 \frac{256}{27} \approx 3.25$  bps.

Les exemples utilisés habituellement pour tester Edgebreaker ou TG sont pris dans une sous-classe d'entropie inférieure.

Edgebreaker ou Touma-Gotsman sont très efficaces en pratique.

typiquement 2 bits par sommets

Pourtant...

**Théorème.** (Tutte 72)

Le nombre de triangulations planaires à  $n + 2$  sommets est

$$\frac{2(4n-3)!}{(3n-1)!n!} \asymp \left(\frac{256}{27}\right)^n$$

⇒ une entropie de  $\log_2 \frac{256}{27} \approx 3.25$  bps.

Les exemples utilisés habituellement pour tester Edgebreaker ou TG sont pris dans une sous-classe d'entropie inférieure.

Ces codeurs ne garantissent pas 3.25bps sur les triangulations "typiques" (=aléatoire).

Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

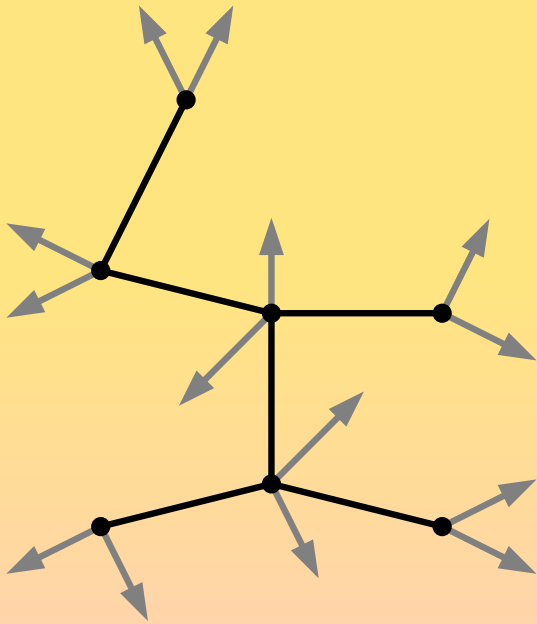


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

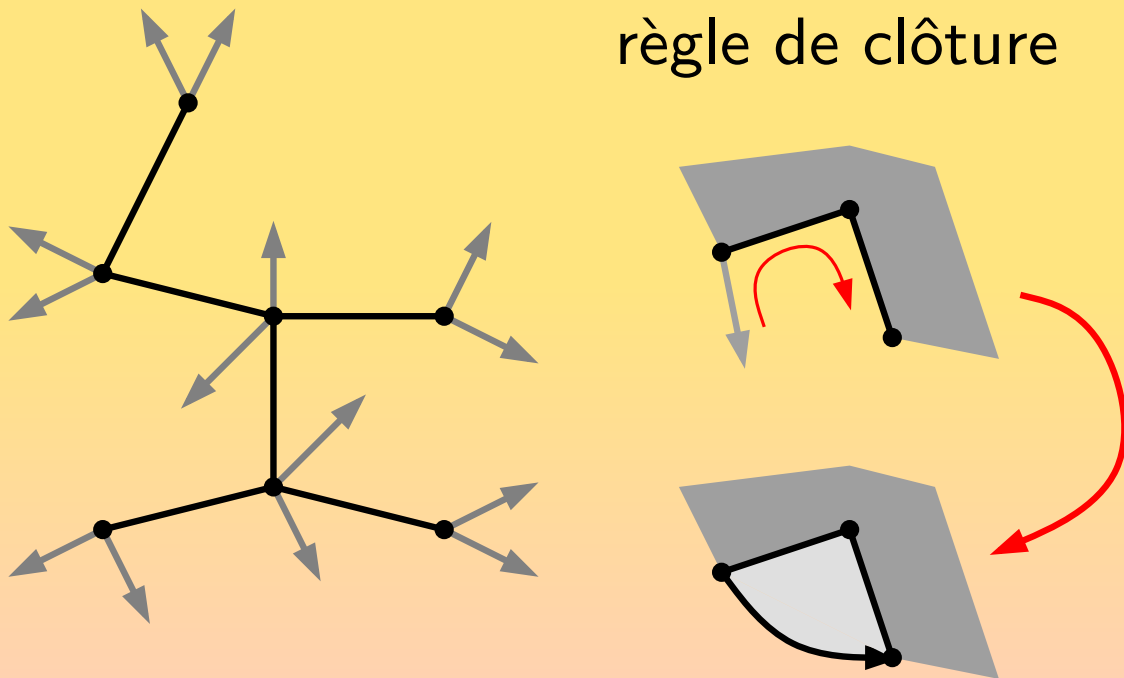


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

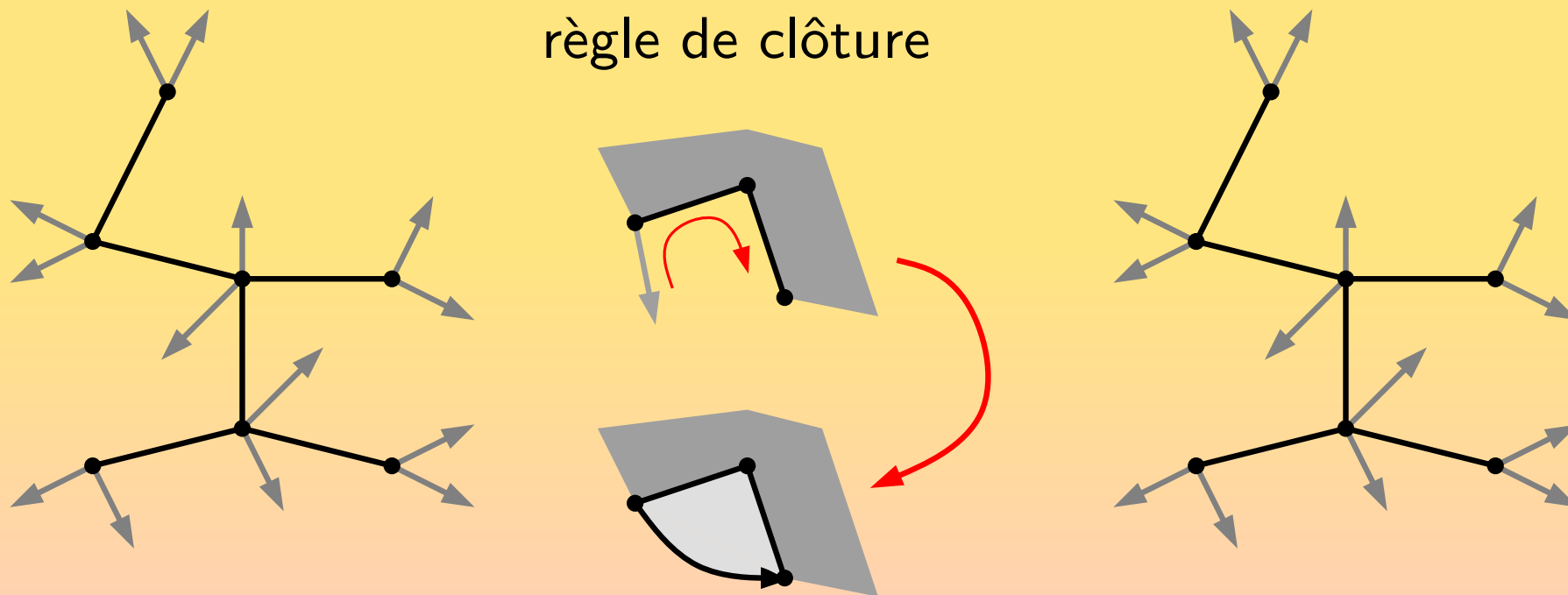


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

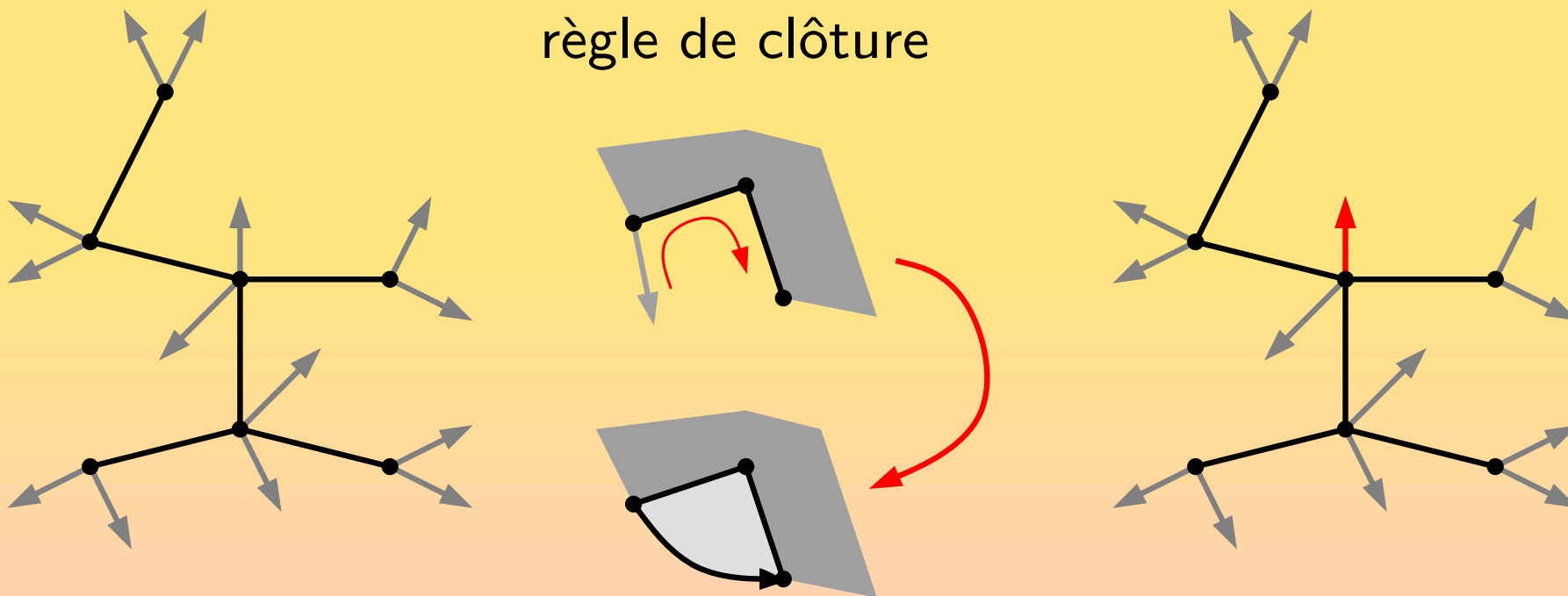


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

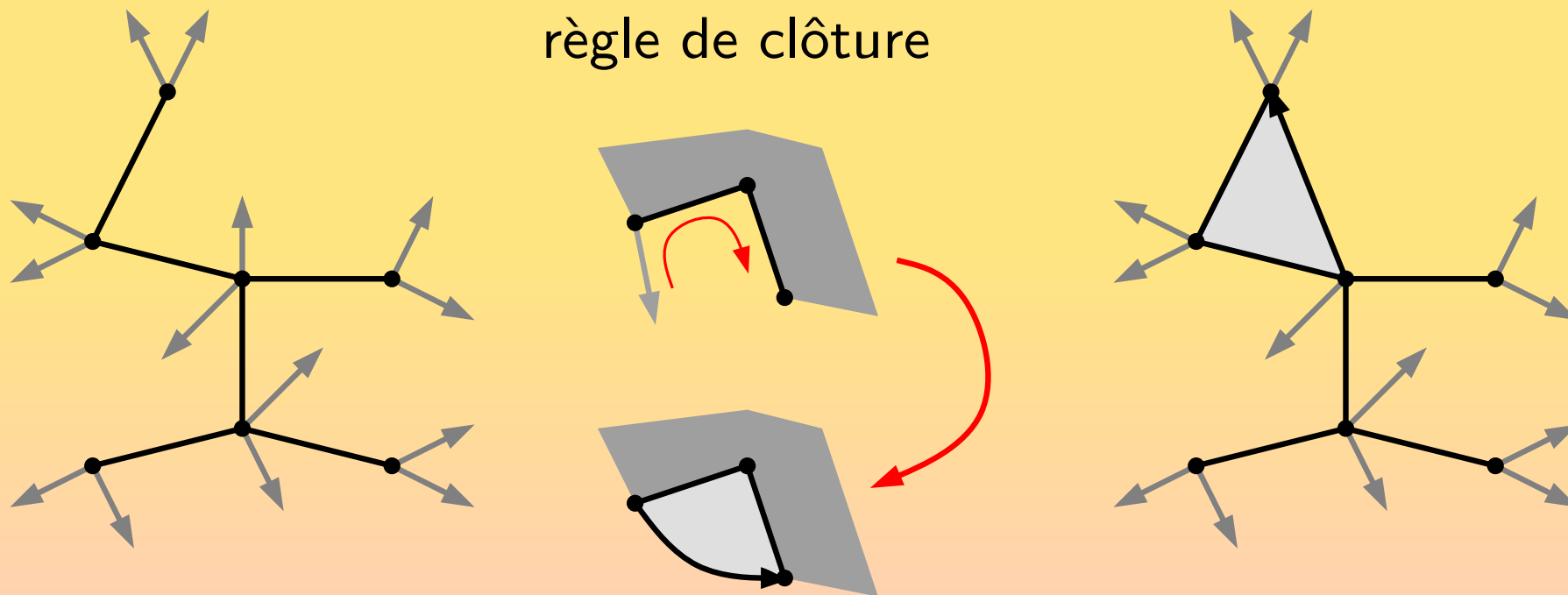


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

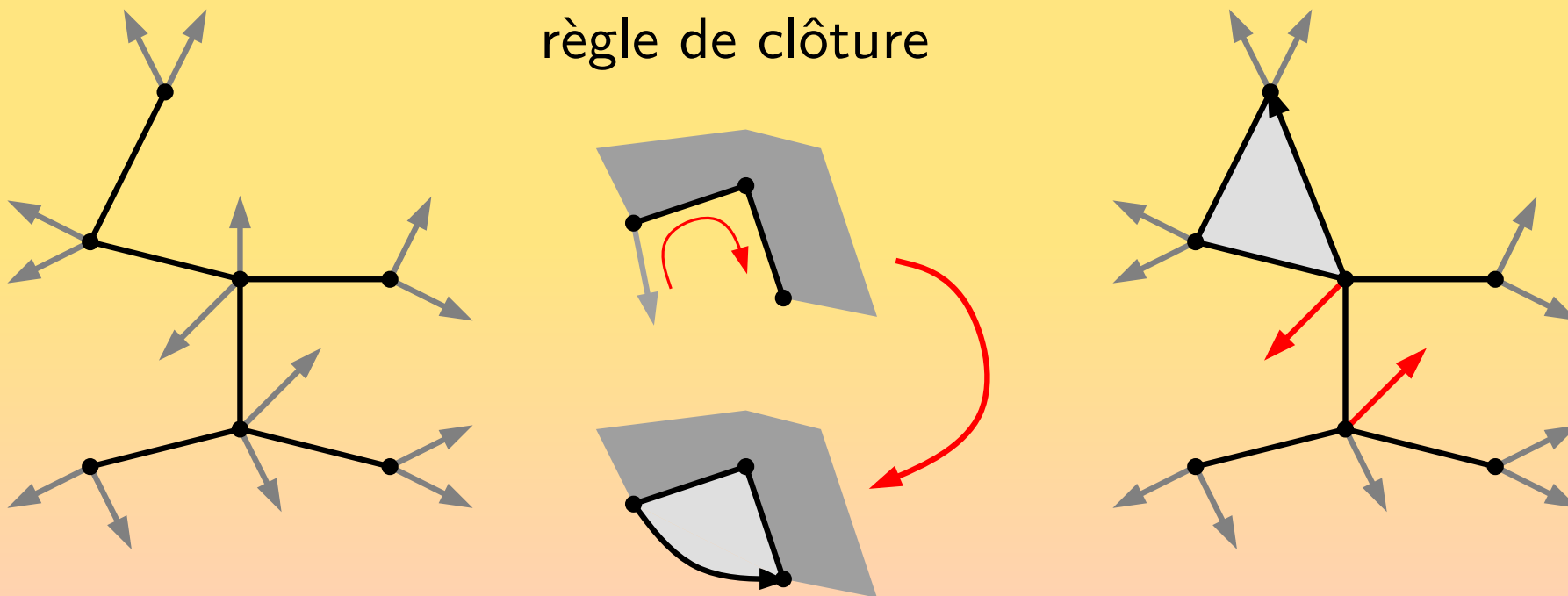


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

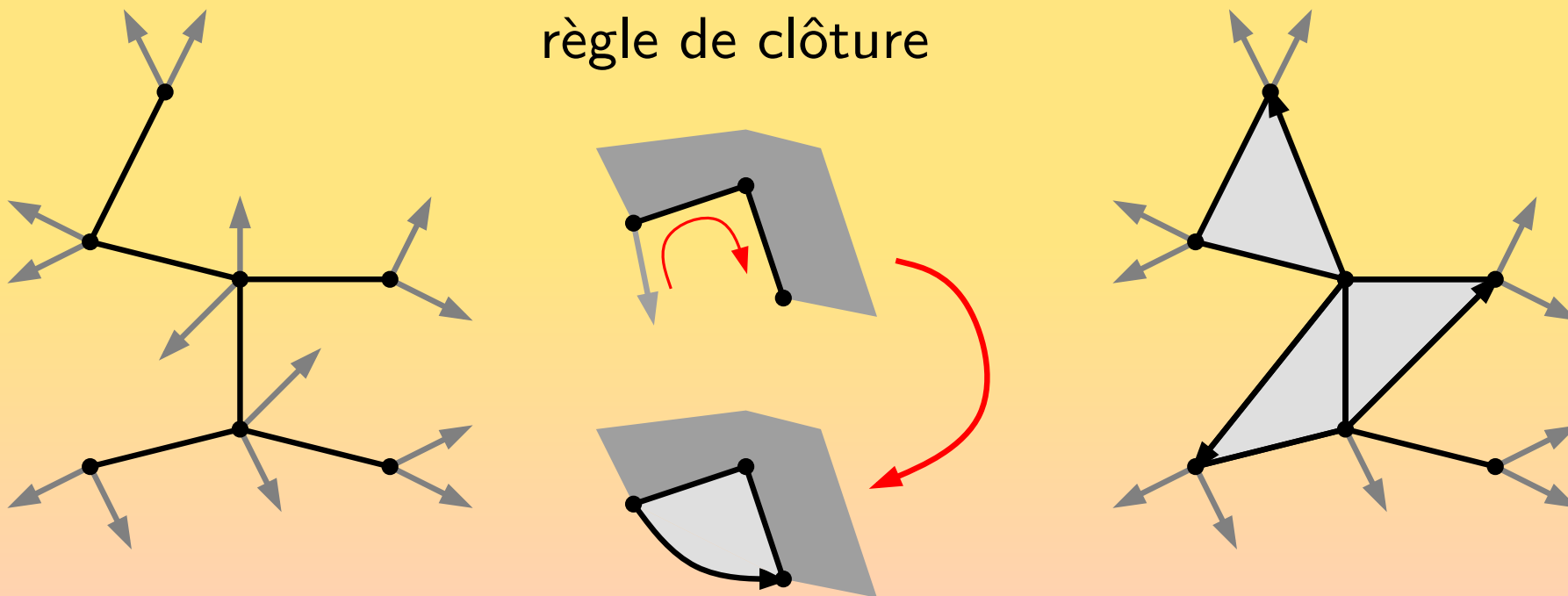


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

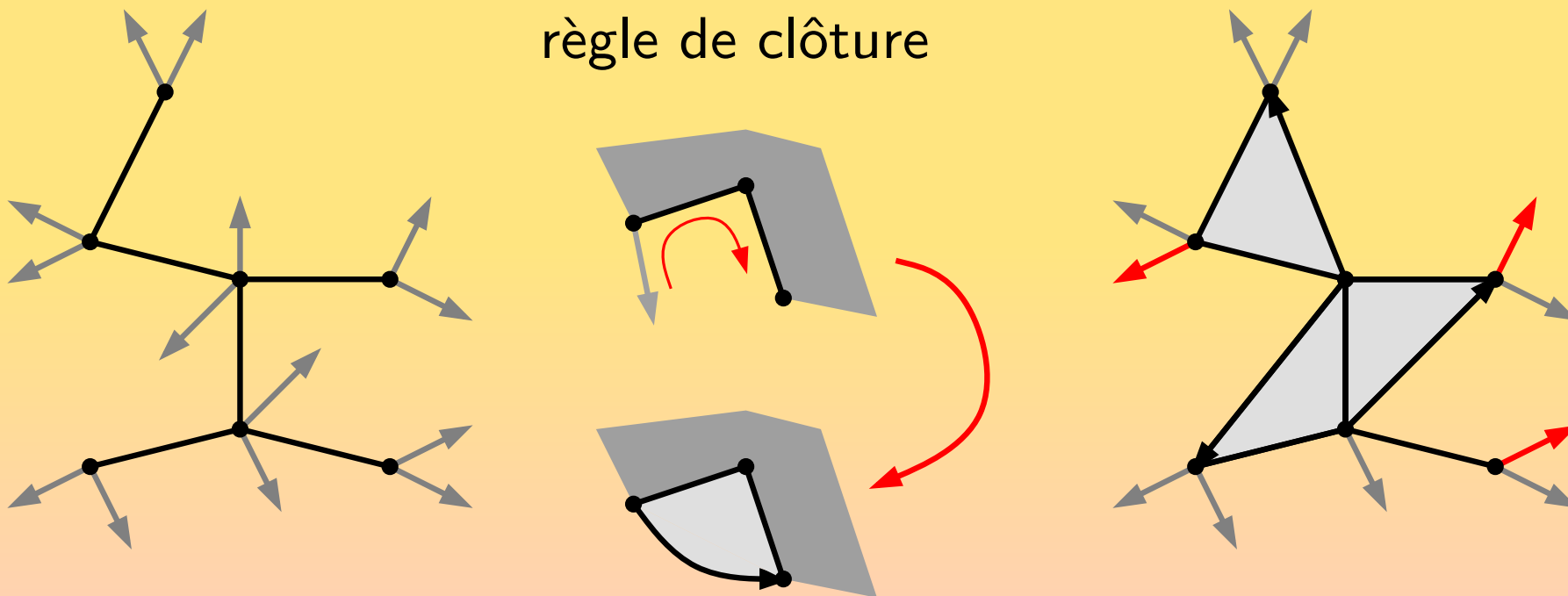


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet



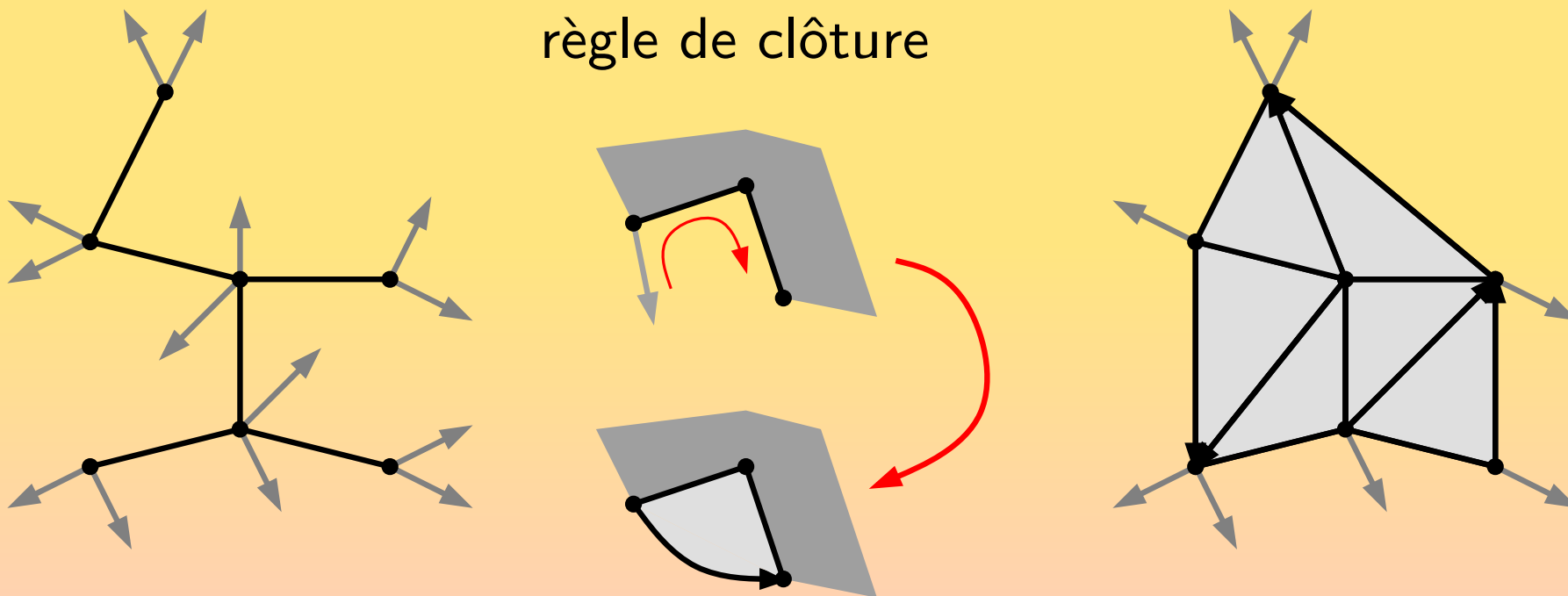


Pour garantir l'optimalité, on s'inspire de l'énumération :

Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet

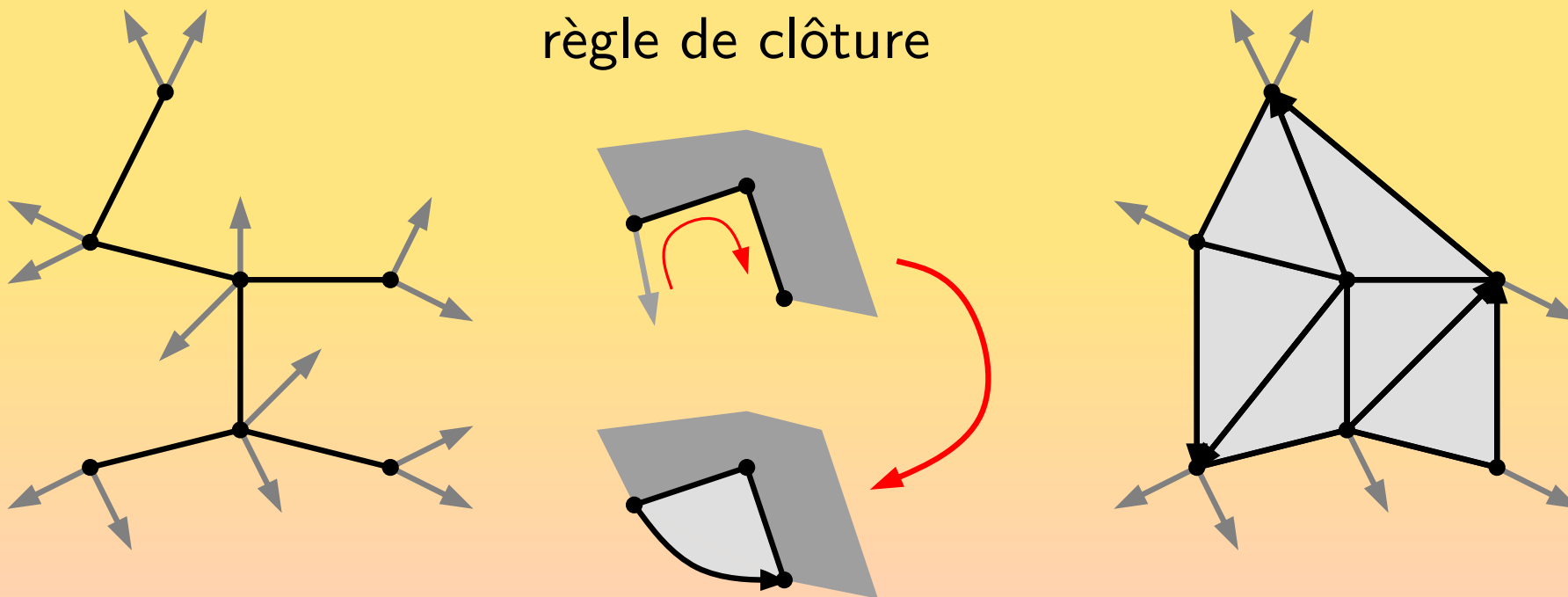


Pour garantir l'optimalité, on s'inspire de l'énumération :

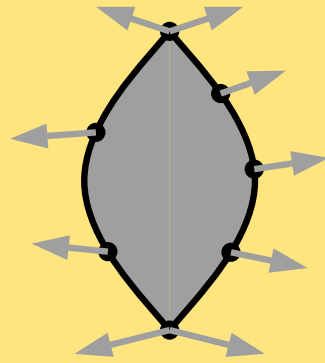
Le théorème de Tutte se réécrit:

$$|\mathcal{T}_n| = \frac{2}{2n} \cdot |\mathcal{A}_n^{(2)}|.$$

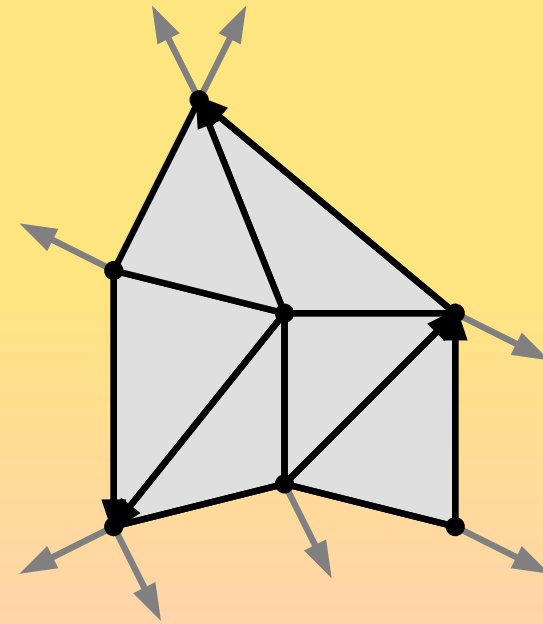
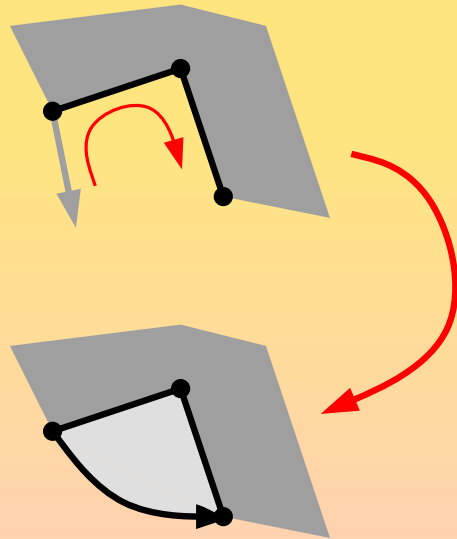
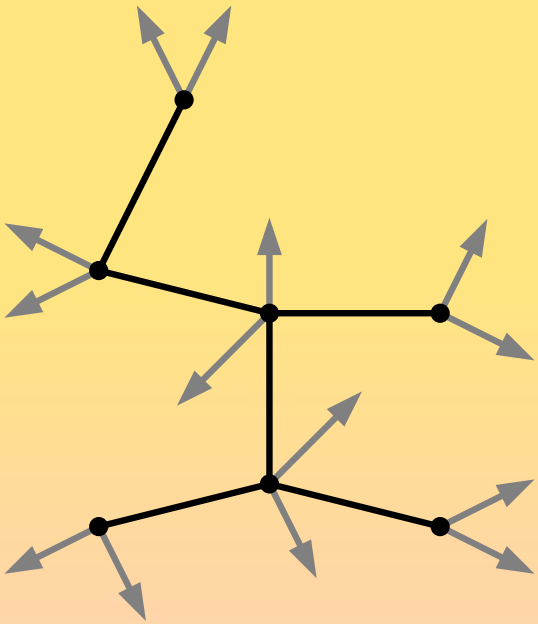
⇒ construire les triangulations à partir des arbres à 2 feuilles par sommet



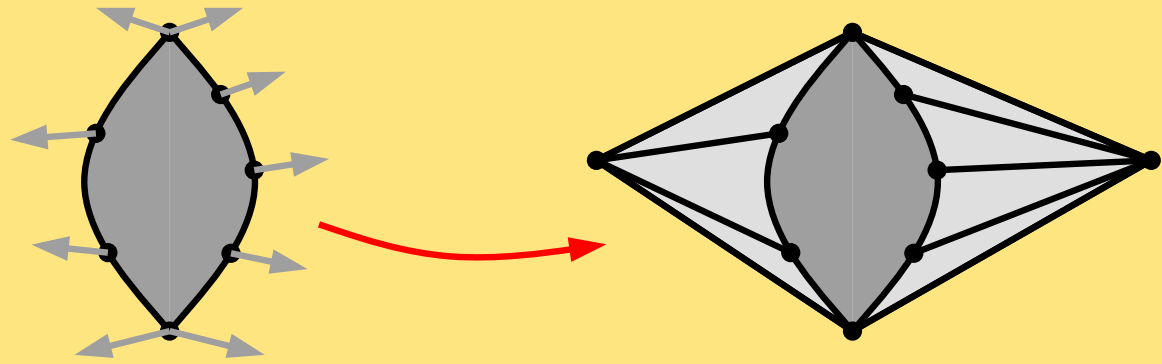
Lorsque la clôture s'arrête, il reste :



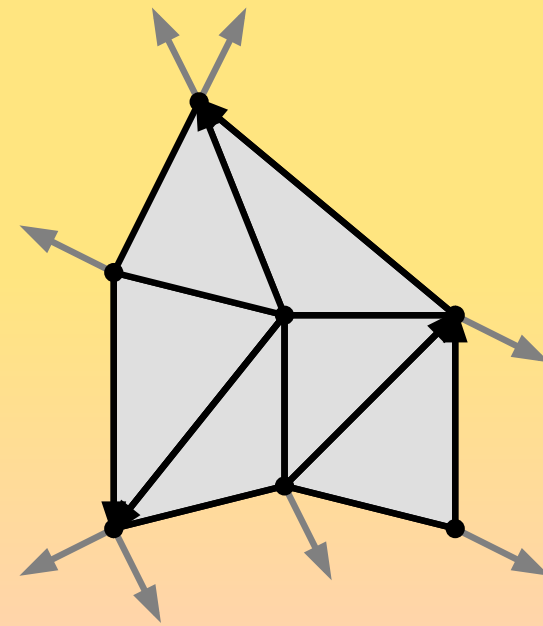
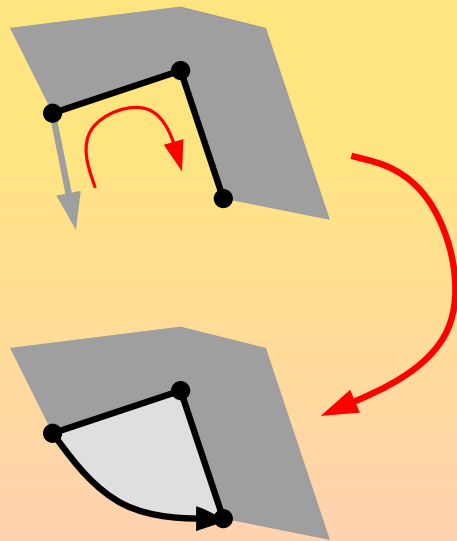
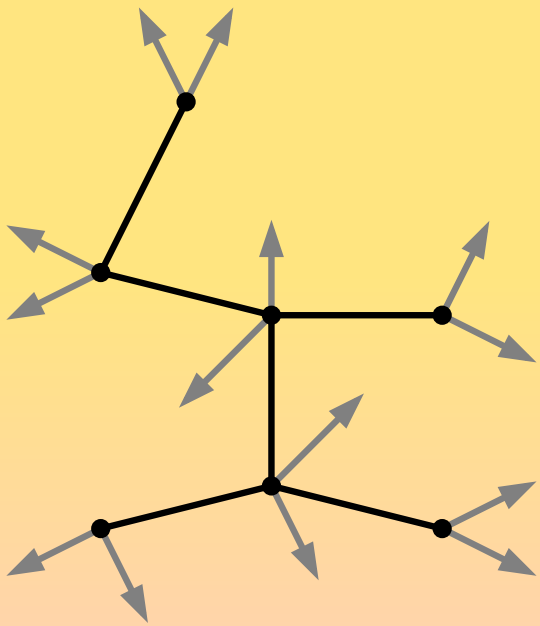
règle de clôture



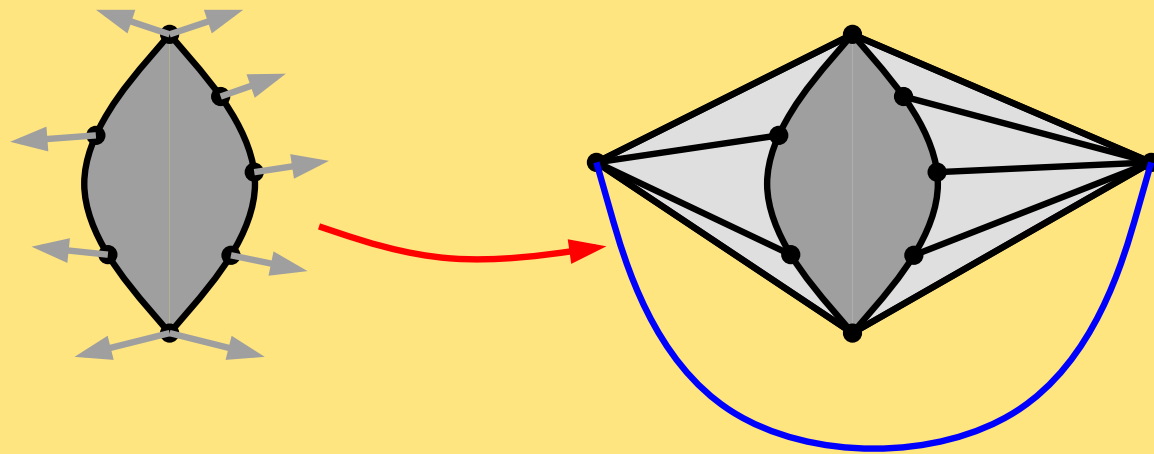
Lorsque la clôture s'arrête, il reste :



règle de clôture

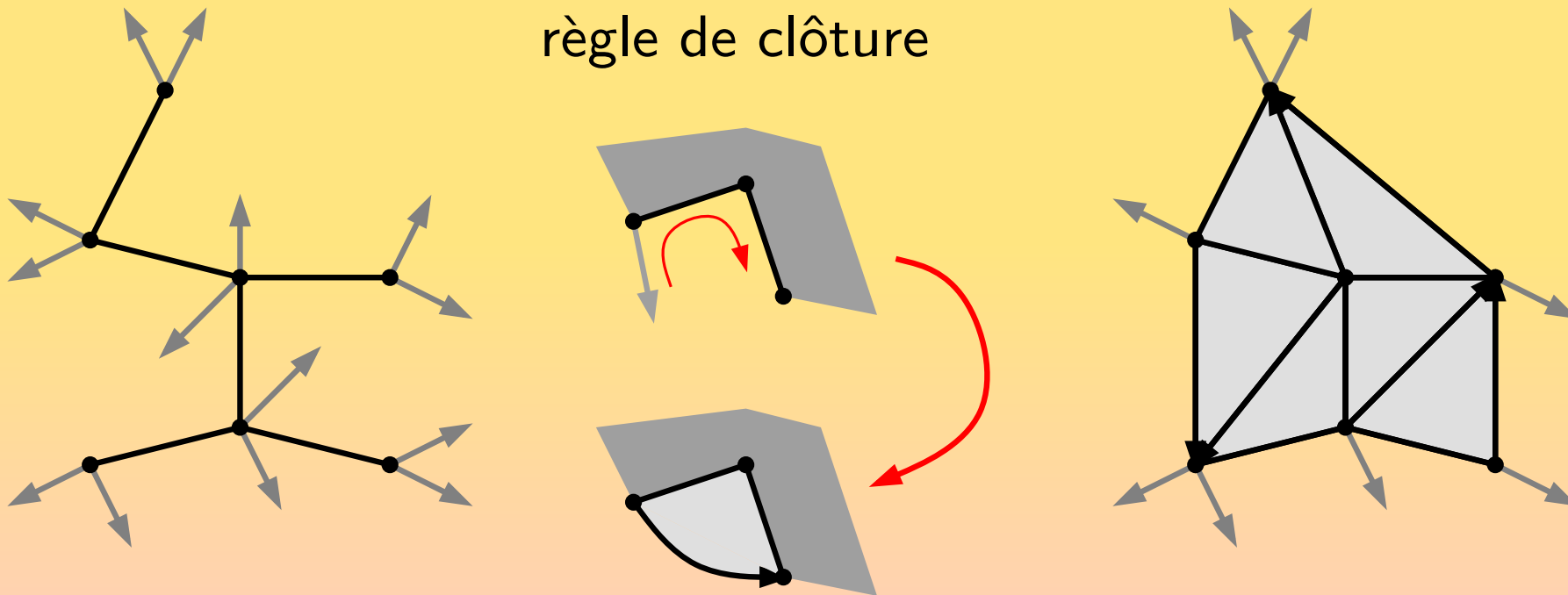


Lorsque la clôture s'arrête, il reste :

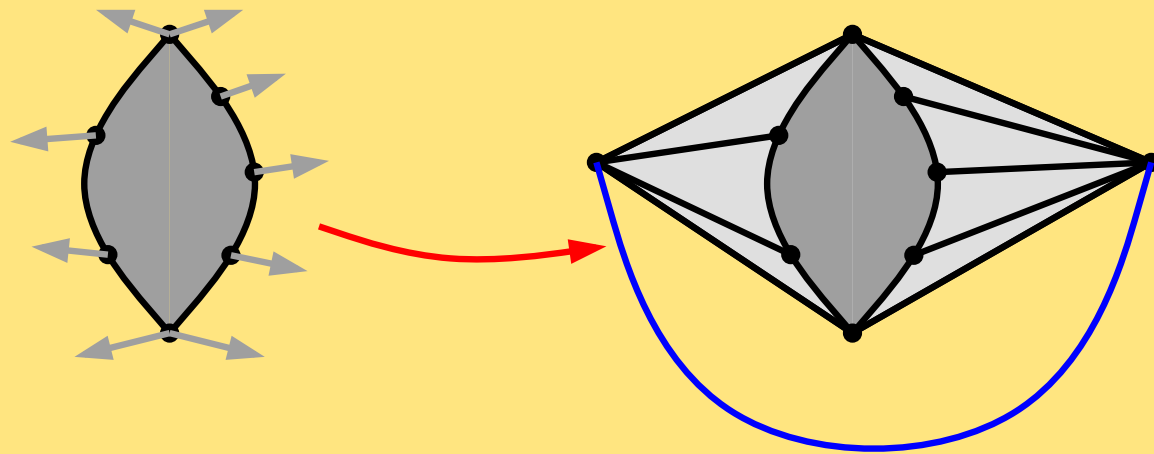


**Théorème.** La clôture est une bijection entre arbres à 2 feuilles par nœuds à  $n$  nœuds et triangulations marquées à  $n + 2$  sommets.

règle de clôture

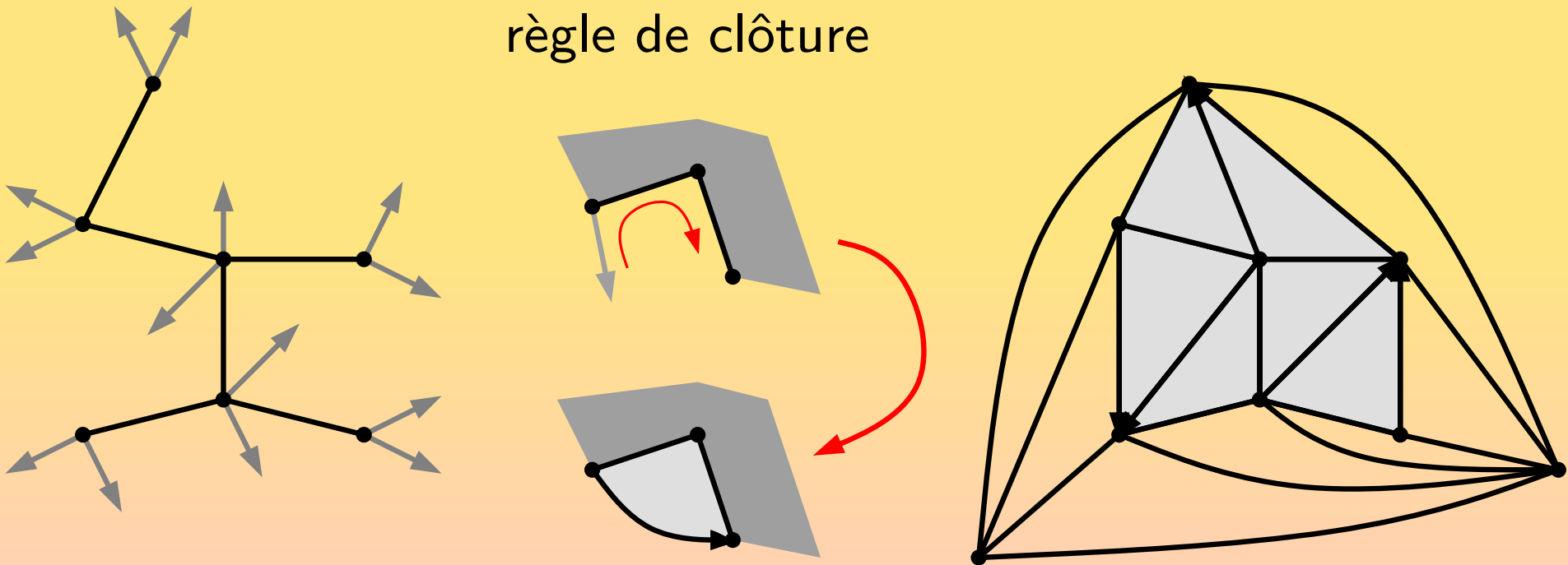


Lorsque la clôture s'arrête, il reste :

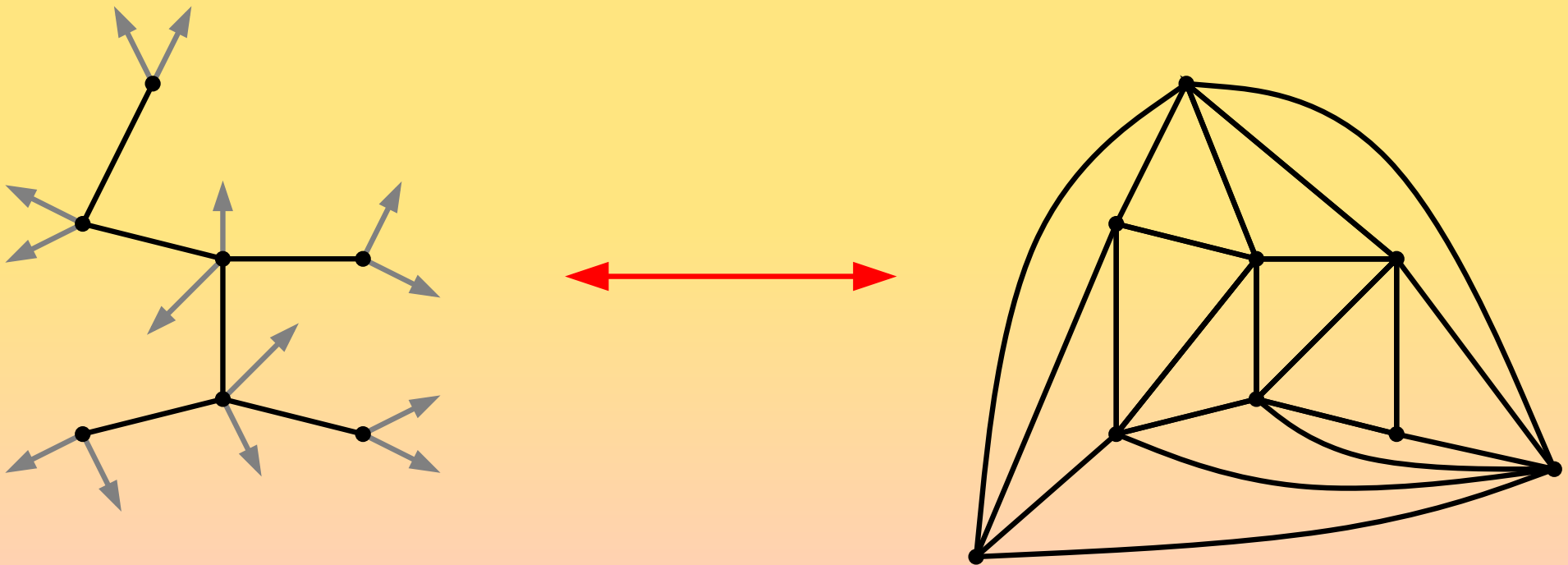


**Théorème.** La clôture est une bijection entre arbres à 2 feuilles par nœuds à  $n$  nœuds et triangulations marquées à  $n + 2$  sommets.

règle de clôture

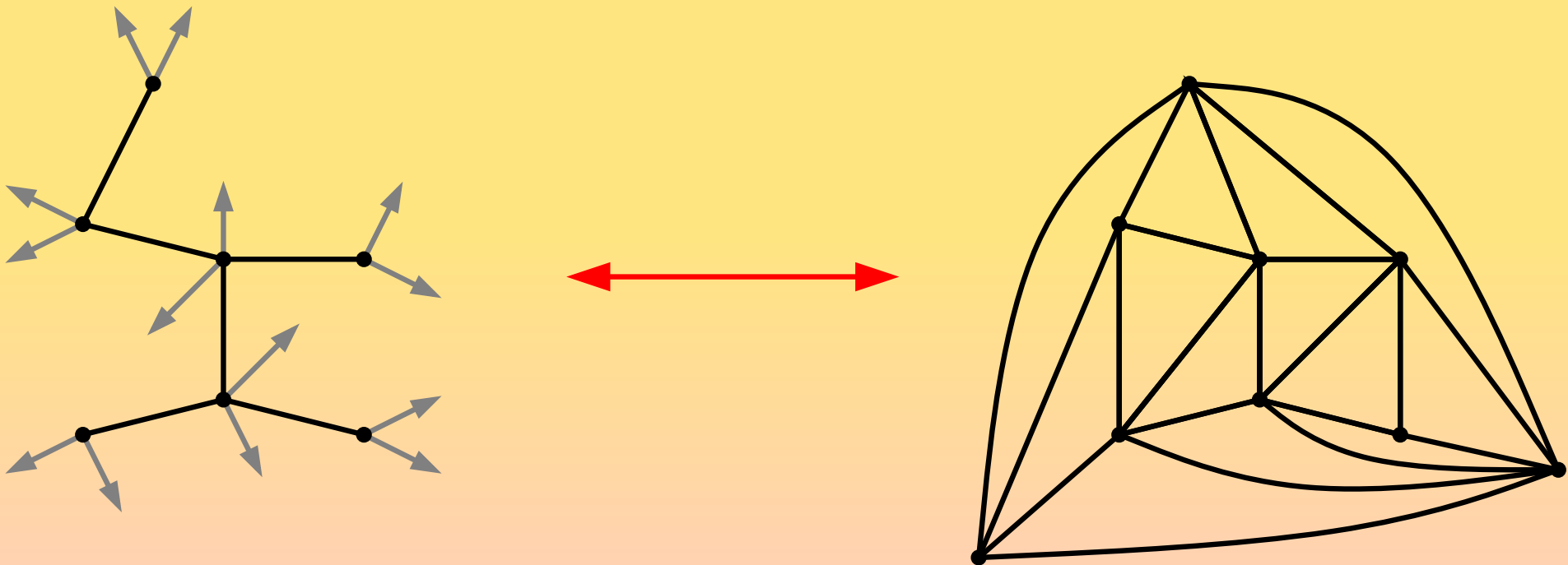


**Théorème.** (Poulalhon–Schaeffer 03) La clôture est une bijection entre arbres à 2 feuilles par nœuds à  $n$  nœuds et triangulations marquées à  $n + 2$  sommets.



**Théorème.** (Poulalhon–Schaeffer 03) La clôture est une bijection entre arbres à 2 feuilles par nœuds à  $n$  nœuds et triangulations marquées à  $n + 2$  sommets.

⇒ le code de l'arbre est un code optimal de la triangulation

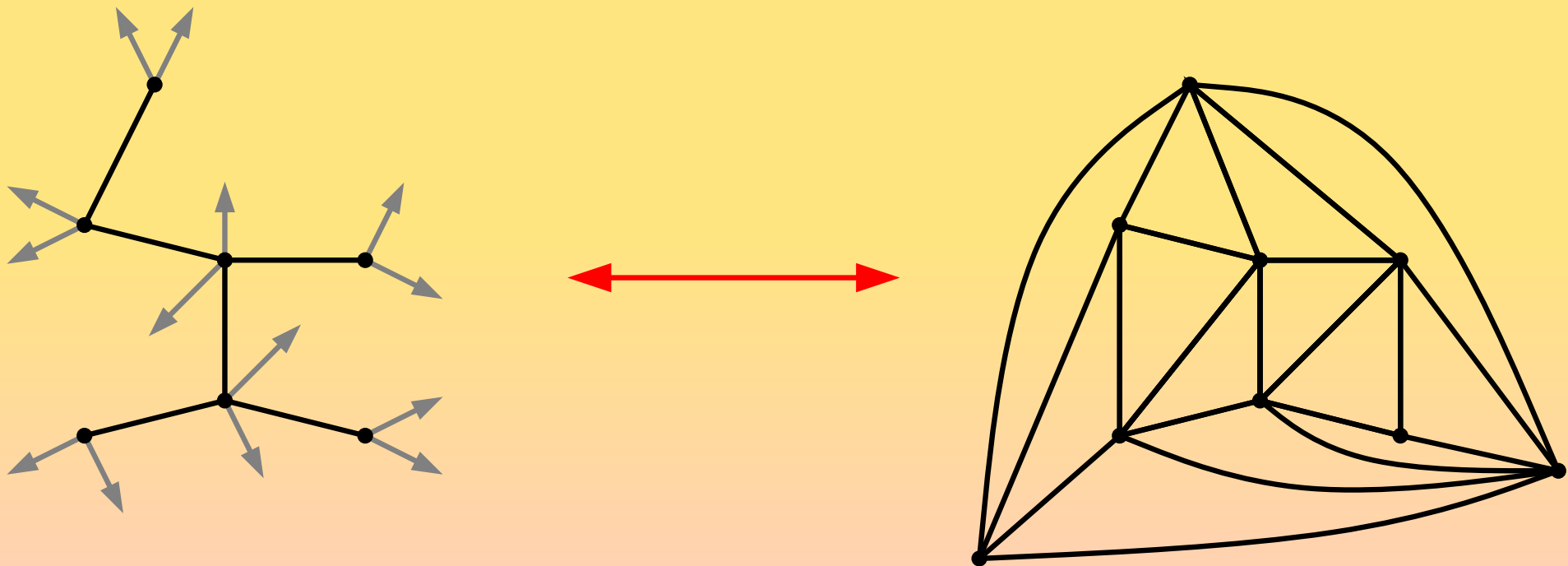




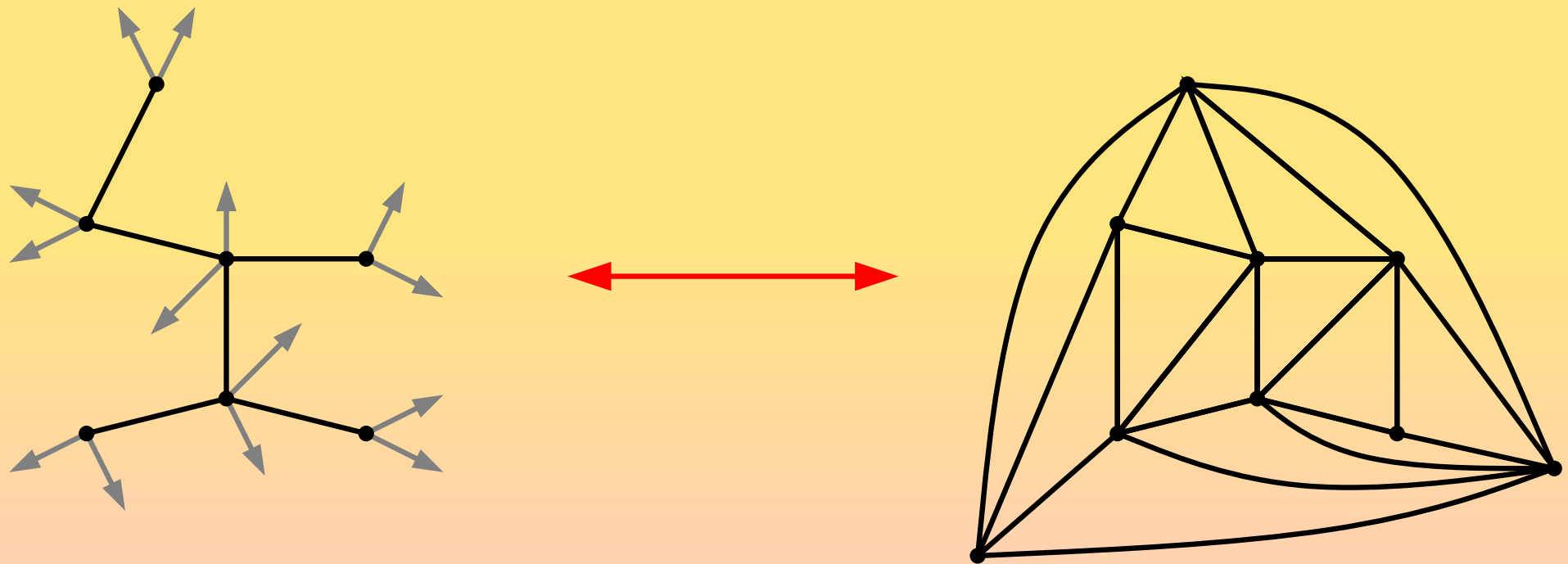
**Théorème.** (Poulalhon–Schaeffer 03) La clôture est une bijection entre arbres à 2 feuilles par nœuds à  $n$  nœuds et triangulations marquées à  $n + 2$  sommets.

⇒ le code de l'arbre est un code optimal de la triangulation

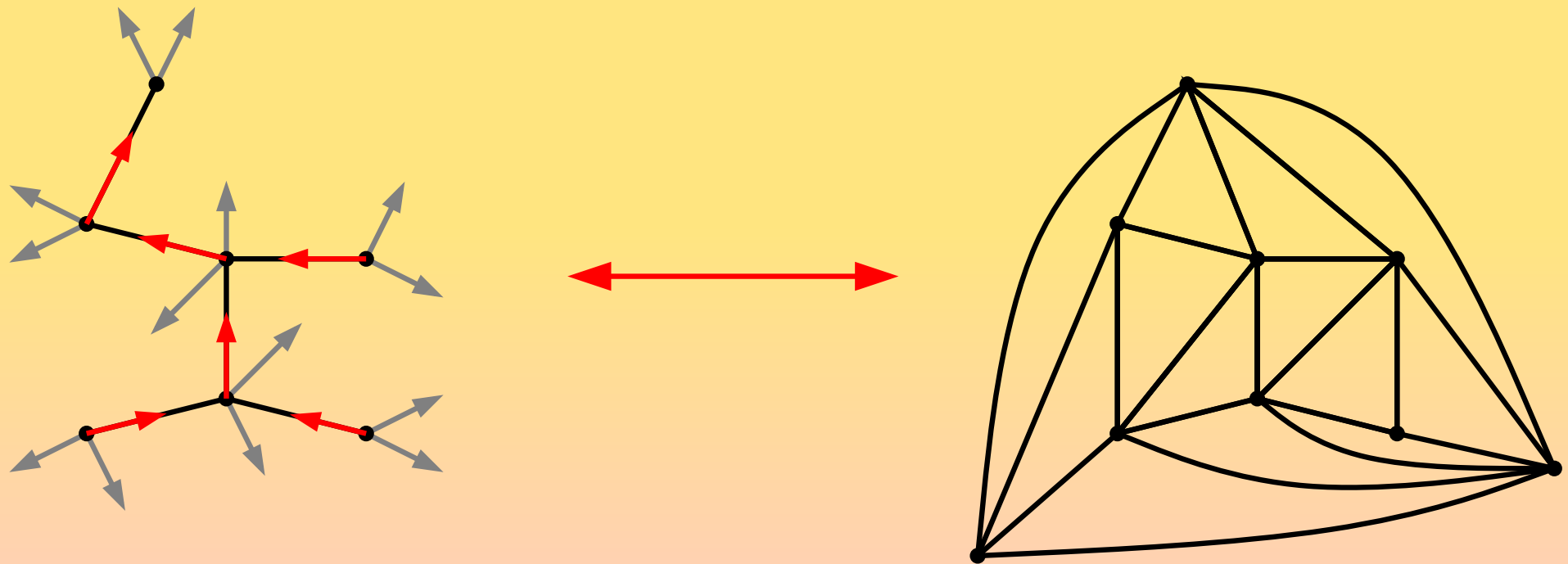
mais pour définir un codage il faut pouvoir retrouver l'arbre...



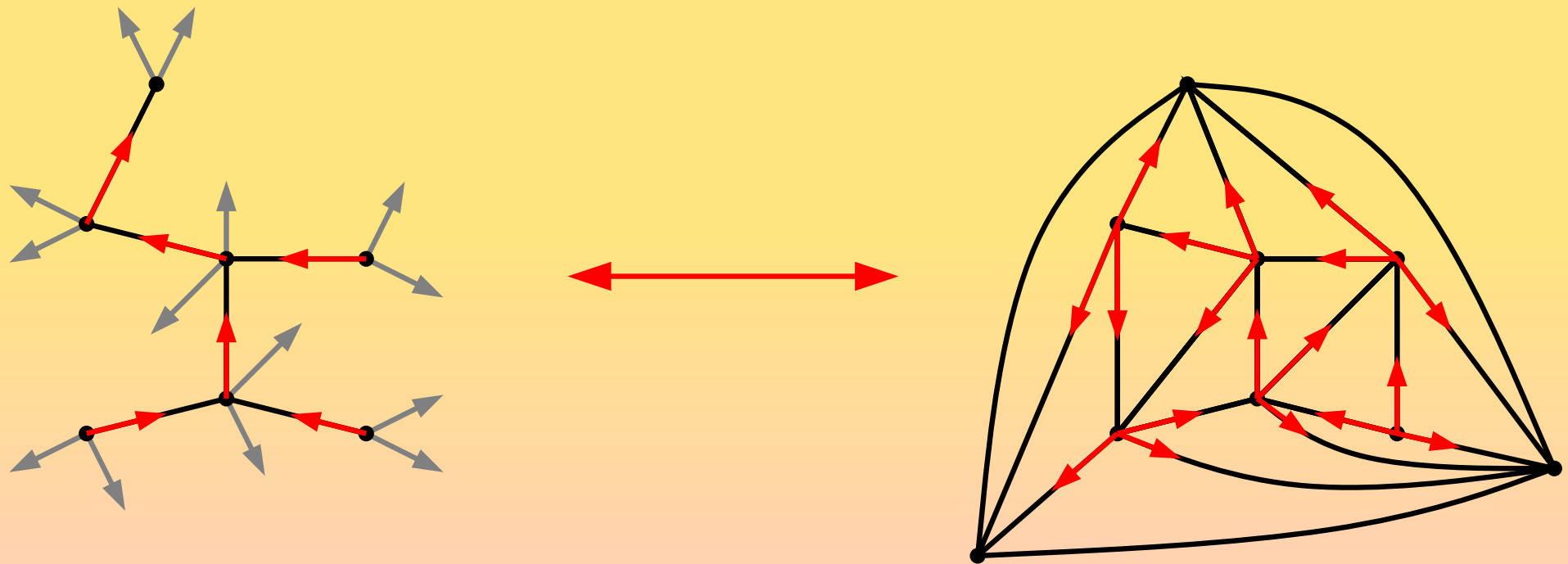
L'arbre est naturellement muni d'une 3-orientation,



L'arbre est naturellement muni d'une 3-orientation,

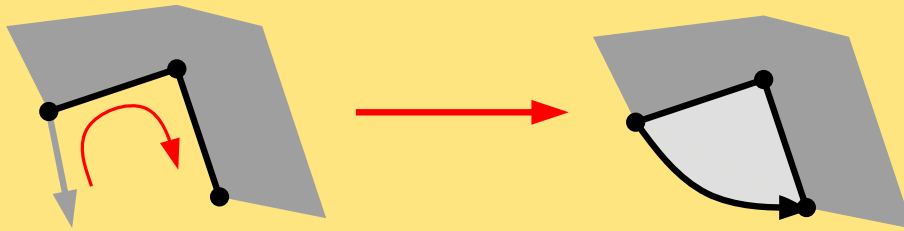


L'arbre est naturellement muni d'une 3-orientation,

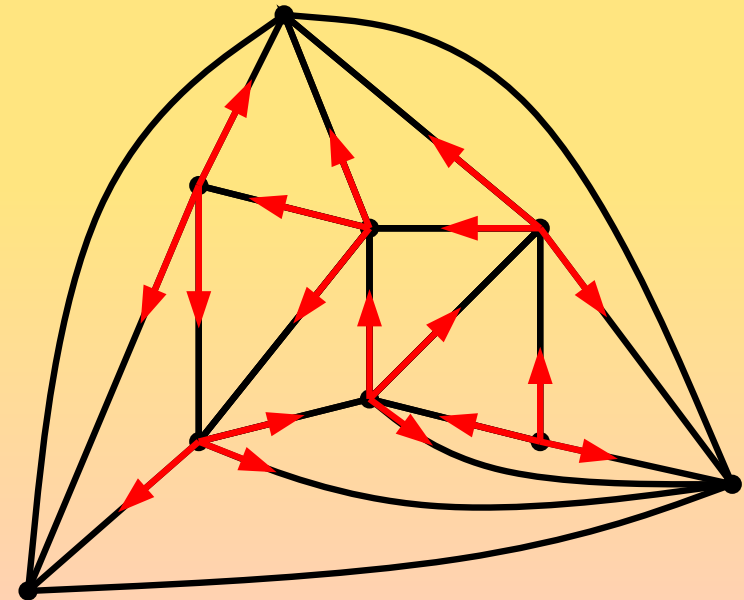
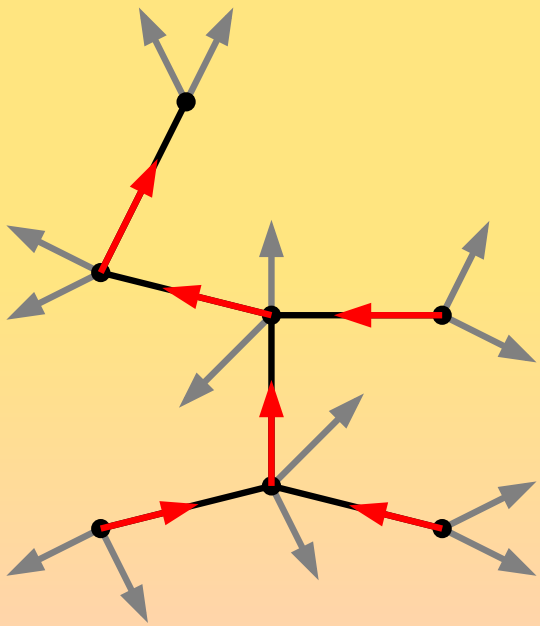


L'arbre est naturellement muni d'une 3-orientation,

**Lemme.** La clôture munit la triangulation de son unique orientation sans cycle indirect.

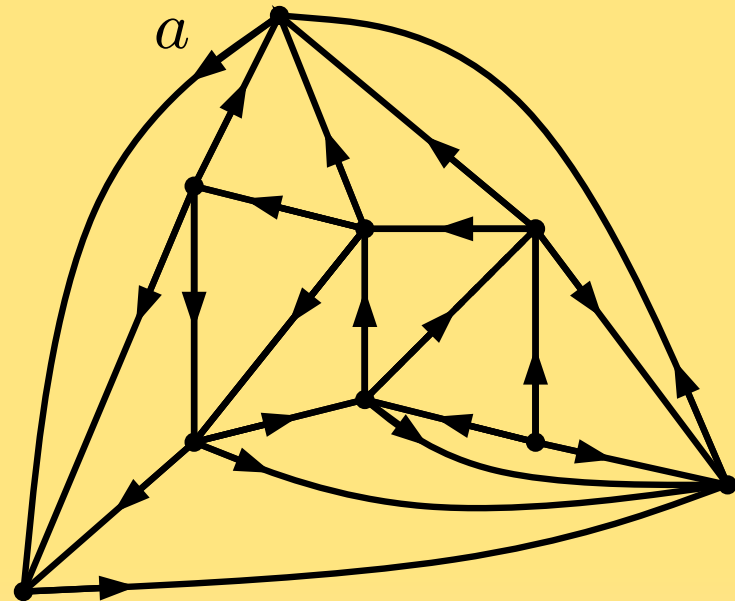


L'arête qui clôt un circuit est de sens direct sur le bord.



Soit  $C$  une carte munie d'une orientation sans cycle indirect.

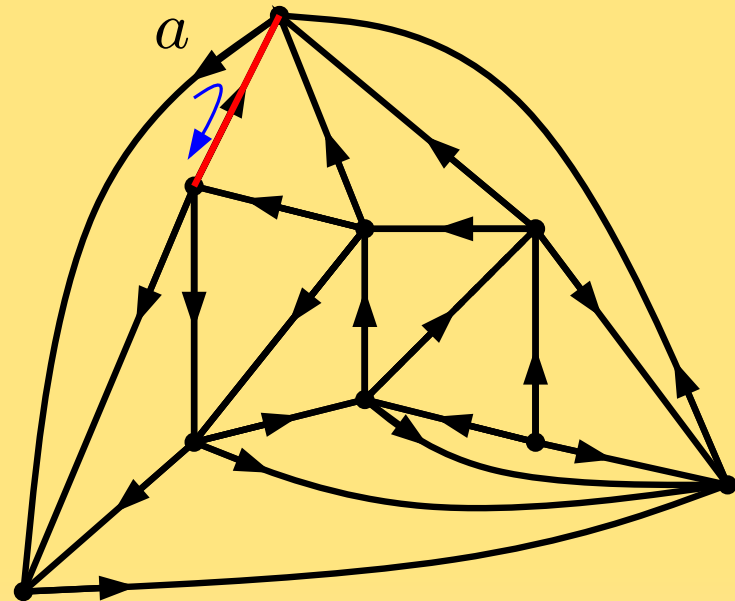
Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.



**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .

Soit  $C$  une carte munie d'une orientation sans cycle indirect.

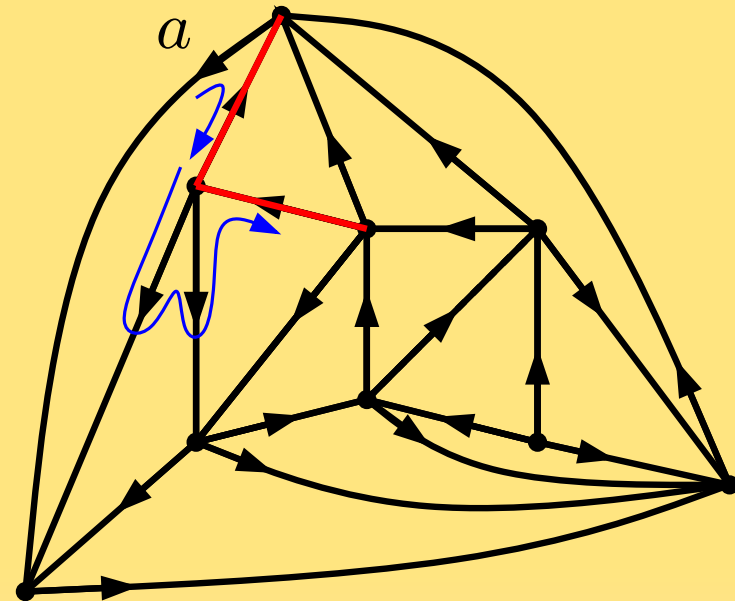
Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.



**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .

Soit  $C$  une carte munie d'une orientation sans cycle indirect.

Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.

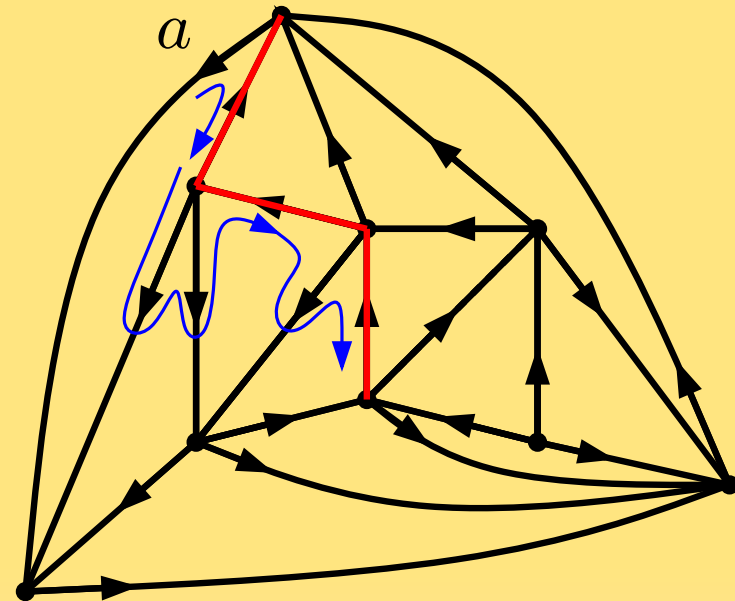


**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .



Soit  $C$  une carte munie d'une orientation sans cycle indirect.

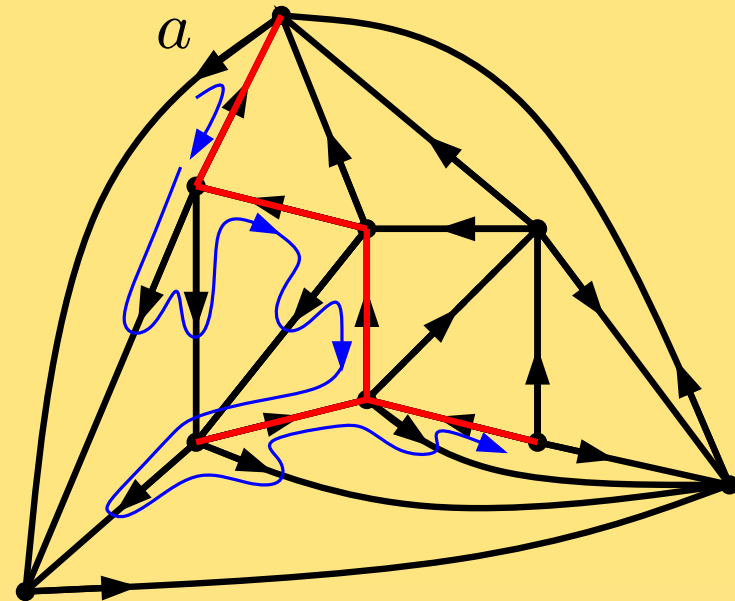
Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.



**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .

Soit  $C$  une carte munie d'une orientation sans cycle indirect.

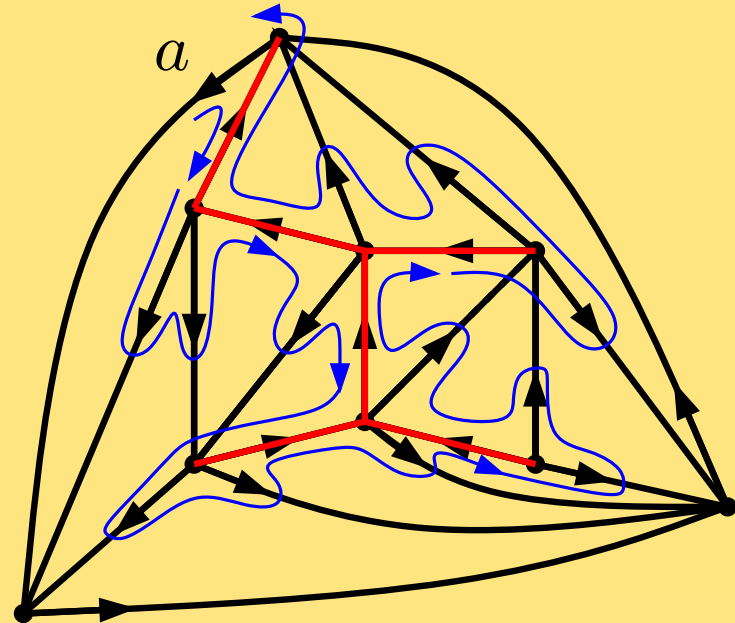
Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.



**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .

Soit  $C$  une carte munie d'une orientation sans cycle indirect.

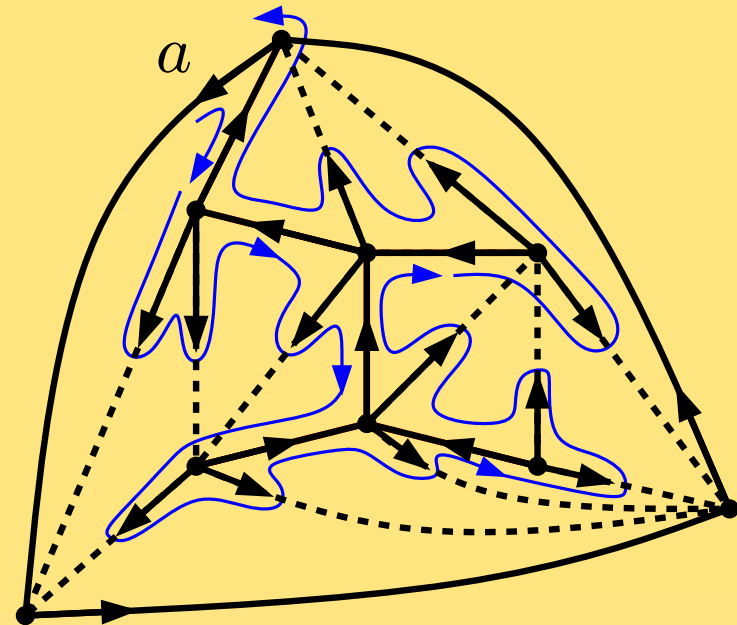
Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.



**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .

Soit  $C$  une carte munie d'une orientation sans cycle indirect.

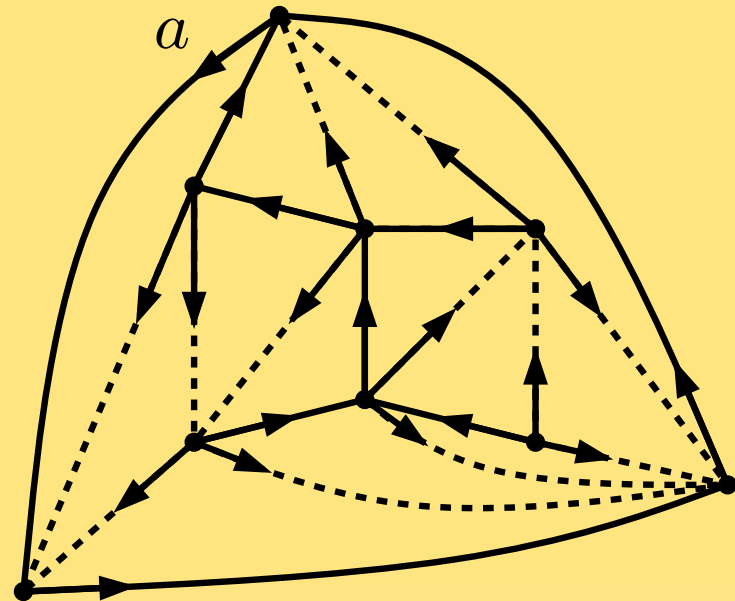
Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.



**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .

Soit  $C$  une carte munie d'une orientation sans cycle indirect.

Soit  $a$  une arête dont l'origine est accessible depuis tout sommet.



**Théorème.** (PS 03, Fusy 04, Bernardi 05) Le parcours à rebours en profondeur à gauche à partir de  $a$  construit un arbre couvrant dont la clôture redonne  $C$ .

En résumé on a donc l'algorithme de codage optimal suivant:

Soit  $T$  une triangulation planeaire.

En résumé on a donc l'algorithme de codage optimal suivant:

Soit  $T$  une triangulation planeaire.

- Construire la forêt de Schnyder minimale de  $T$ .  
Utilise l'algorithme glouton pour l'ordre minimal avec choix à chaque étape du sommet le plus à droite.

En résumé on a donc l'algorithme de codage optimal suivant:

Soit  $T$  une triangulation planeaire.

- Construire la forêt de Schnyder minimale de  $T$ .  
Utilise l'algorithme glouton pour l'ordre minimal avec choix à chaque étape du sommet le plus à droite.
- Obtenir l'arbre par le parcours à rebours en profondeur à gauche, et coder l'arbre au fur et à mesure.



En résumé on a donc l'algorithme de codage optimal suivant:

Soit  $T$  une triangulation planeaire.

- Construire la forêt de Schnyder minimale de  $T$ .  
Utilise l'algorithme glouton pour l'ordre minimal avec choix à chaque étape du sommet le plus à droite.
- Obtenir l'arbre par le parcours à rebours en profondeur à gauche, et coder l'arbre au fur et à mesure.

Le décodage peut se faire au fur et à mesure de l'arrivée des bits du code: on reconstruit ainsi une région initiale.

Ces résultats s'étendent à d'autres types de cartes, en particulier les cartes 3-connexes des maillages polyédraux (Fusy–P–S 05), les triangulations 4-connexes (Fusy 05)...

Au delà du codage on peut étendre et combiner ces outils avec d'autres pour donner des représentations compactes (avec requêtes)  $\Rightarrow$  cf. Luca Castelli Aleardi.

Ces résultats s'étendent à d'autres types de cartes, en particulier les cartes 3-connexes des maillages polyédraux (Fusy–P–S 05), les triangulations 4-connexes (Fusy 05)...

Au delà du codage on peut étendre et combiner ces outils avec d'autres pour donner des représentations compactes (avec requêtes)  $\Rightarrow$  cf. Luca Castelli Aleardi.

Merci de votre attention.