

À propos du nombre de graphes planaires

Nicolas Bonichon - LaBRI

Cyril Gavoille - LaBRI

Nicolas Hanusse - LaBRI

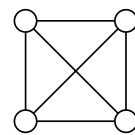
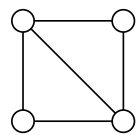
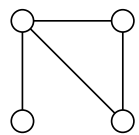
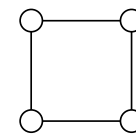
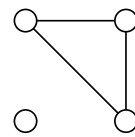
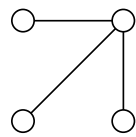
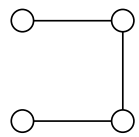
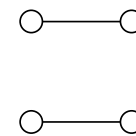
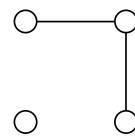
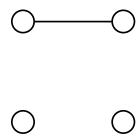
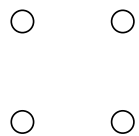
19 avril 2002

- Comment coder “efficacement” un graphe planaire à n sommets ?
- Quelle est la quantité maximale d’information que peut contenir un graphe planaire à n sommets ?
- Combien vaut $p(n)$, le nombre de graphes planaires à n sommets ?

(graphes sans boucle ni multi-arête, non étiquetés, non dessinés)

$$p(n = 1, 2, 3, 4, 5, 6 \dots) = 1, 2, 4, 11, 33, 147, \dots$$

$$p(4) = 11$$



Asymptotique sur $p(n)$? sur $\log_2 p(n)$?

$$\frac{1}{O(n)} \cdot T_n \leq p(n) \leq 2^{3n-6} \cdot T_n$$

T_n = nombre de triangulations du plan avec n sommets
(dont une faces a été fixée)

$$T_n \sim \Theta(n^{-c}) \cdot \left(\frac{256}{27}\right)^n \quad c > 0 \text{ constante [Tutte '62]}$$

$$\Rightarrow 3.24n + \Theta(\log n) \leq \log_2 p(n) \leq 6.24n + \Theta(\log n)$$

avec $\log_2(256/27) \approx 3.24\dots$

Théorème. Pour tout n assez grand,

$$\beta n - \Theta(\log n) \leq \log_2 p(n) \leq \alpha n + O(\log n)$$

avec $\beta \approx 4.71\dots$ et $\alpha \approx 5.01\dots$

β ? Nombre de graphes planaires étiquetés 2-connexes
(non dessinés)

$$\sim \Theta(n^{-5/7}) \cdot 2^{\beta n} \cdot n! \quad [\text{Bender-Gao-Wormald '99}]$$

α ???

α ???

Nombre de graphes planaires étiquetés

$$\leq 2^{5.22n} \cdot n! \quad [\text{Osthus-Prömel-Taraz '01}]$$

$$\not\Rightarrow \alpha \leq 5.22$$

Nombre de cartes planaires enracinées simples à n sommets

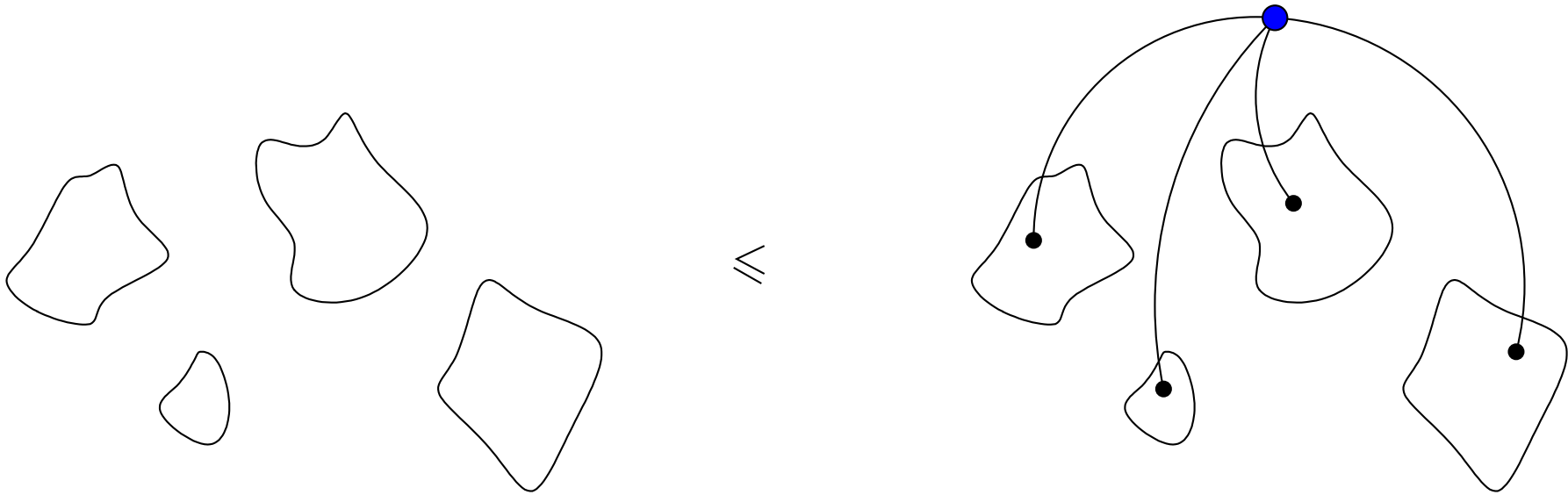
$$\sim 2^{5.098n+o(n)} \quad [\text{Bousquet-Mélou '02}]$$

$$\Rightarrow \alpha \leq 5.098$$

Remarque. Soit $q(n)$ le nombre de graphes planaires connexes.

Alors,

$$p(n) \leq (n+1) \cdot q(n+1) \leq (n+1) \cdot (n+1)^5 \cdot q(n)$$



$$\Rightarrow \boxed{\log_2 p(n) \leq \log_2 q(n) + O(\log n)}$$

Codage d'un graphe planaire (connexe) :

Codage d'un graphe planaire (connexe) :

1. Construction d'une carte particulière du graphe

Codage d'un graphe planaire (connexe) :

1. Construction d'une carte particulière du graphe
2. Triangulation spécifique des faces de la carte
+ 2 sommets (\Rightarrow "super-triangulation")

Codage d'un graphe planaire (connexe) :

1. Construction d'une carte particulière du graphe
2. Triangulation spécifique des faces de la carte
+ 2 sommets (\Rightarrow "super-triangulation")
3. Codage de la "super-triangulation"
+ codage de certaines arêtes à supprimer

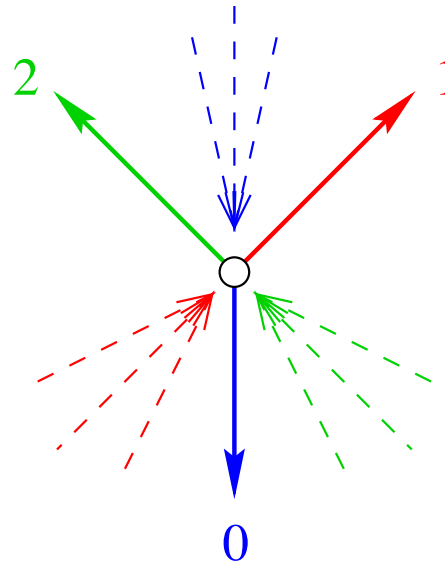
Partie I

Préliminaires

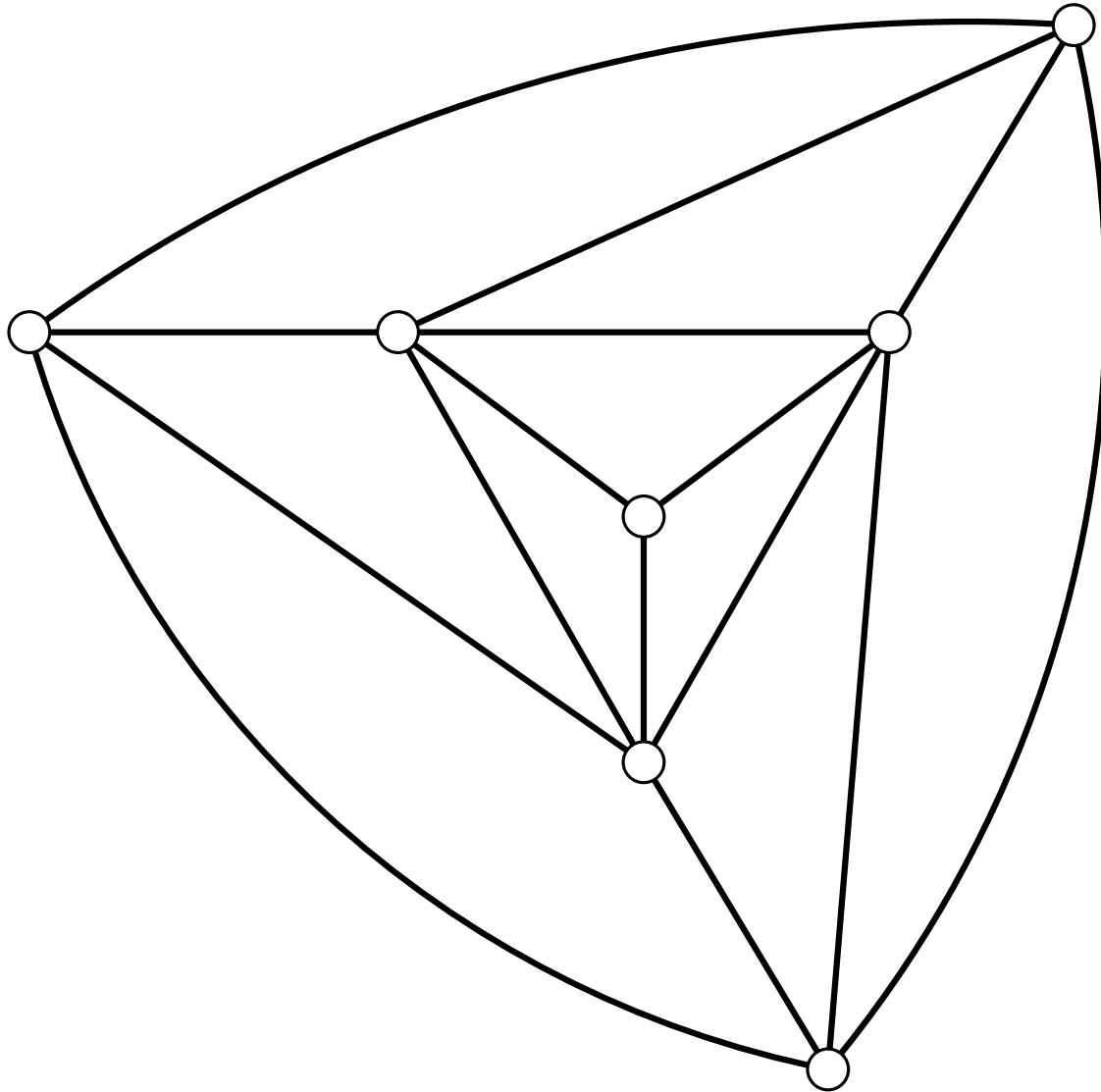
Arbres de Schnyder – réalisateur

[Schnyder '89] Toute triangulation du plan (=planaire maximal dessiné) a une partition de ses arêtes internes en trois arbres (T_0, T_1, T_2), chacun enraciné en un sommet de la face extérieure telle que :

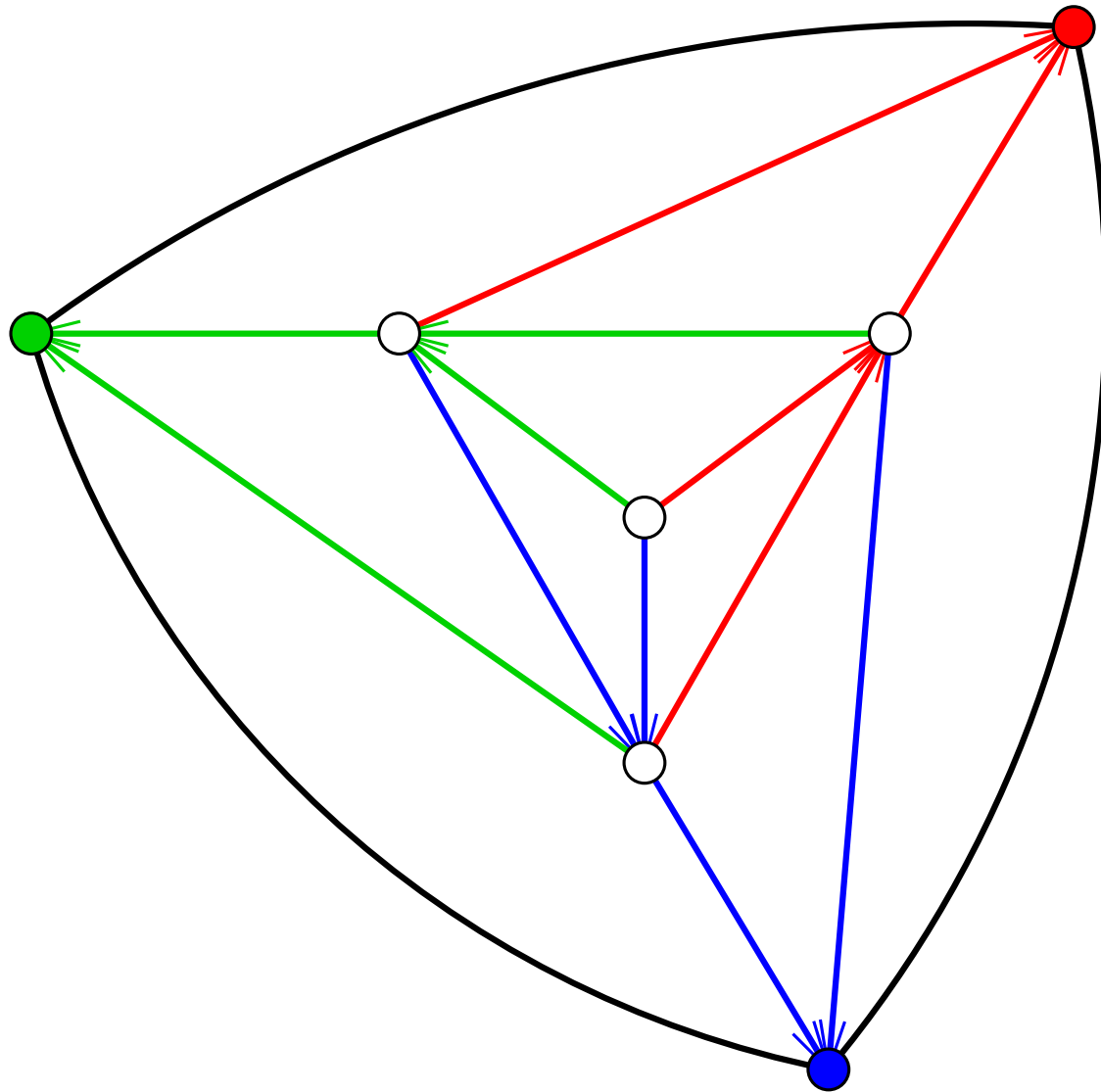
pour tout sommet interne :



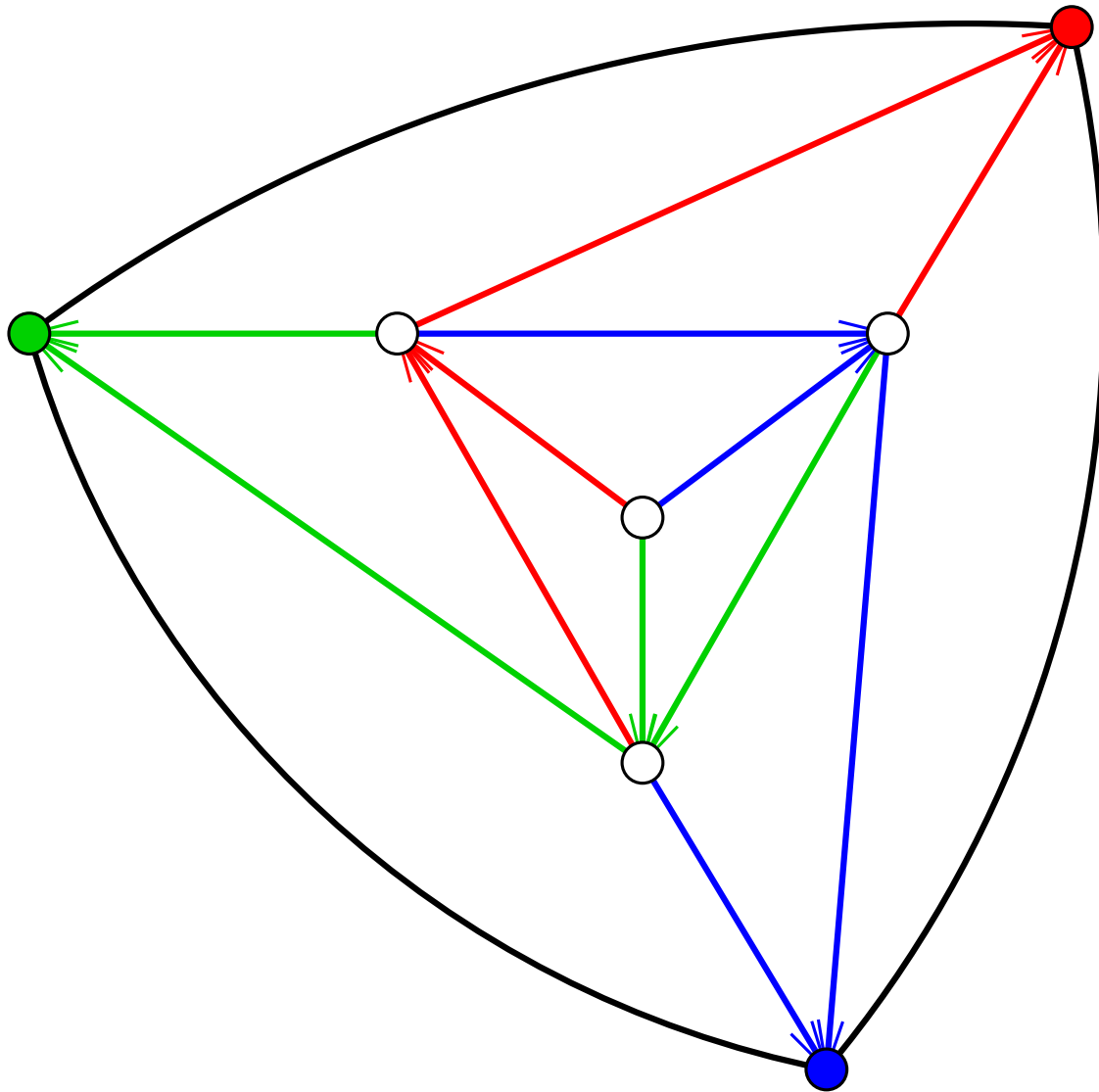
Une triangulation ...



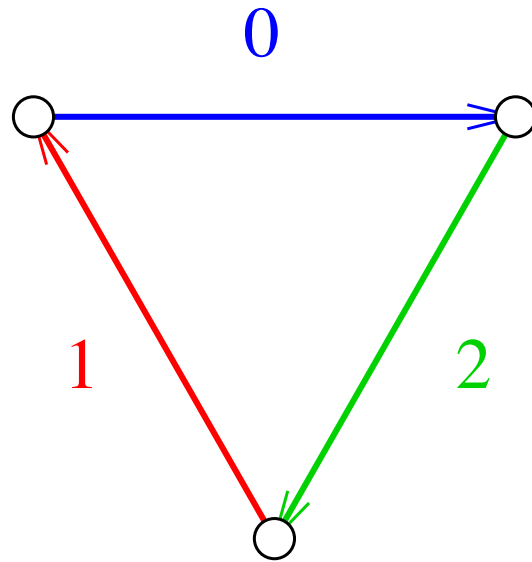
... un réalisateur



... un autre réalisateur.



Un cw-triangle



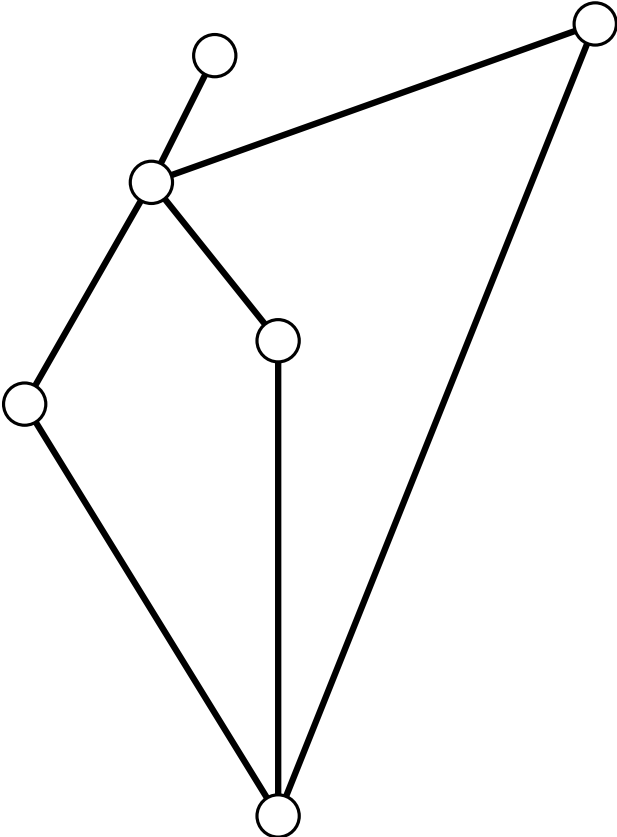
“Super”-triangulation

$$p_i(v) = \text{père de } v \text{ dans } T_i$$

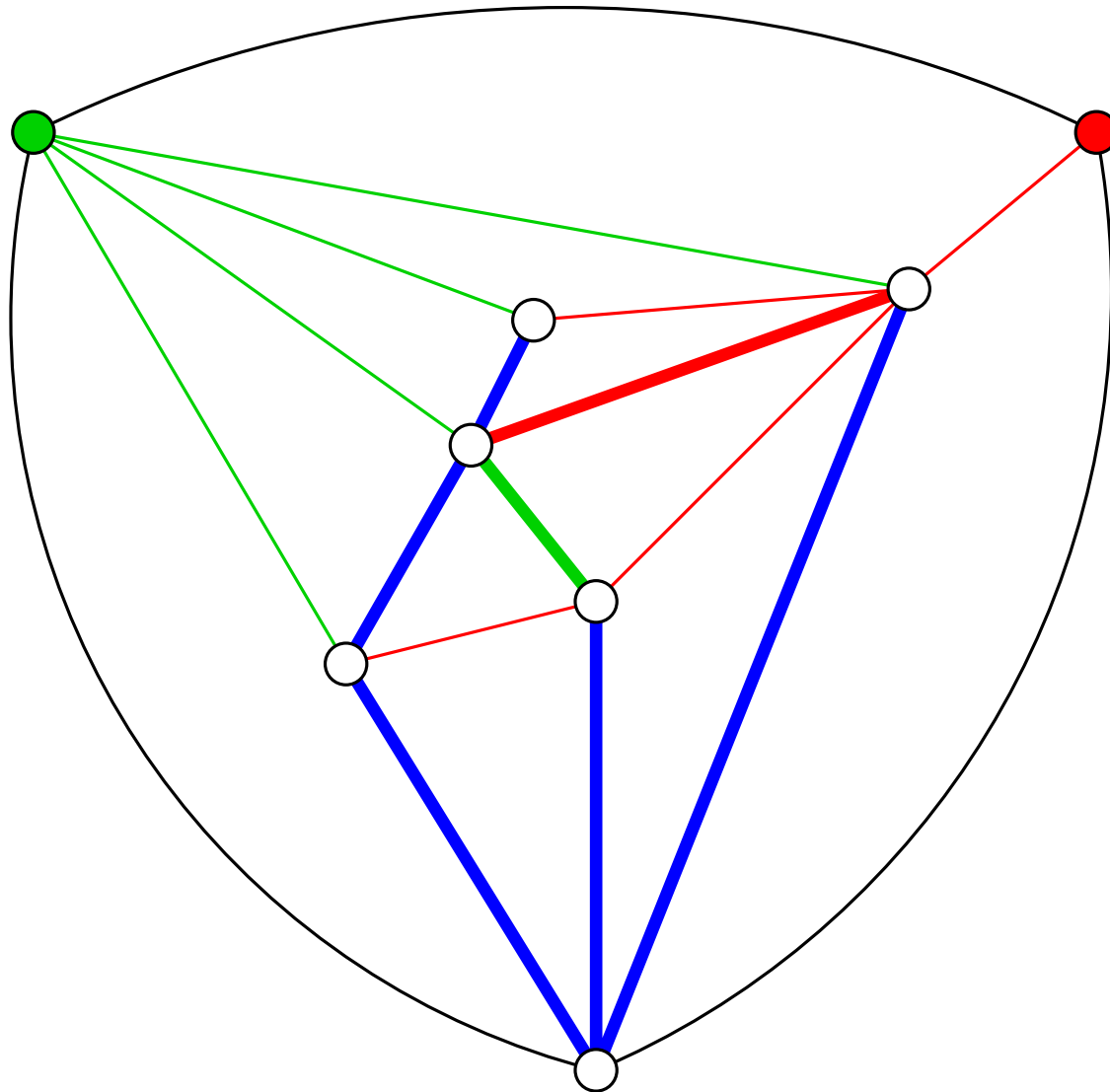
Une **super-triangulation** d'un graphe planaire connexe G est un réalisateur (T_0, T_1, T_2) tel que :

1. $V(G) = V(T_0)$;
2. $E(T_0) \subseteq E(G) \subset E(T_0 \cup T_1 \cup T_2)$;
3. (T_0, T_1, T_2) ne contient pas de cw-triangle ;
4. $(v, p_1(v)) \in E(G)$ pour tout nœud interne v de T_2 .

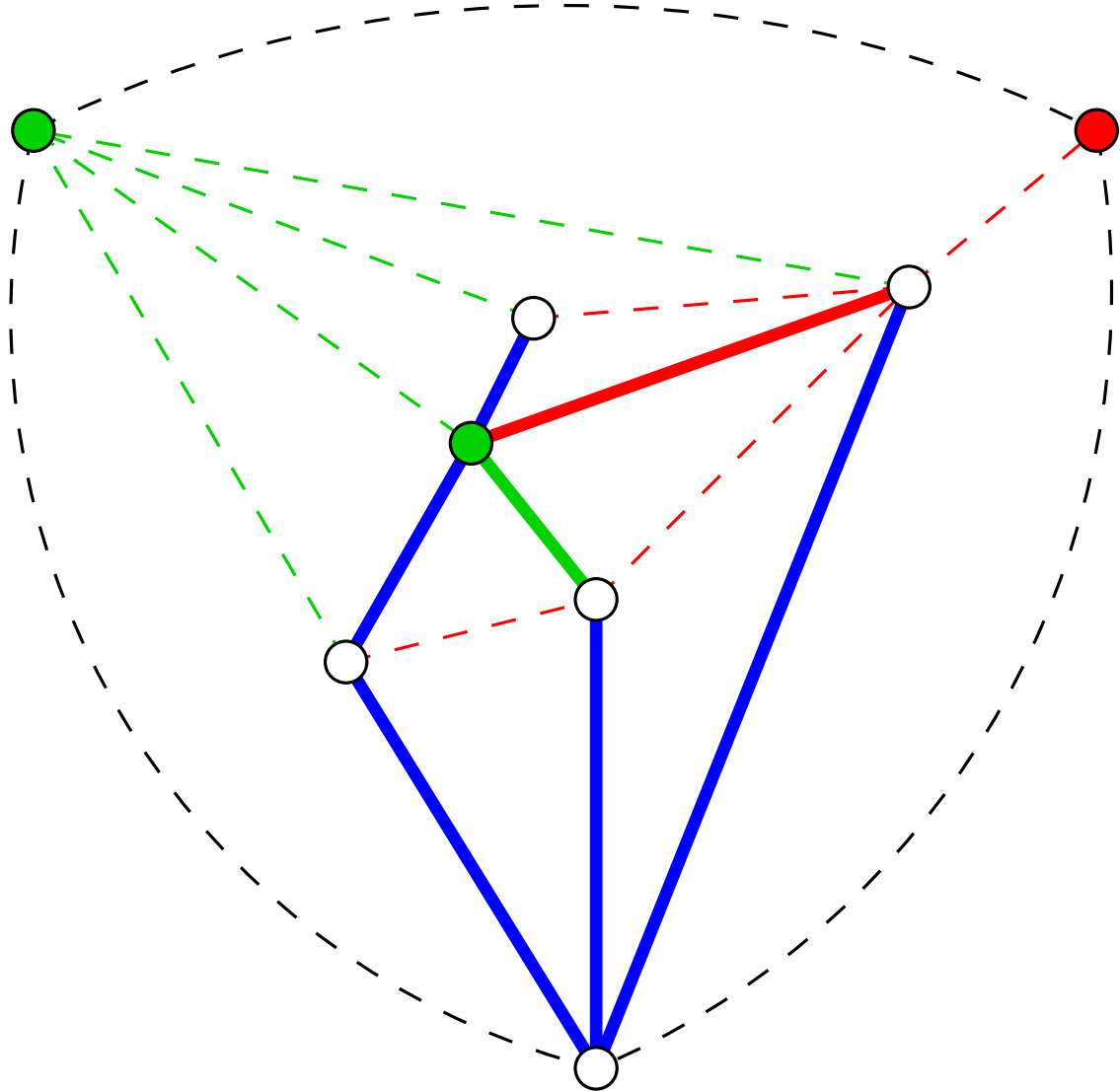
Un graphe G ...



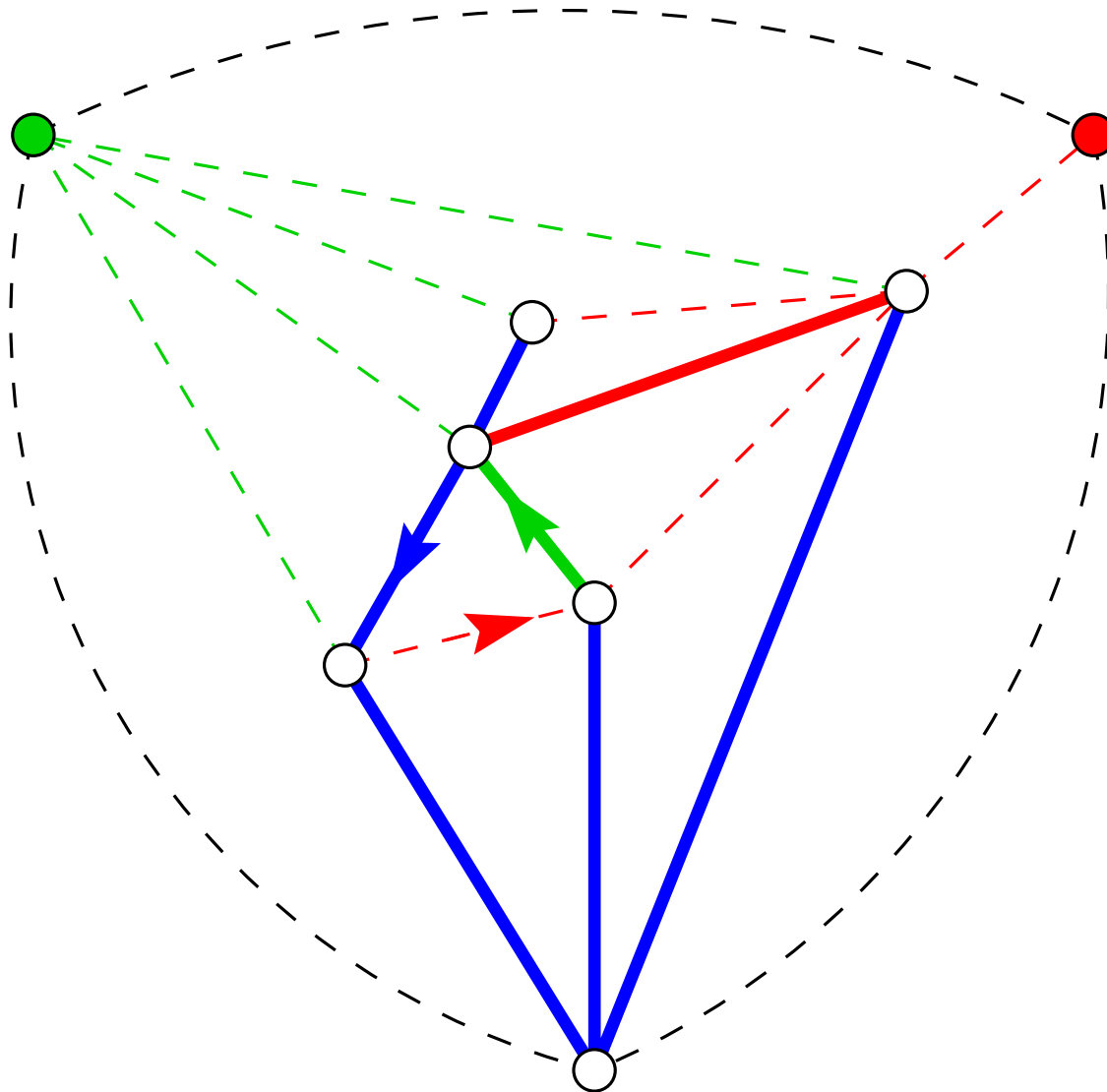
... une super-triangulation de G



avec les nœuds internes verts ...



... et sans cw-triangle.

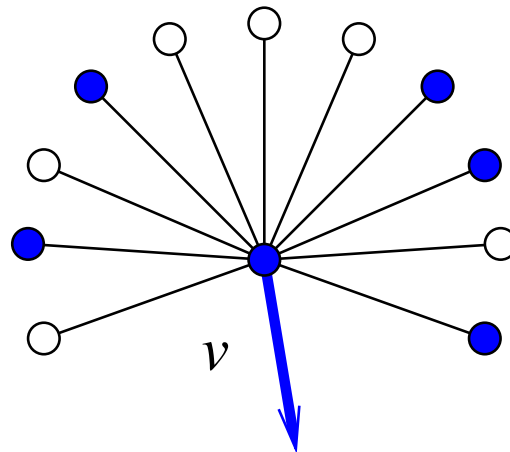


Partie II
Construction d'une
super-triangulation

Arbres “plus-à-gauche”

Un arbre **plus-à-gauche** défini pour une carte M de G est un arbre enraciné T de G (pas nécessairement couvrant) obtenu en parcourant M de la manière suivante :

- 1) racine de T = un sommet de la face extérieure de M ;
- 2) pour tout sommet v

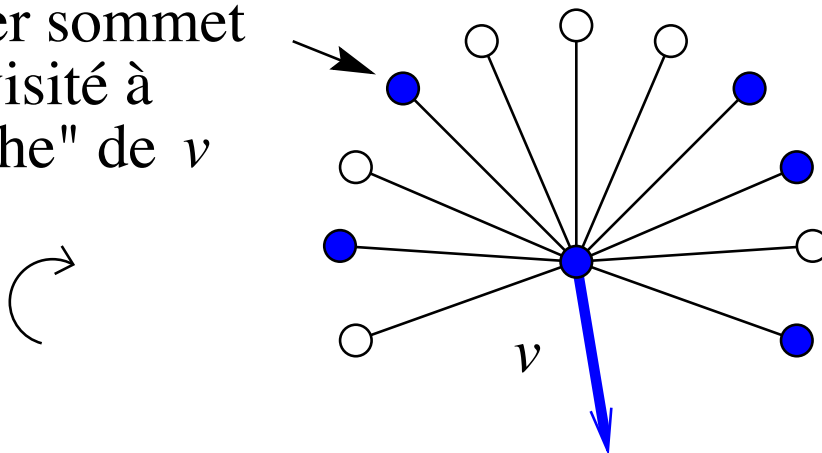


Arbres "plus-à-gauche"

Un arbre **plus-à-gauche** défini pour une carte M de G est un arbre enraciné T de G (pas nécessairement couvrant) obtenu en parcourant M de la manière suivante :

- 1) racine de $T =$ un sommet de la face extérieure de M ;
- 2) pour tout sommet v

dernier sommet
déjà visité à
"gauche" de v

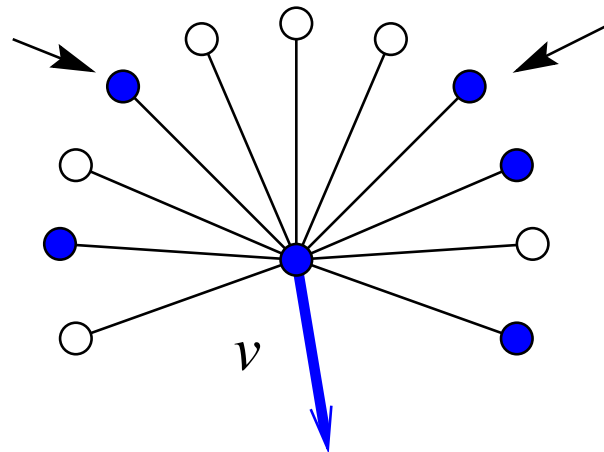


Arbres "plus-à-gauche"

Un arbre **plus-à-gauche** défini pour une carte M de G est un arbre enraciné T de G (pas nécessairement couvrant) obtenu en parcourant M de la manière suivante :

- 1) racine de $T =$ un sommet de la face extérieure de M ;
- 2) pour tout sommet v

dernier sommet
déjà visité à
"gauche" de v

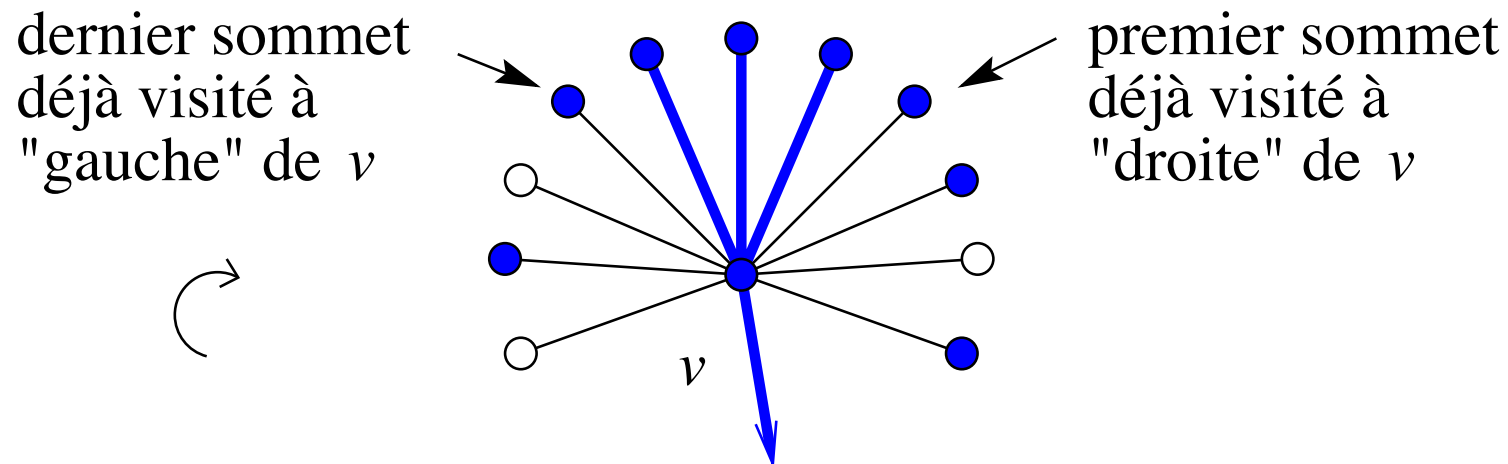


premier sommet
déjà visité à
"droite" de v

Arbres "plus-à-gauche"

Un arbre **plus-à-gauche** défini pour une carte M de G est un arbre enraciné T de G (pas nécessairement couvrant) obtenu en parcourant M de la manière suivante :

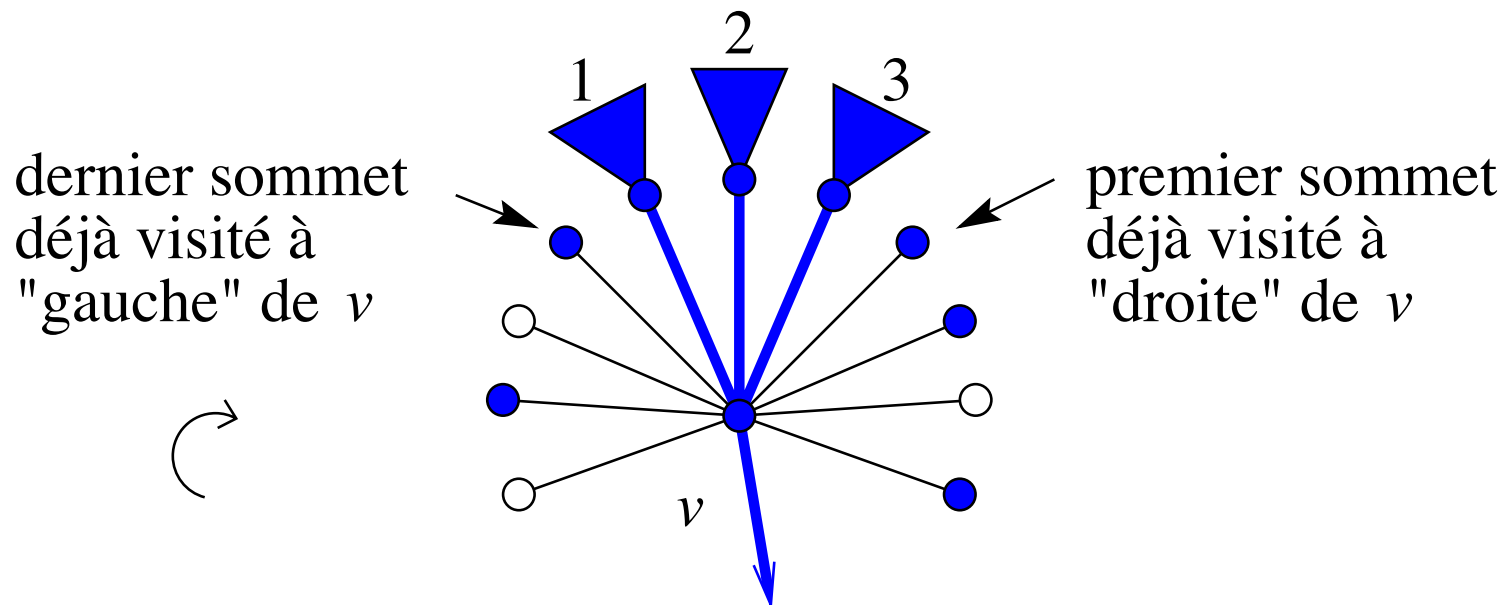
- 1) racine de $T =$ un sommet de la face extérieure de M ;
- 2) pour tout sommet v

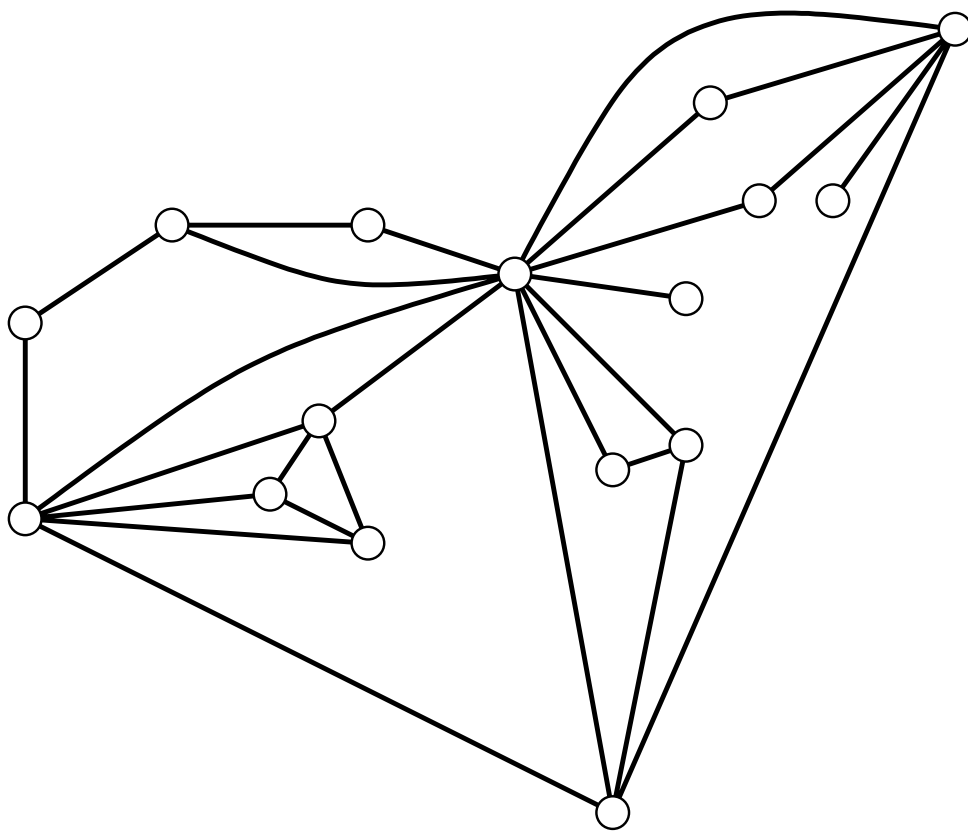


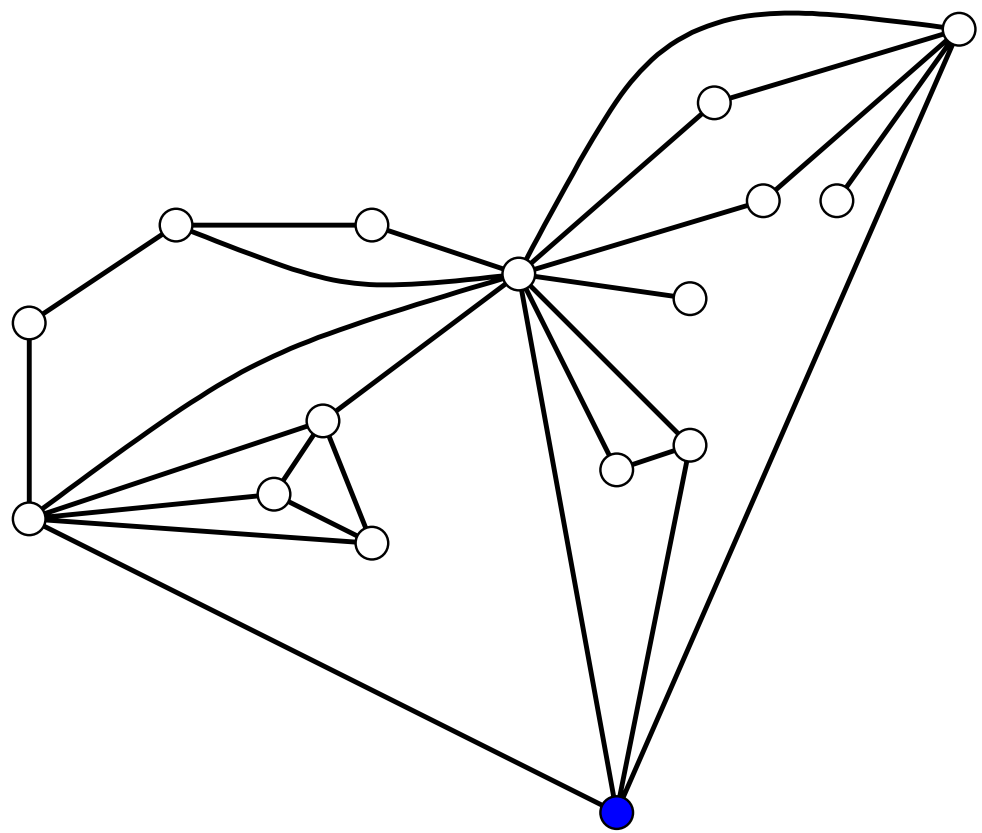
Arbres "plus-à-gauche"

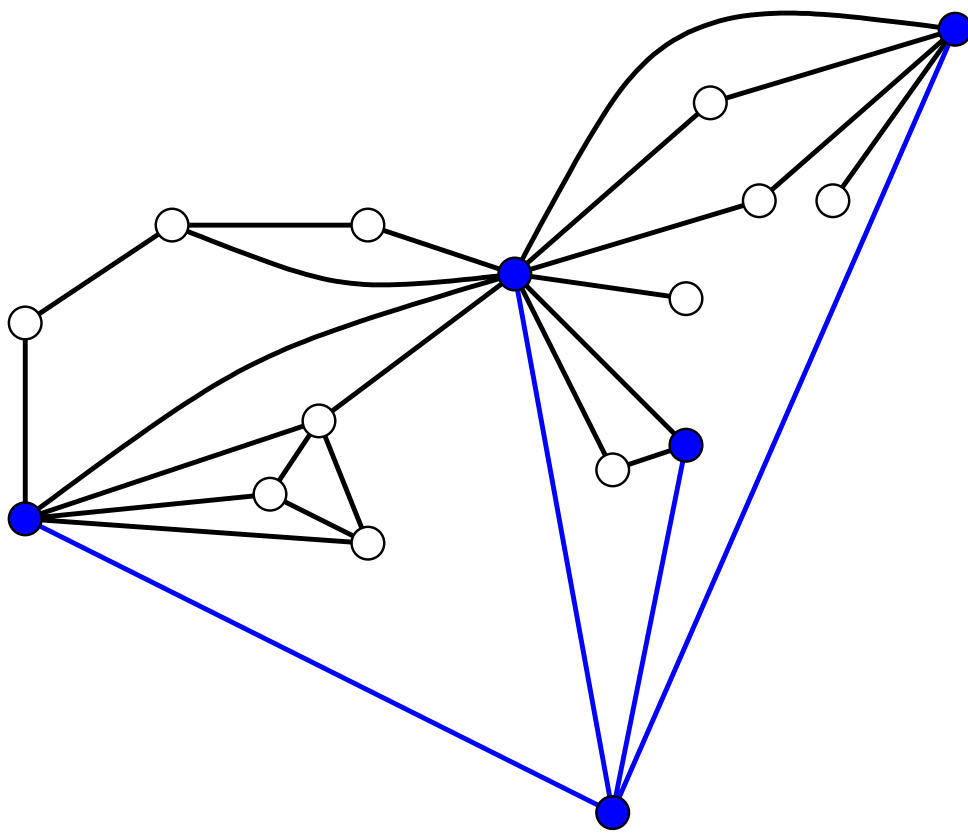
Un arbre **plus-à-gauche** défini pour une carte M de G est un arbre enraciné T de G (pas nécessairement couvrant) obtenu en parcourant M de la manière suivante :

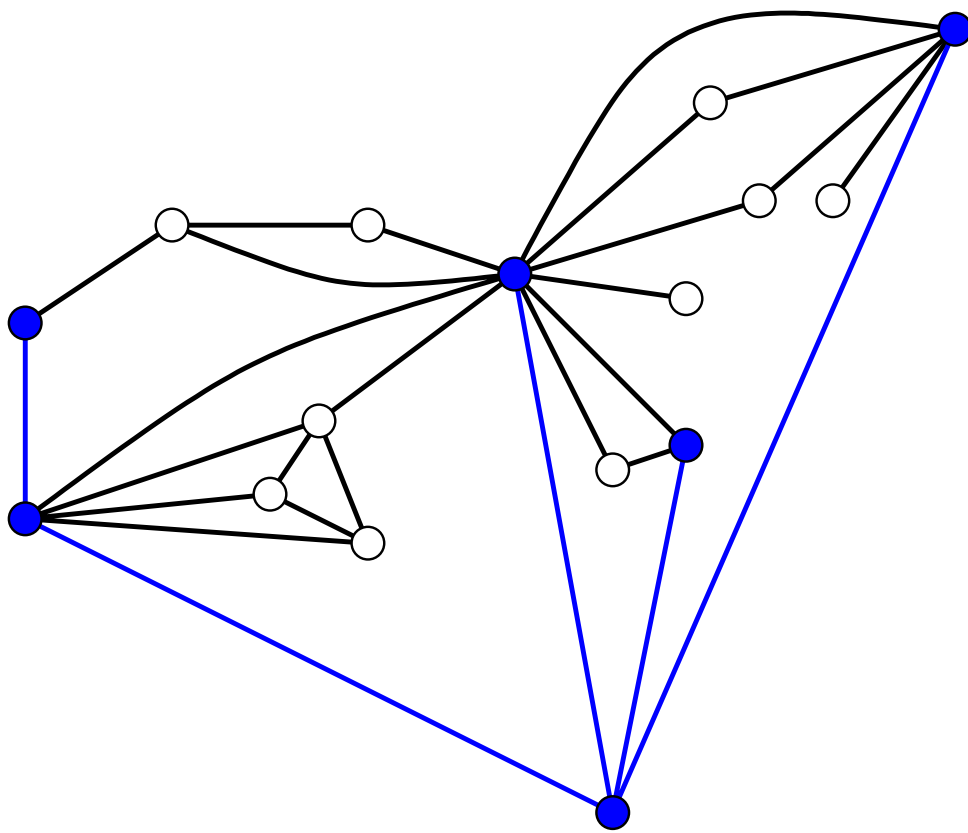
- 1) racine de $T =$ un sommet de la face extérieure de M ;
- 2) pour tout sommet v

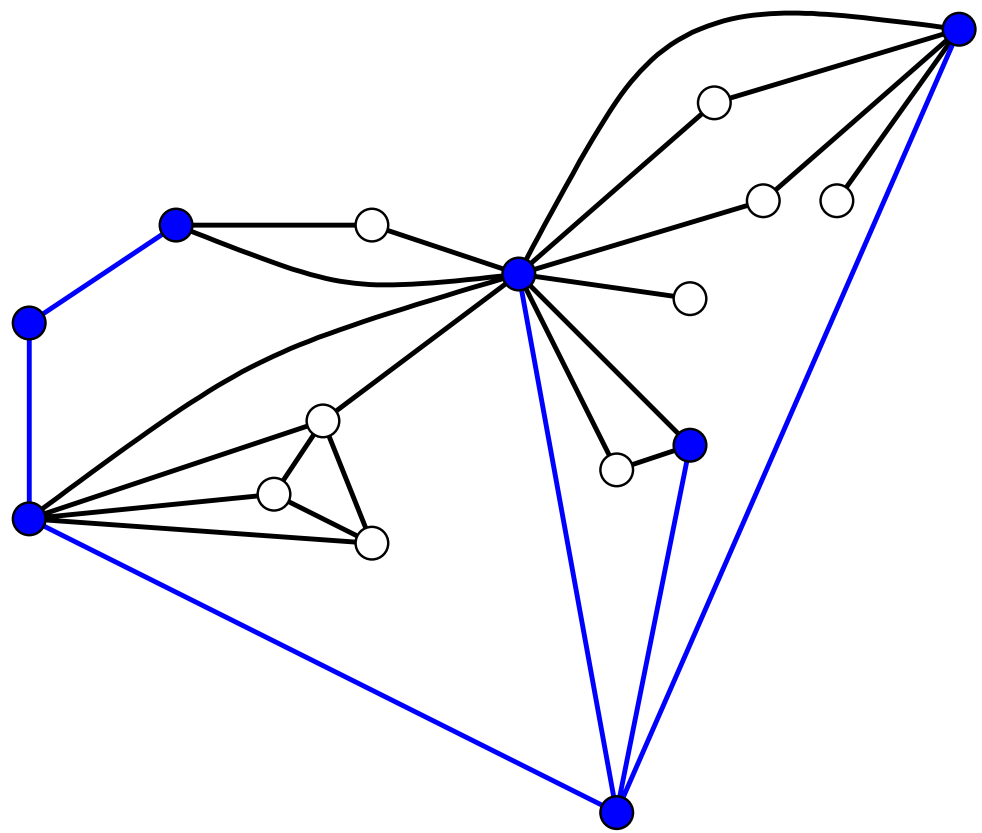


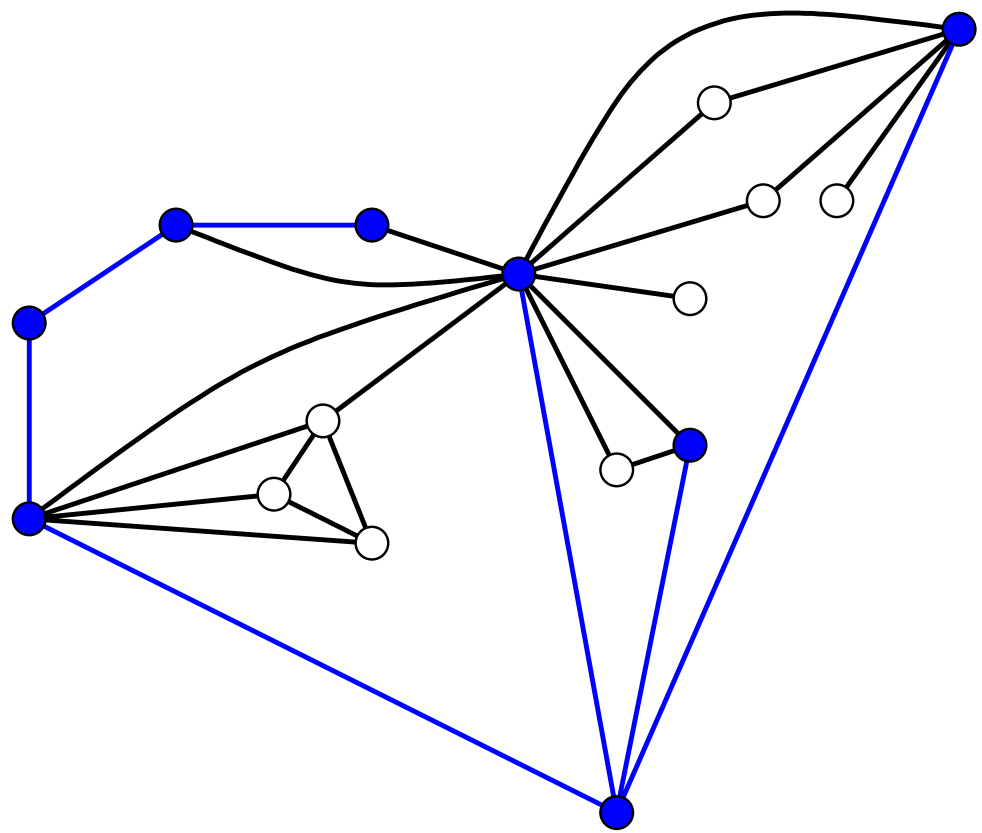




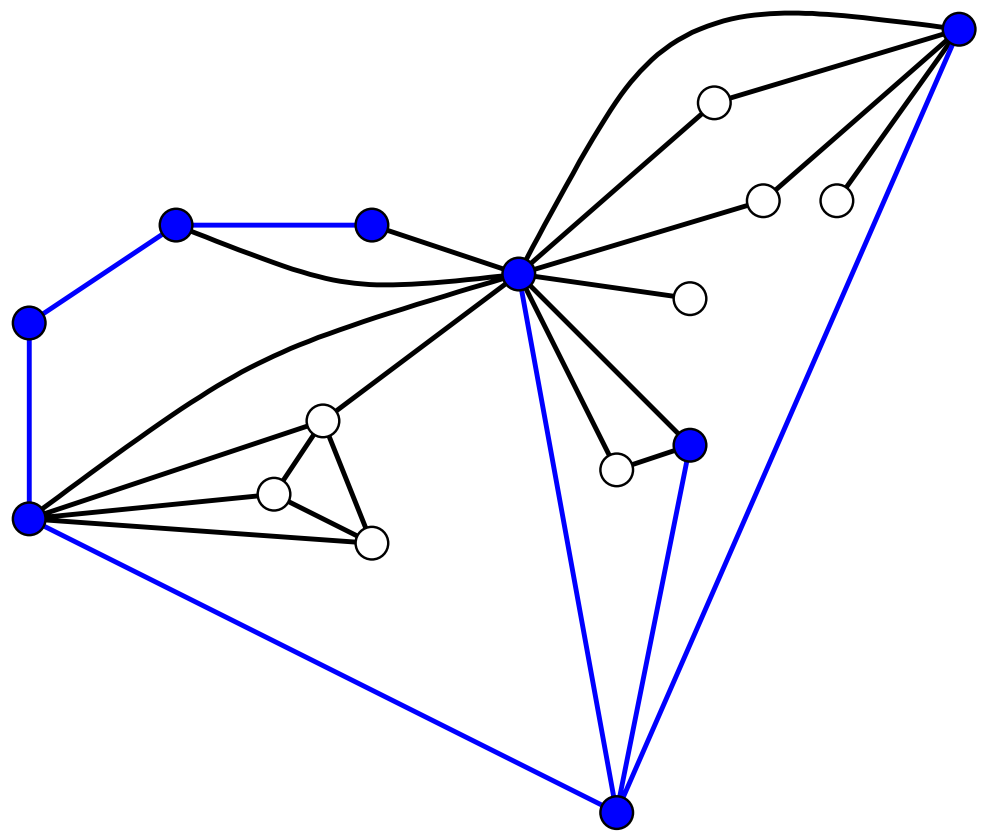


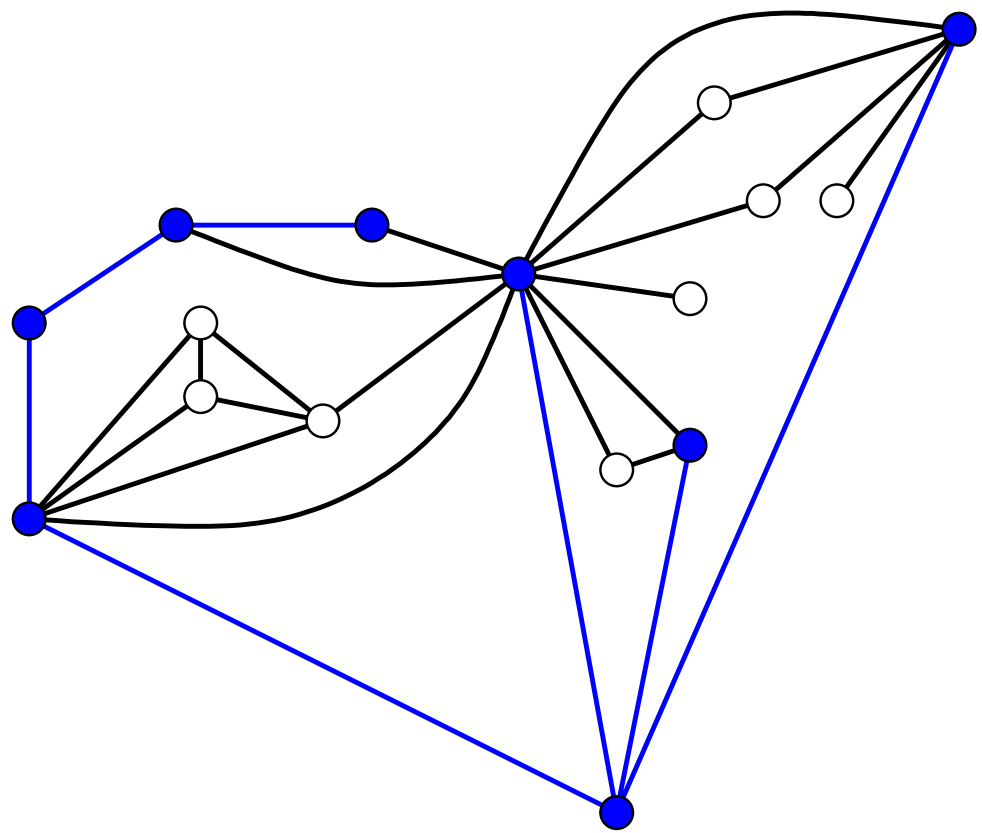


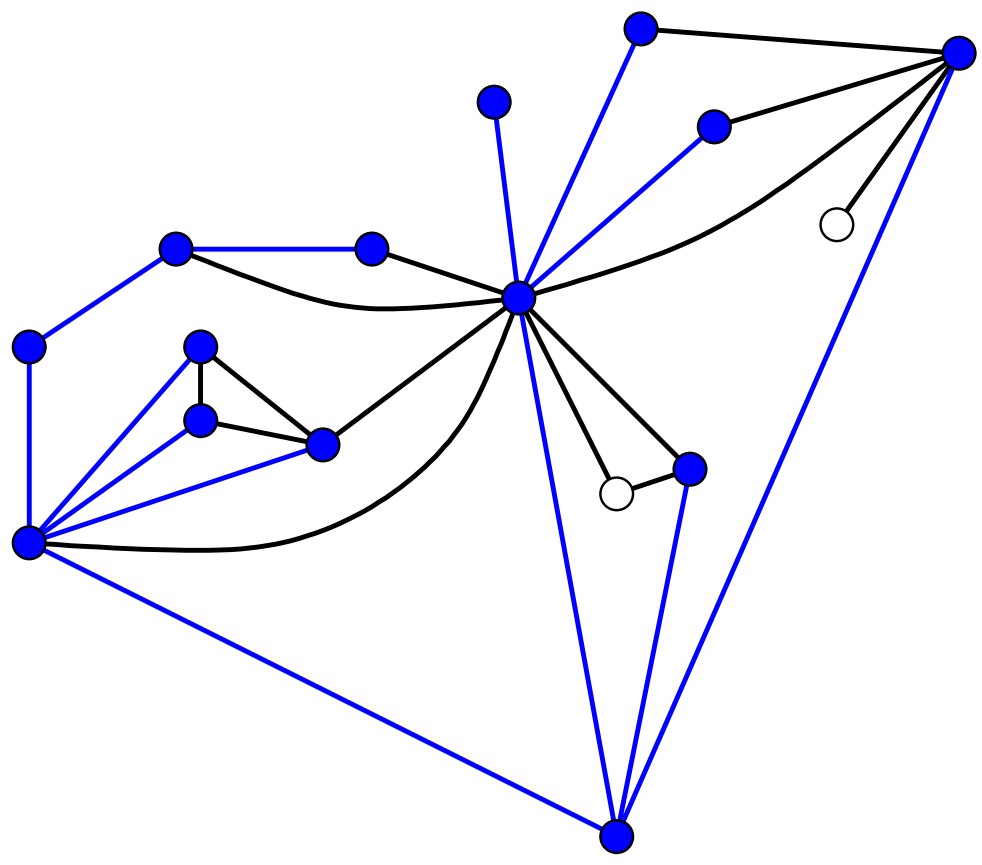


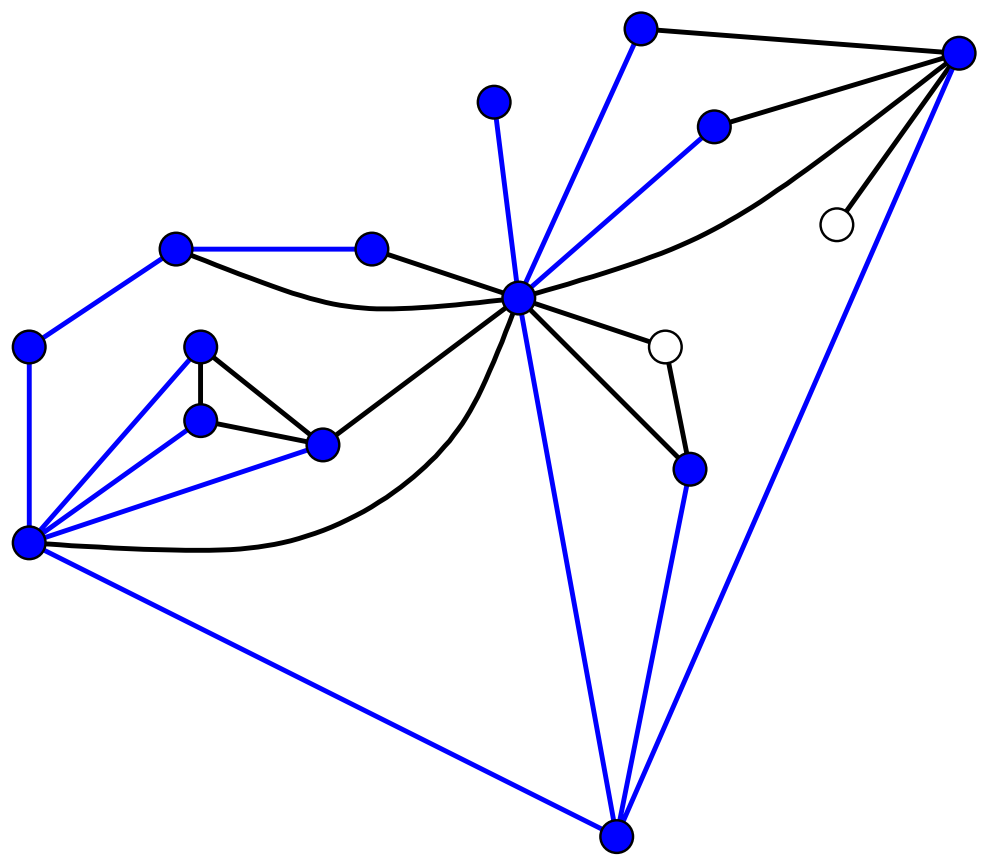


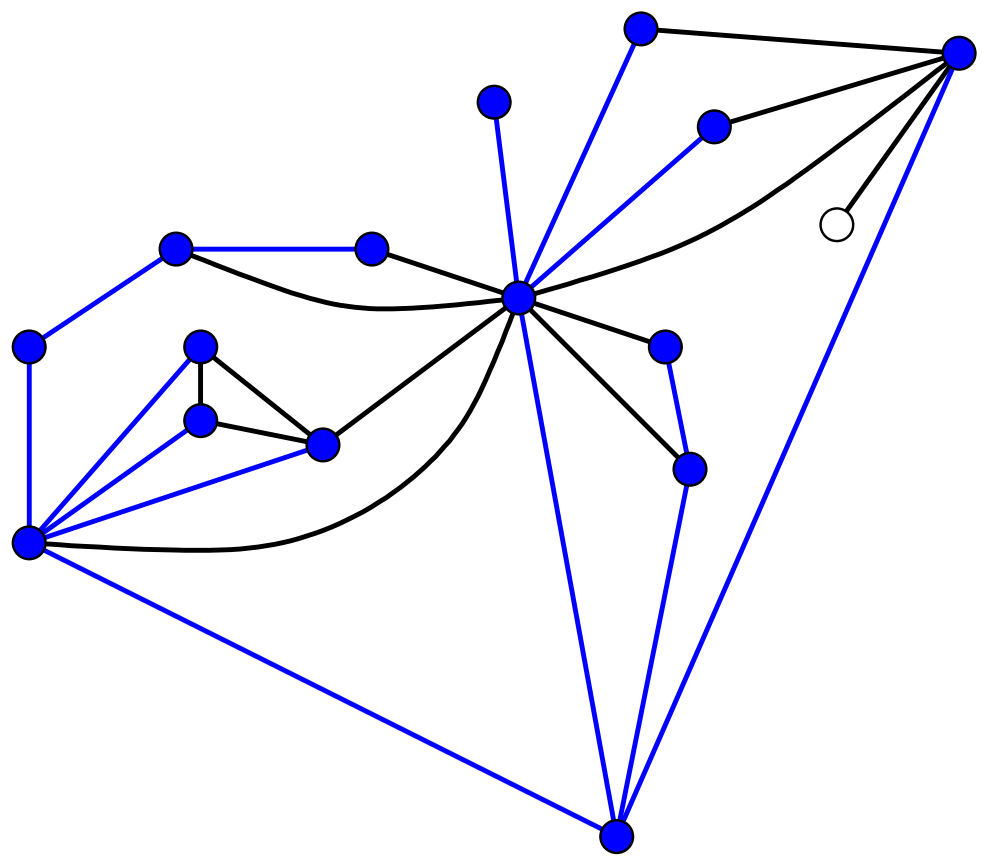
Observation. Il existe toujours une carte M' de G possédant un arbre plus-à-gauche T' qui couvre tous les sommets de G .

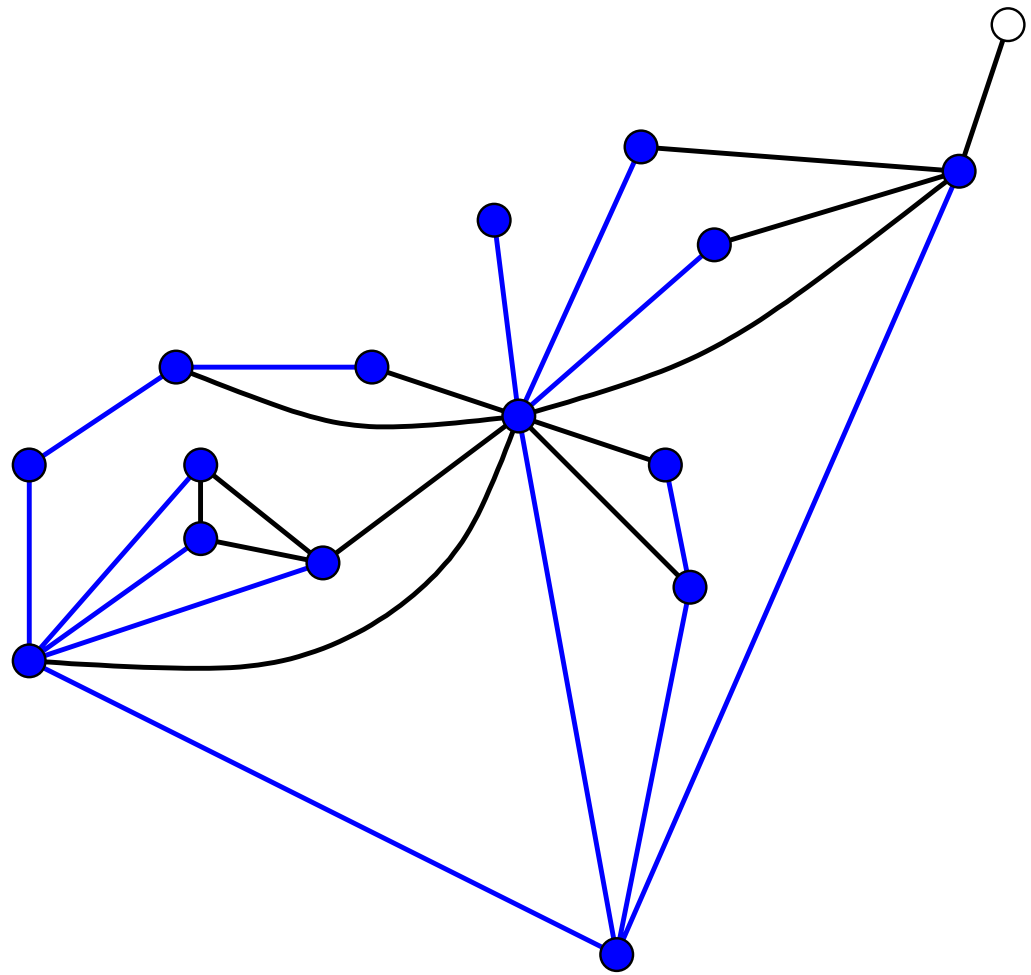


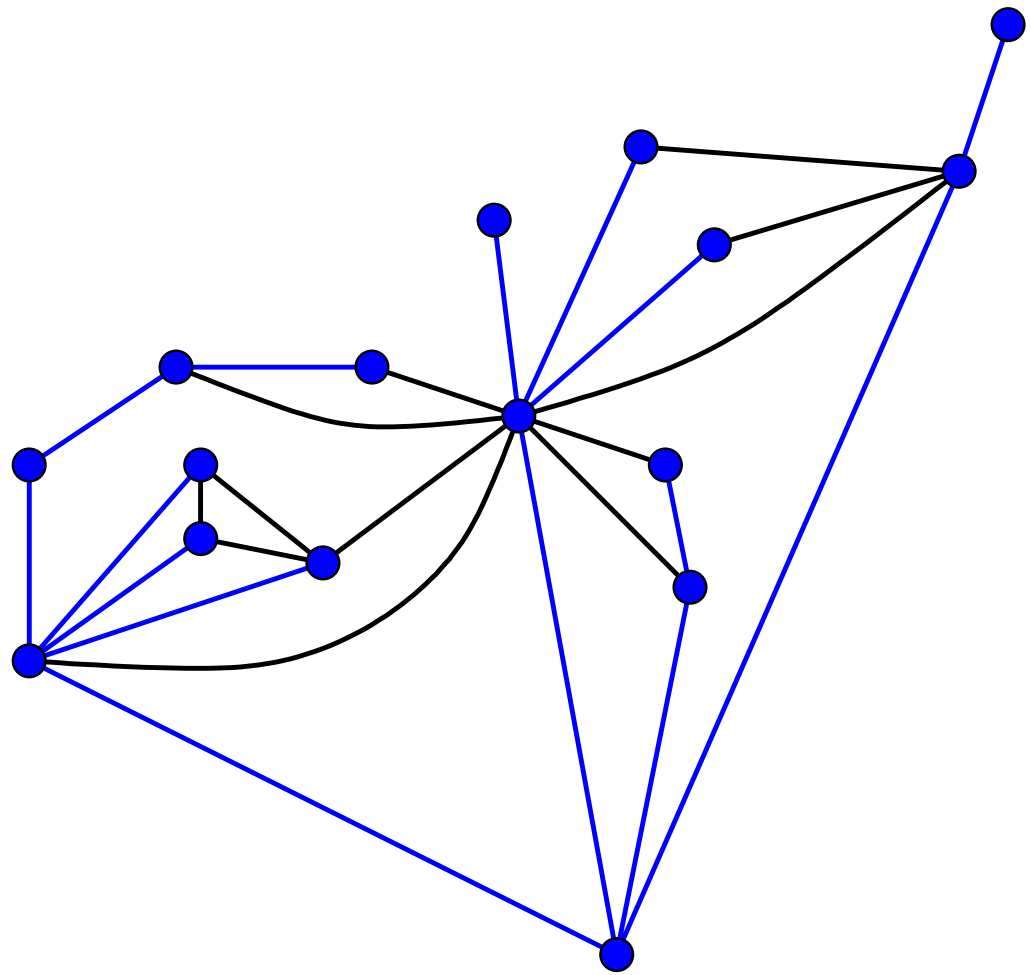






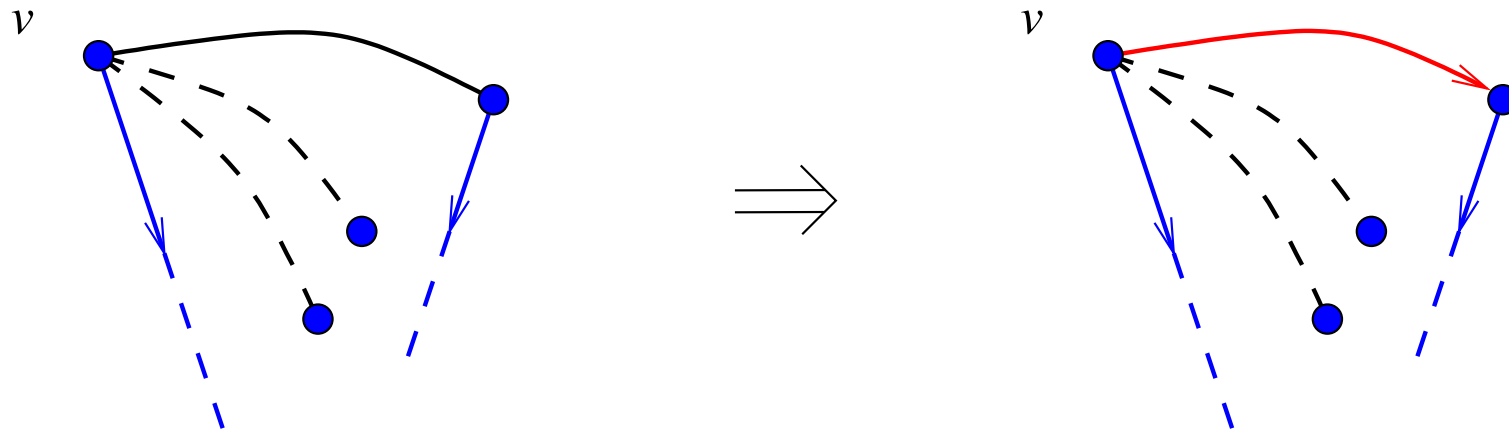






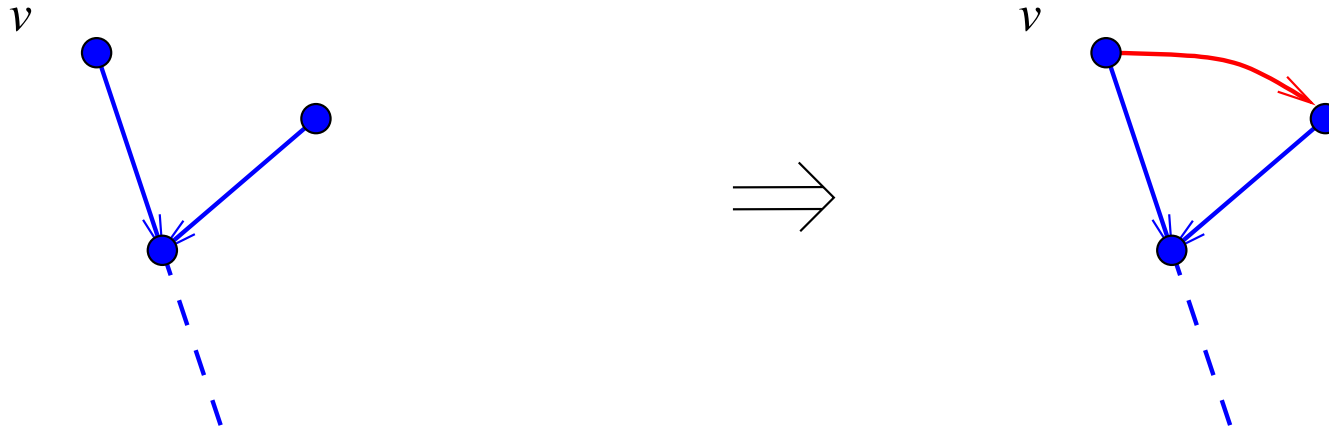
Arêtes rouges : 3 cas

Cas 1: v a un premier voisin à "droit"



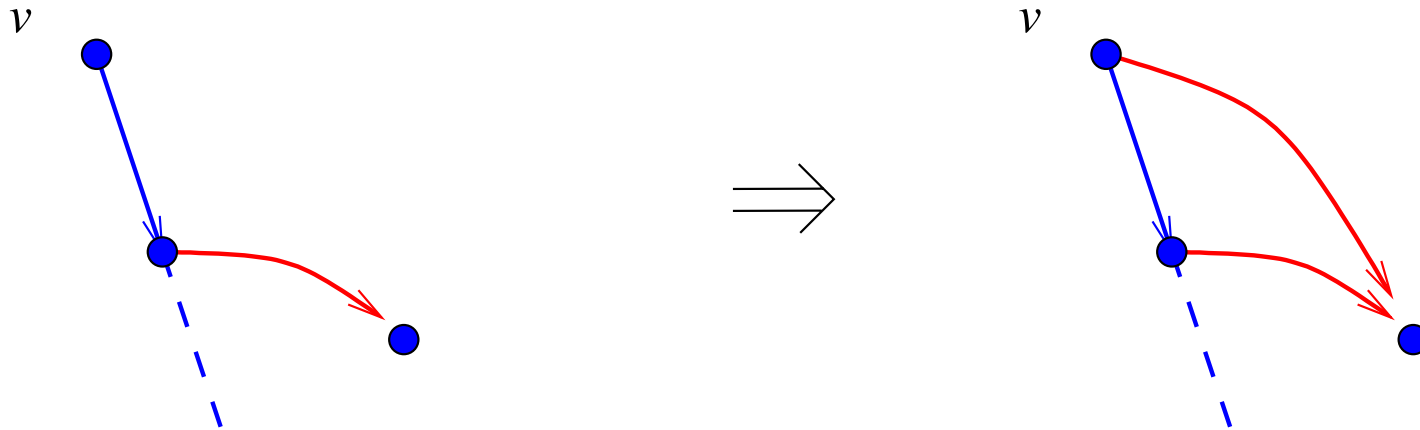
Arêtes rouges : 3 cas

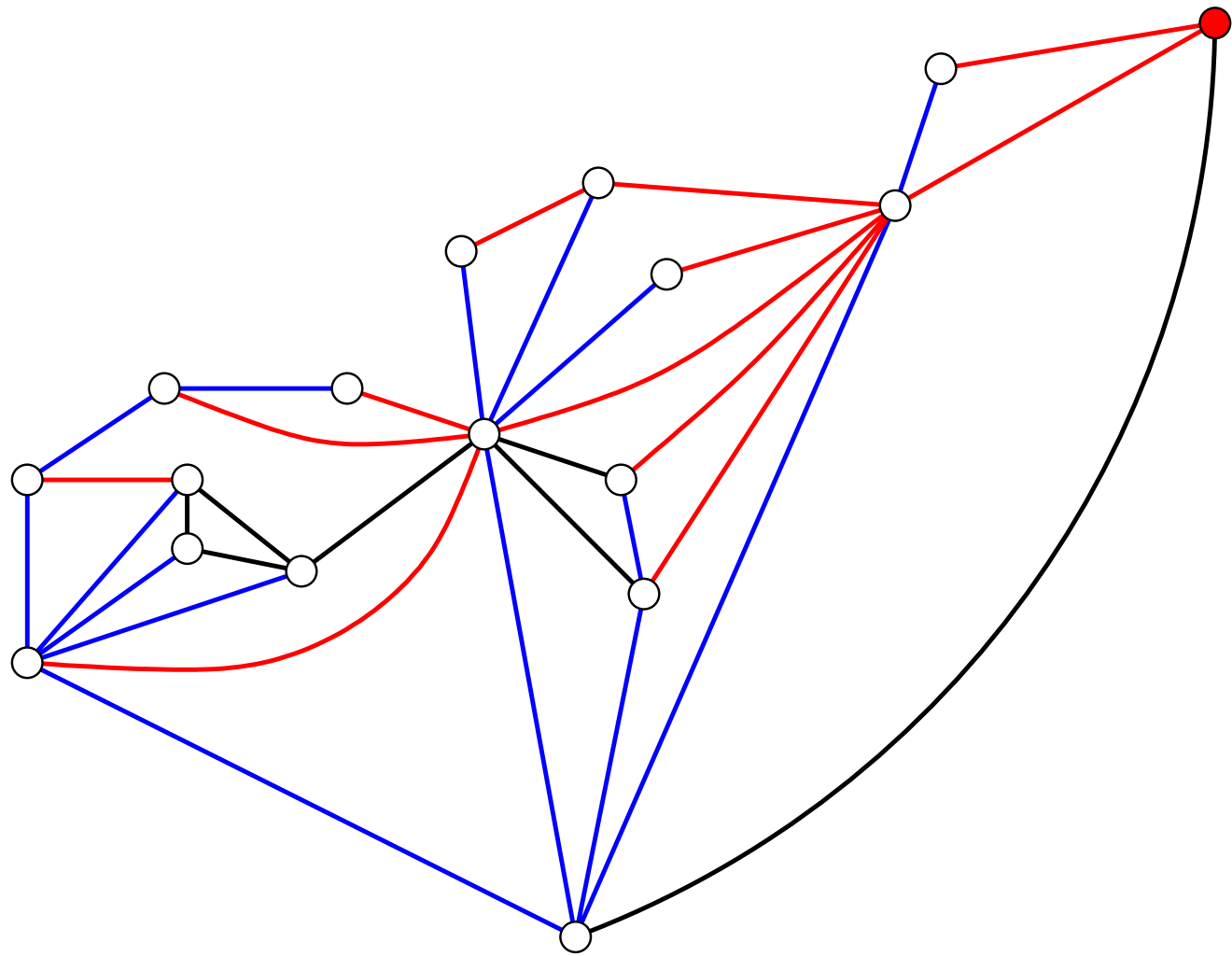
Sinon, **Cas 2:** v a un frère à "droit"

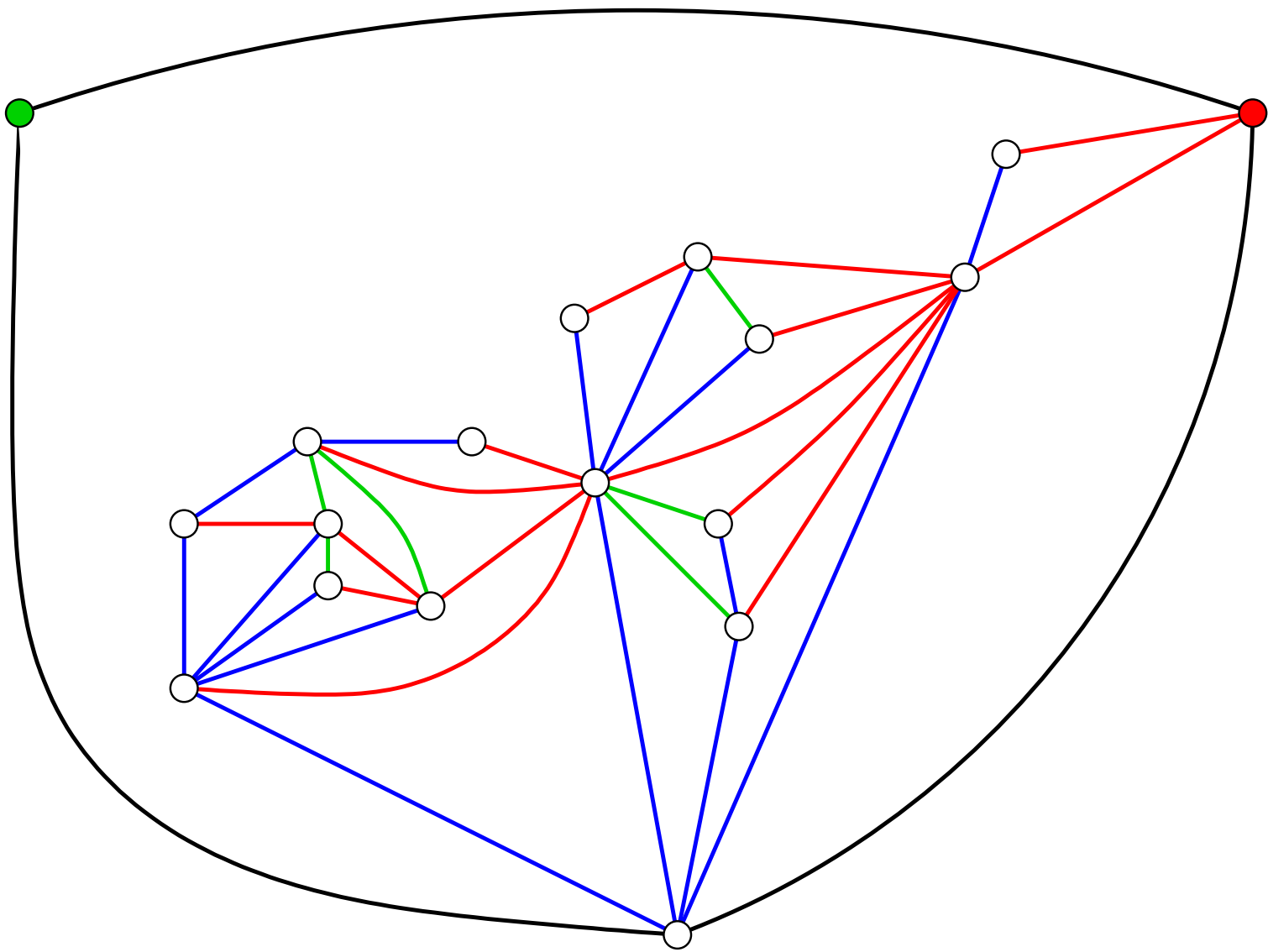


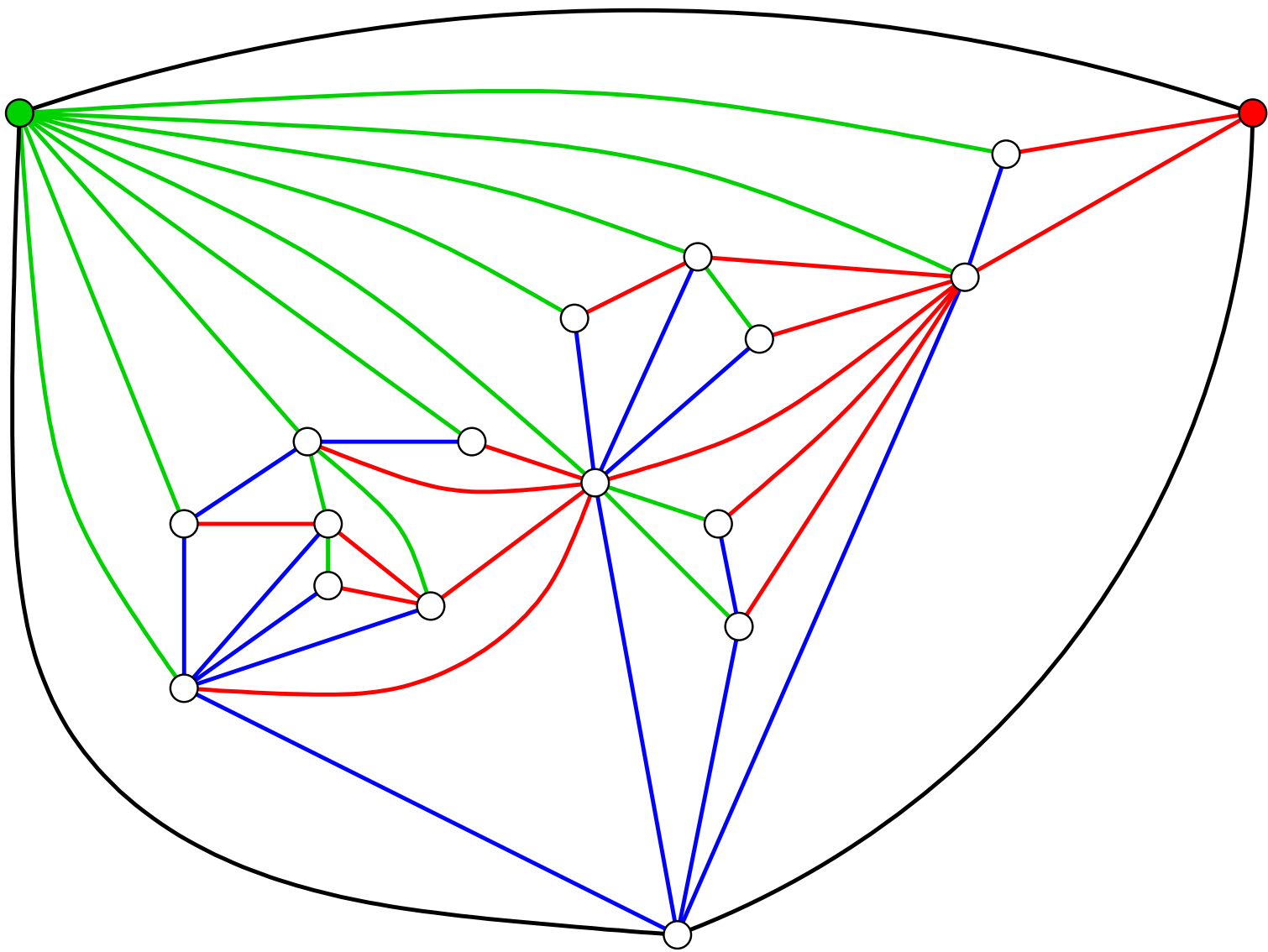
Arêtes rouges : 3 cas

Sinon, **Cas 3**: $p_1(v) = p_1(p_0(v))$.





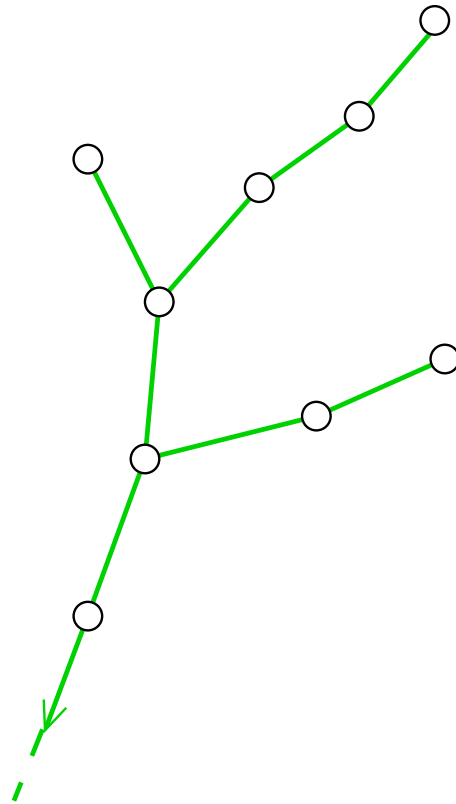




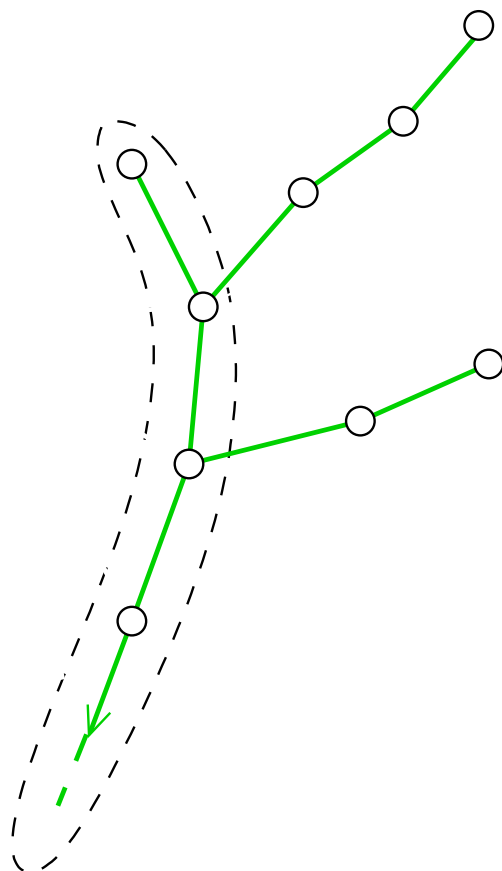
Partie III

Codage

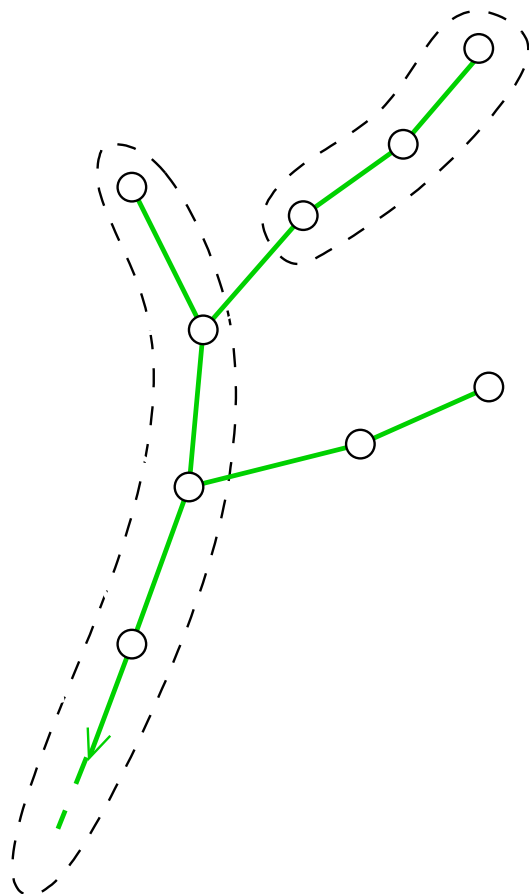
Branches (déf. & propriété)



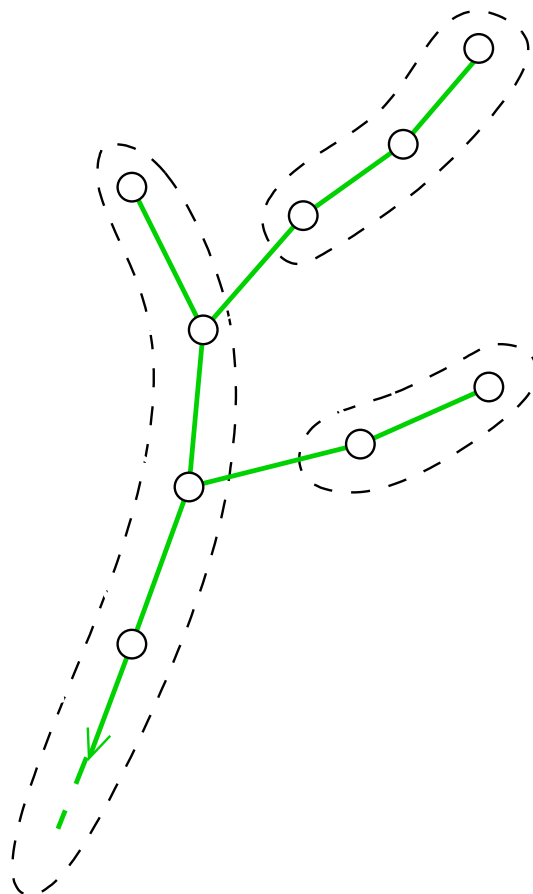
Branches (déf. & propriété)



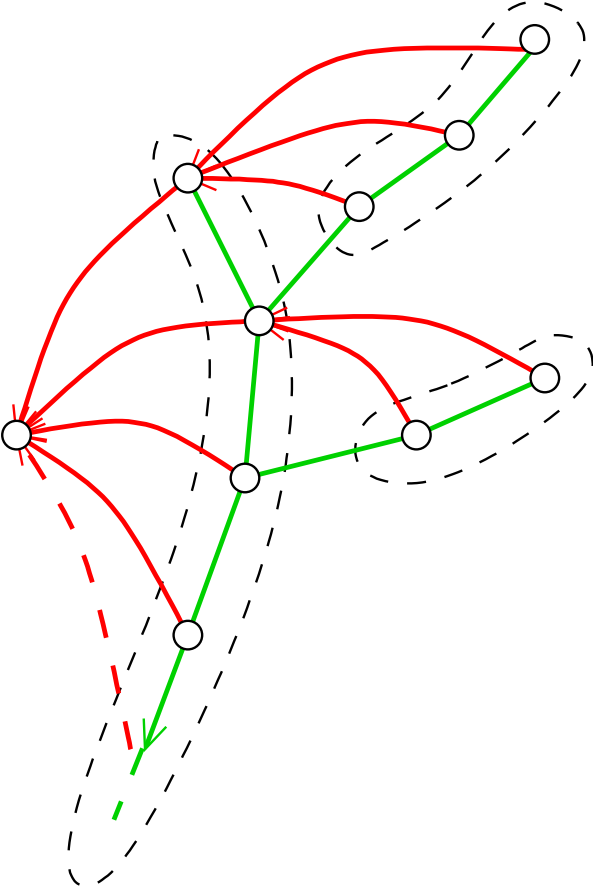
Branches (déf. & propriété)



Branches (déf. & propriété)



Branches (déf. & propriété)



Arêtes utiles, arêtes manquantes

Soit $S = (T_0, T_1, T_2)$ une super-triangulation de G . On pose:

1) l'ensemble des arêtes "utiles" :

$$U_S := \{(v, p_1(v)) \mid v \text{ feuille de } T_2\}$$

Arêtes utiles, arêtes manquantes

Soit $S = (T_0, T_1, T_2)$ une super-triangulation de G . On pose:

1) l'ensemble des arêtes "utiles" :

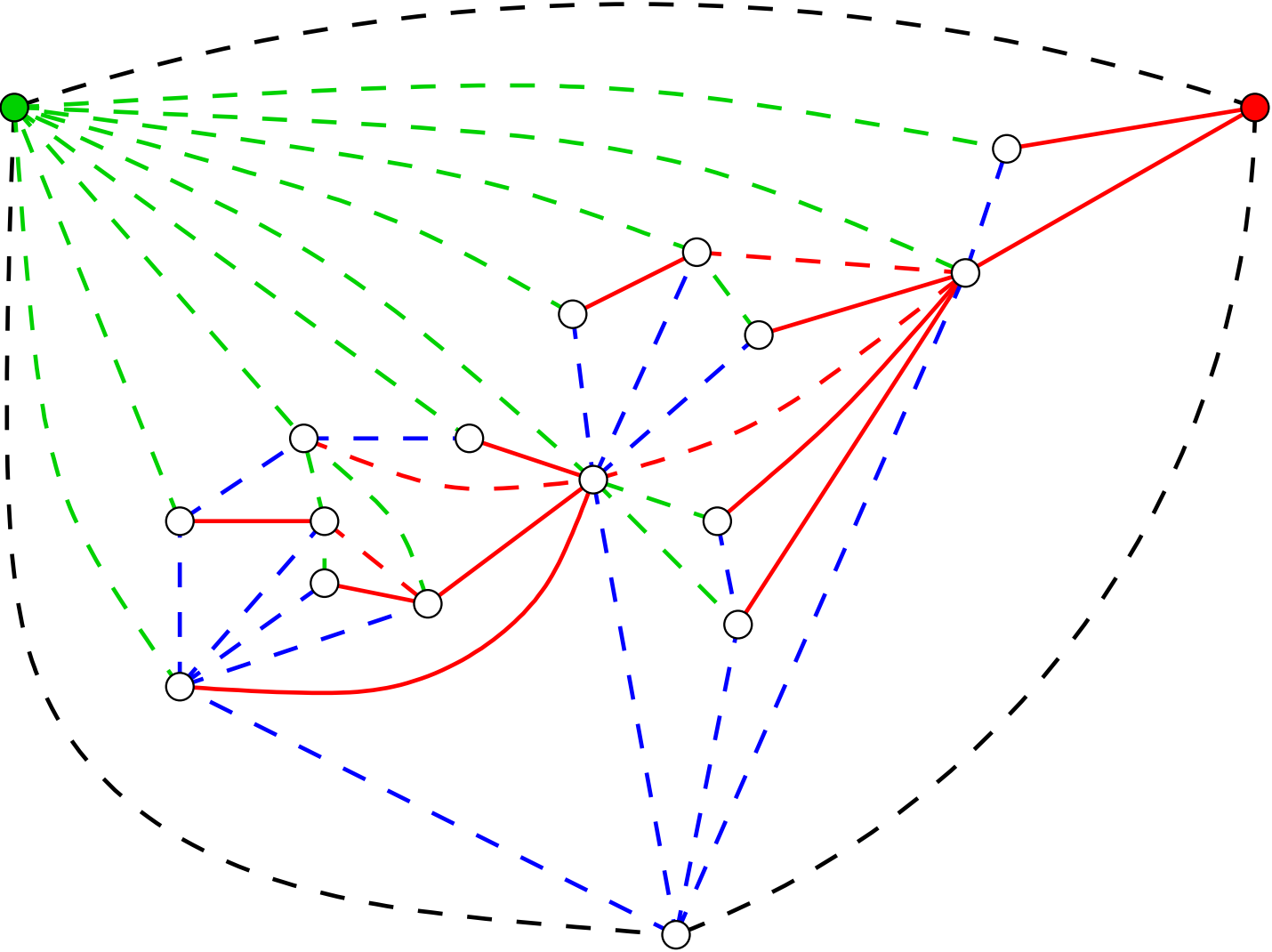
$$U_S := \{(v, p_1(v)) \mid v \text{ feuille de } T_2\}$$

2) l'ensemble des arêtes "manquantes" :

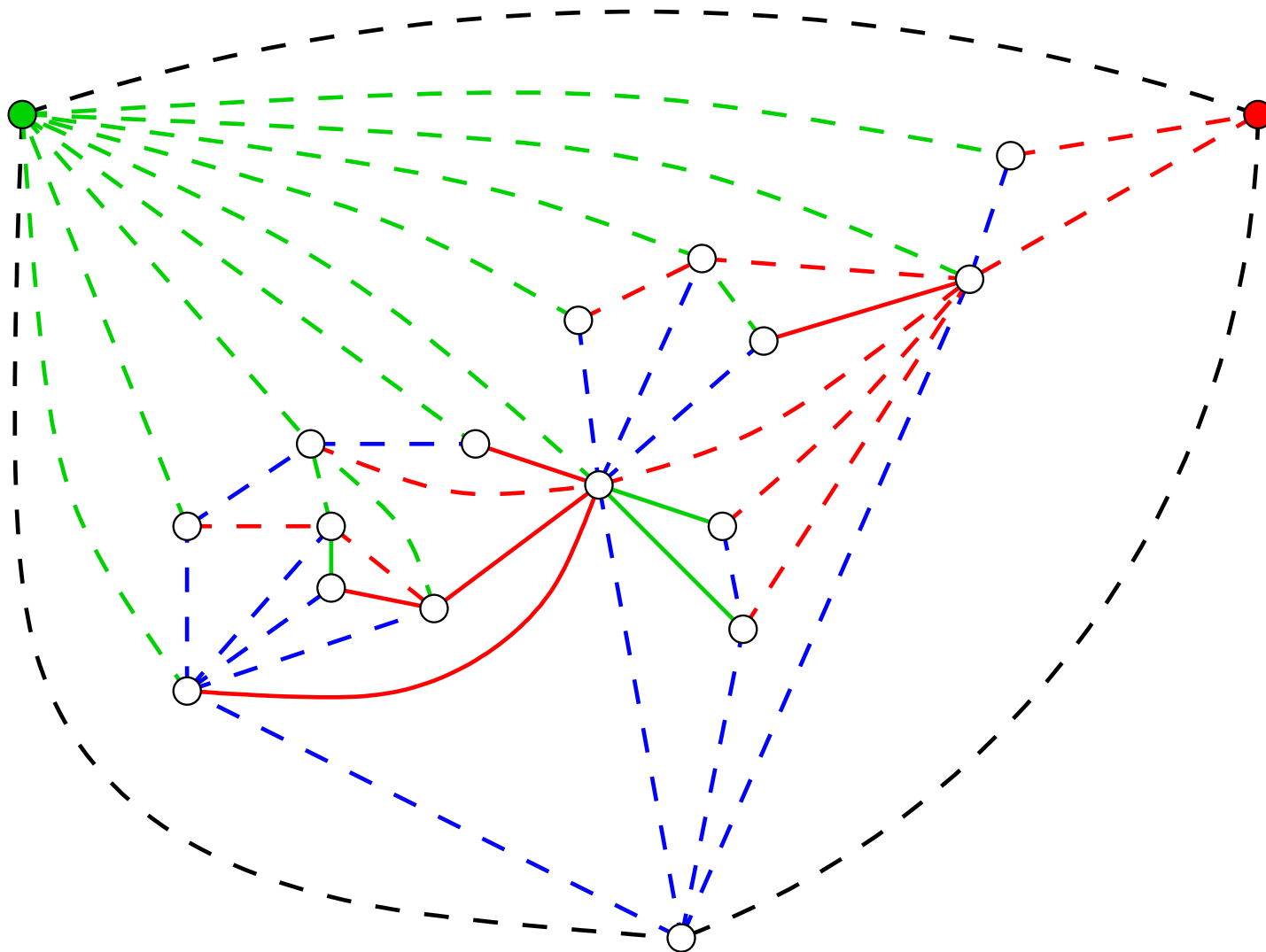
$$M_S := E(G) \setminus E(T_0) \setminus \{(v, p_1(v)) \mid v \text{ nœud interne de } T_2\}$$

Remarque. $M_S \subset E(T_2) \cup \{(v, p_1(v)) \mid v \text{ feuille de } T_2\}$

Arêtes utiles

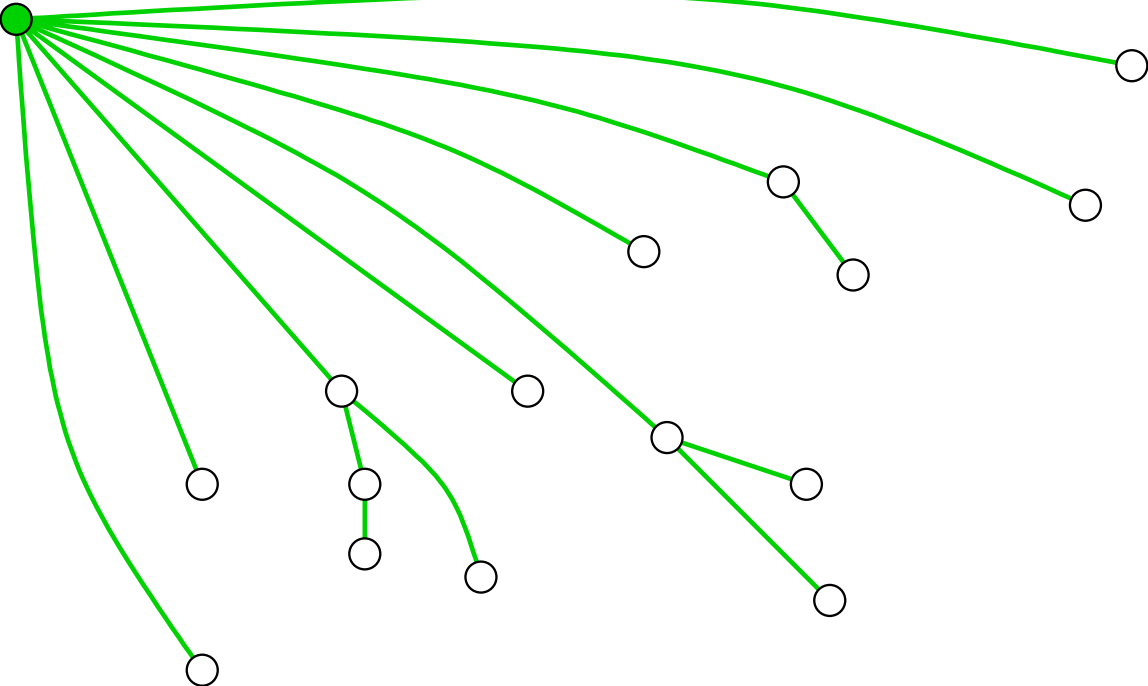


Arêtes manquantes

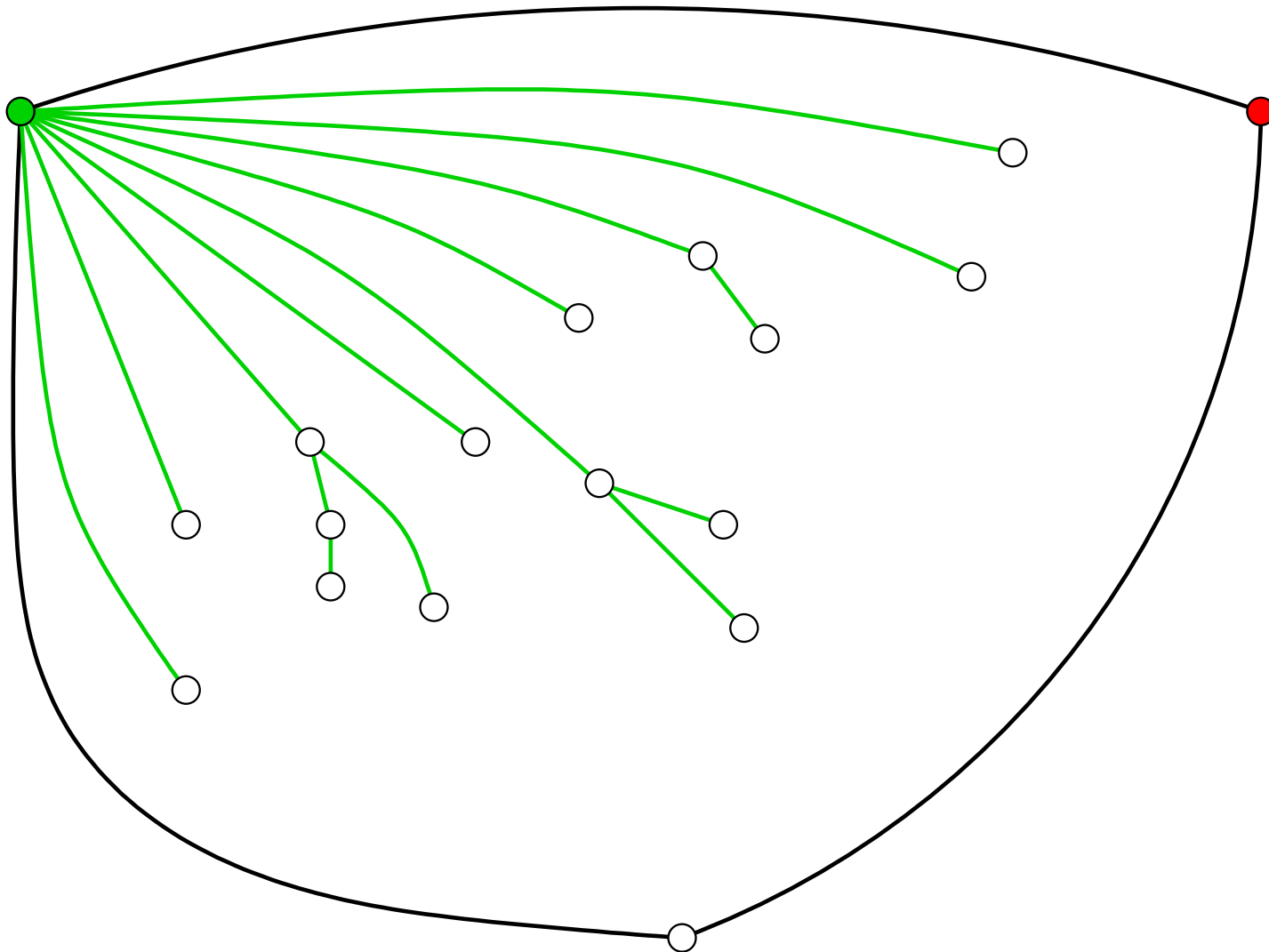


Théorème. Si $S = (T_0, T_1, T_2)$ est une super-triangulation de G , alors (T_2, U_S, M_S) détermine de manière unique G .

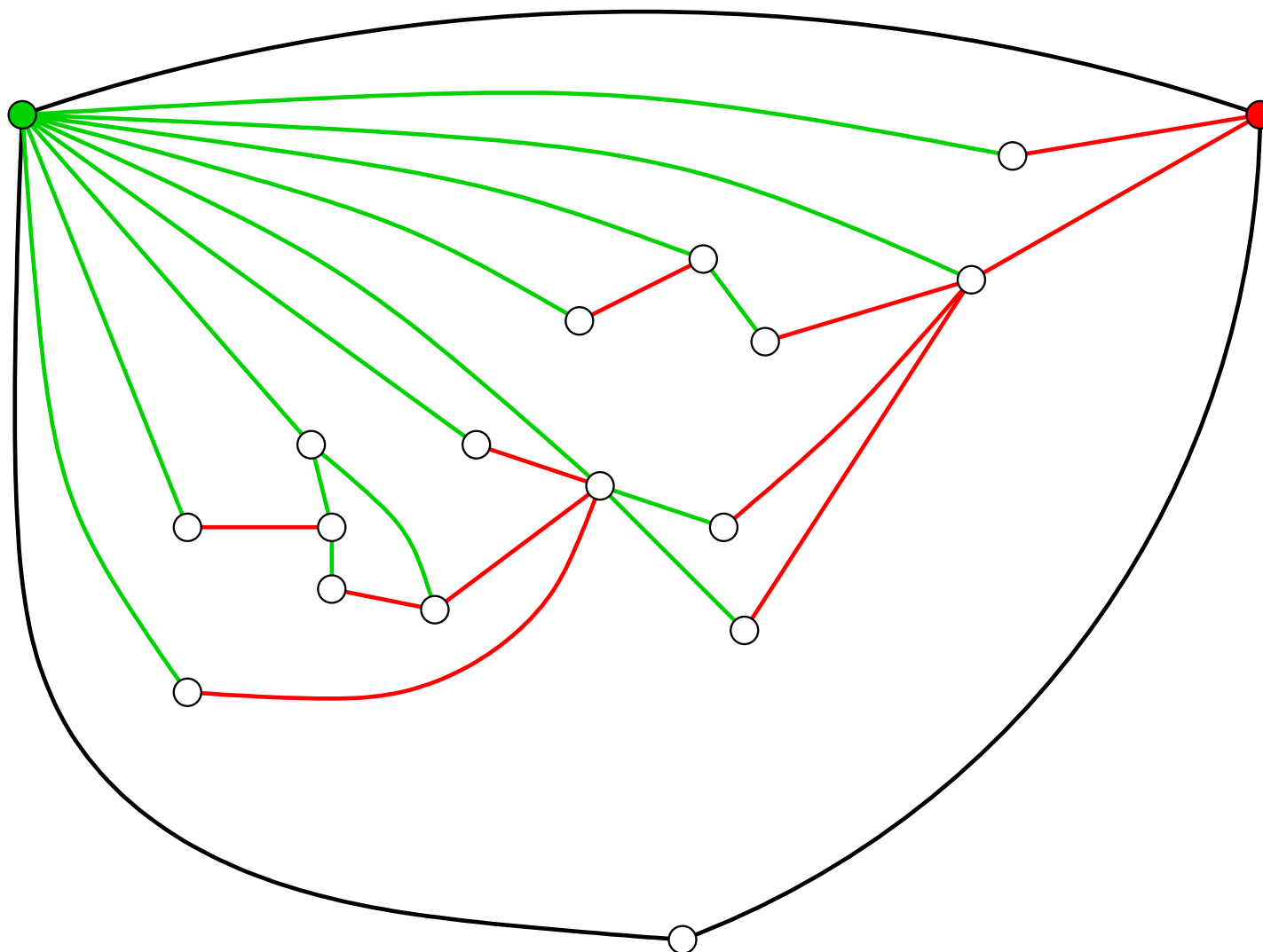
Arbre T_2



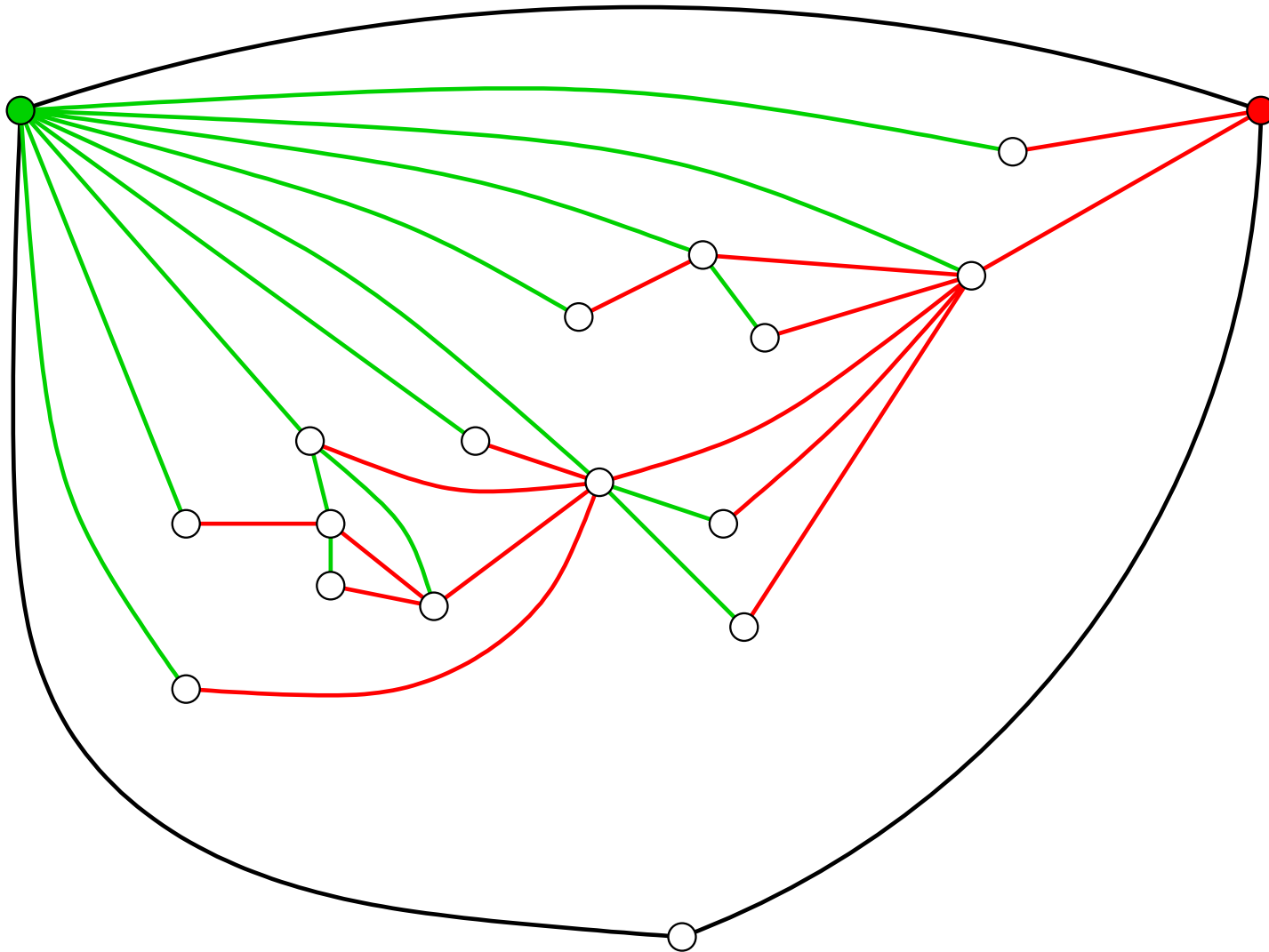
+ face extérieure



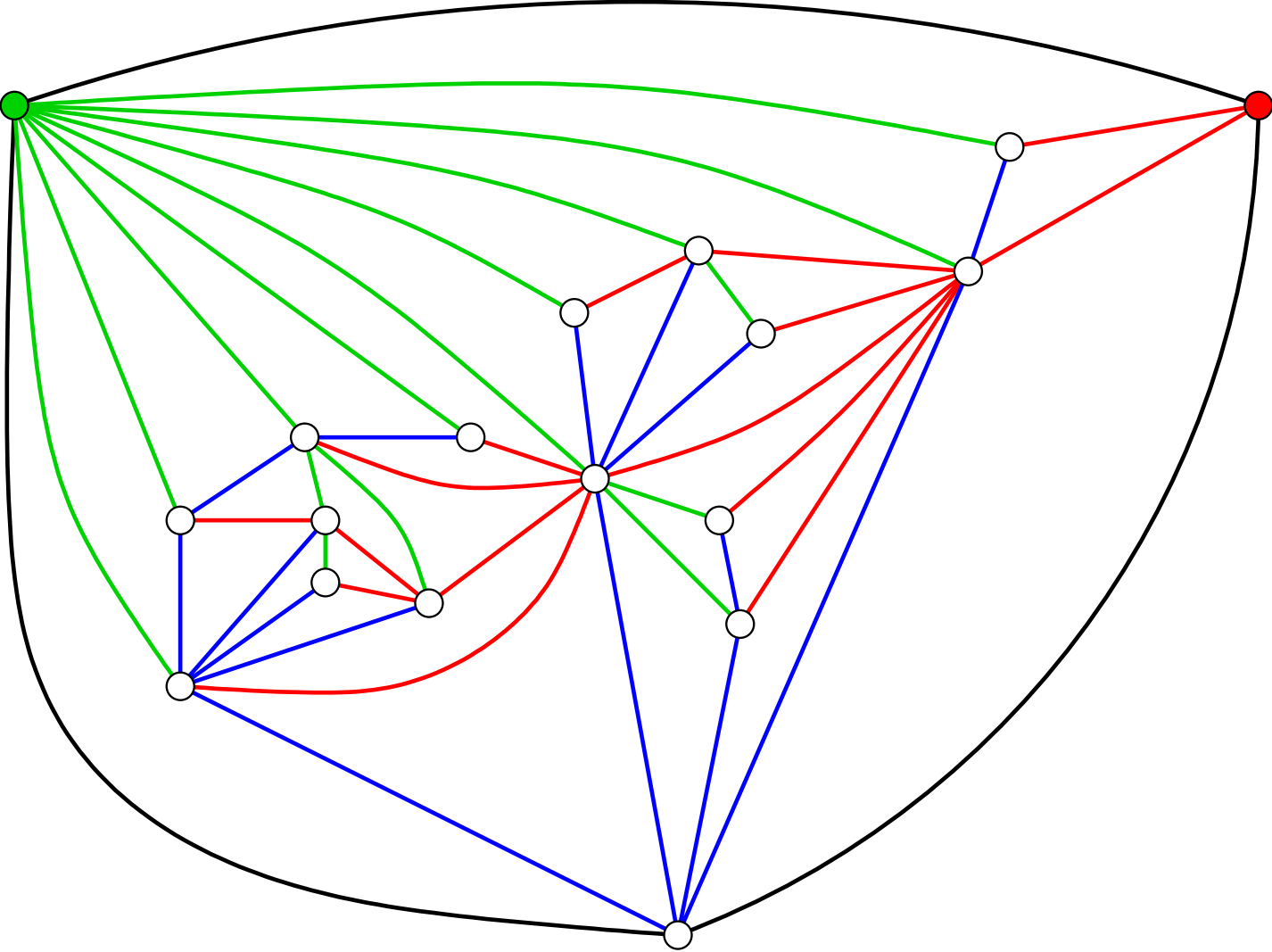
+ arêtes utiles



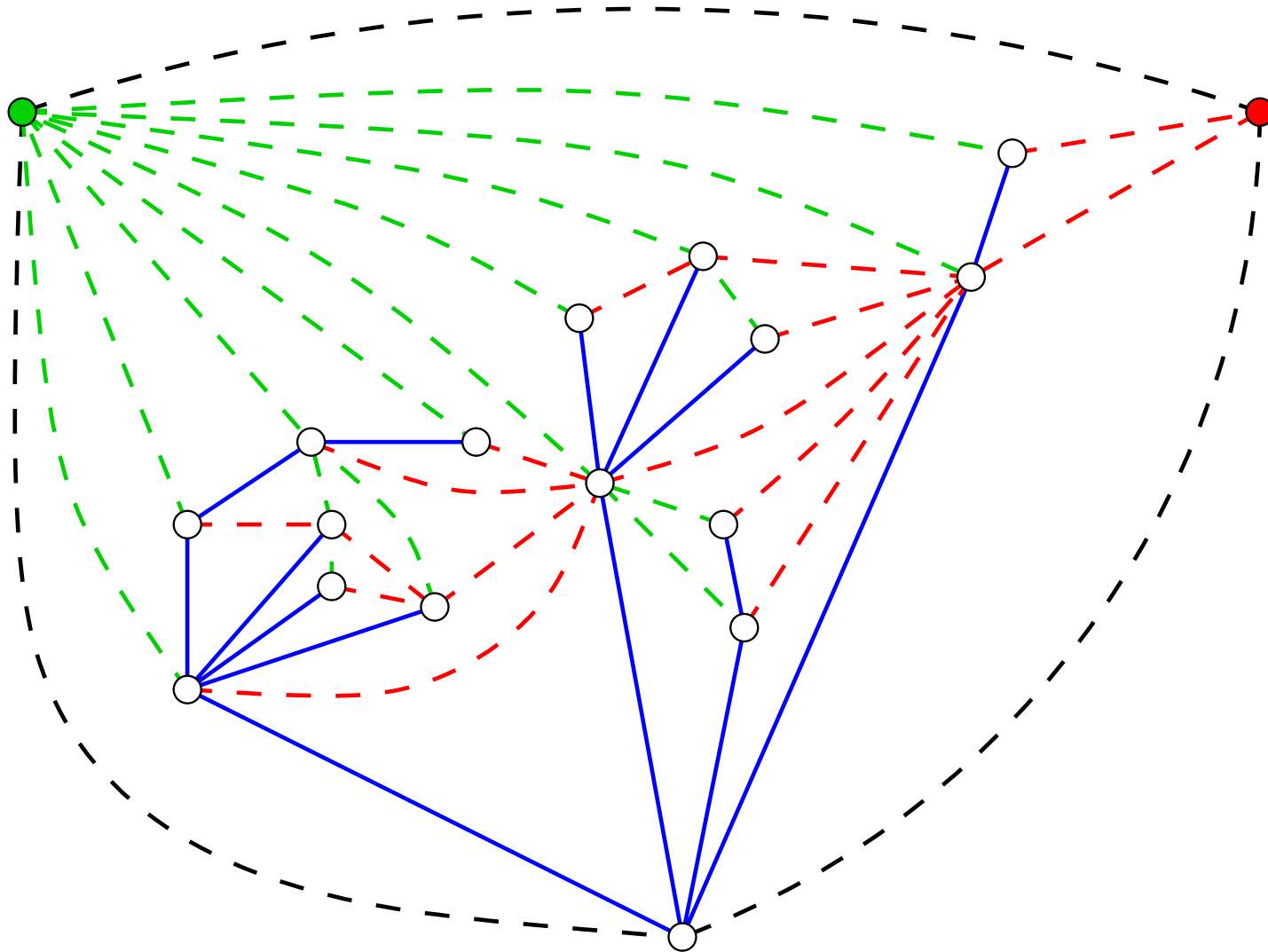
⇒ toutes les arêtes rouges, donc T_1



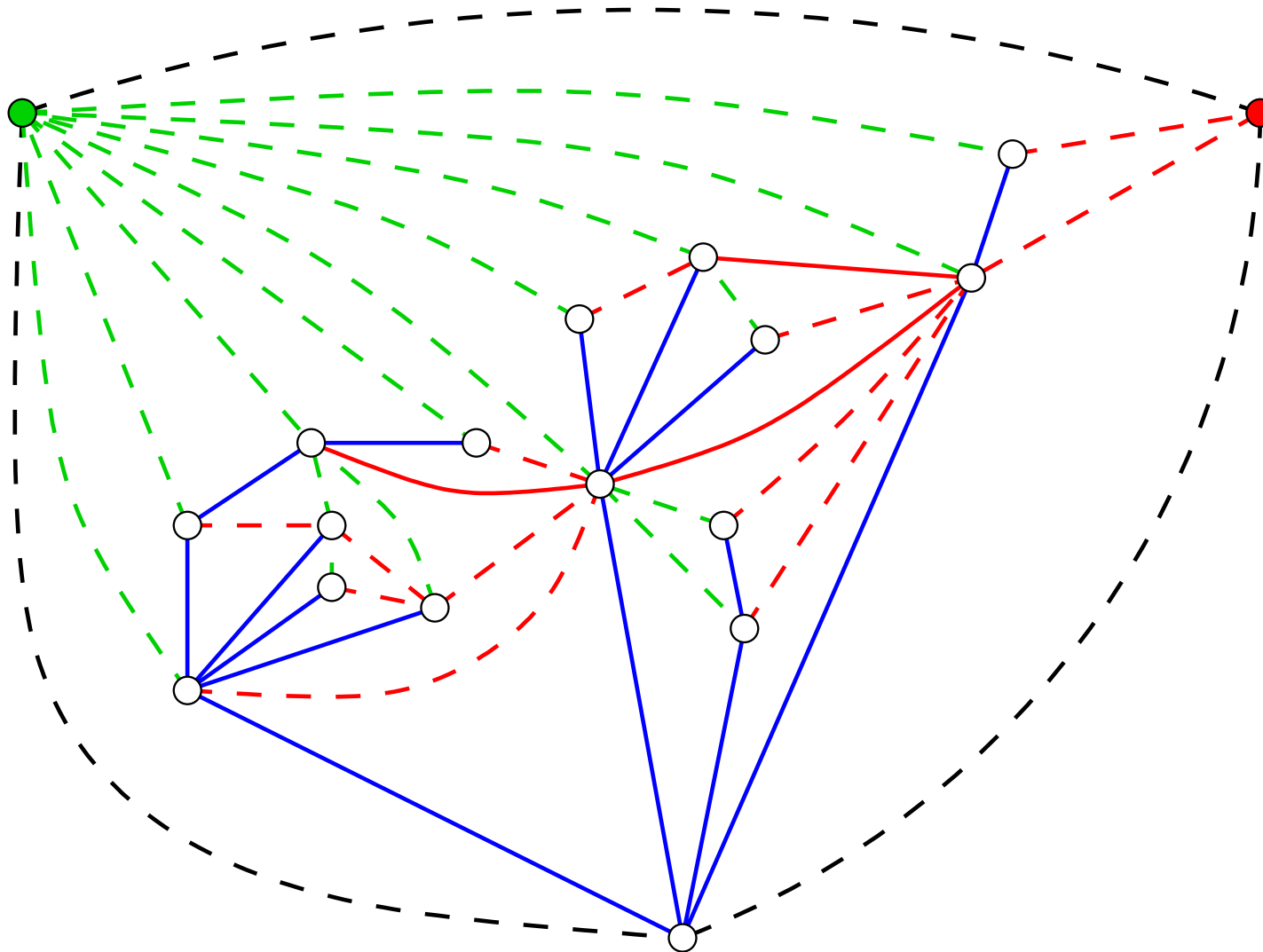
$T_2 + T_1 \Rightarrow T_0$, donc $\Rightarrow S$



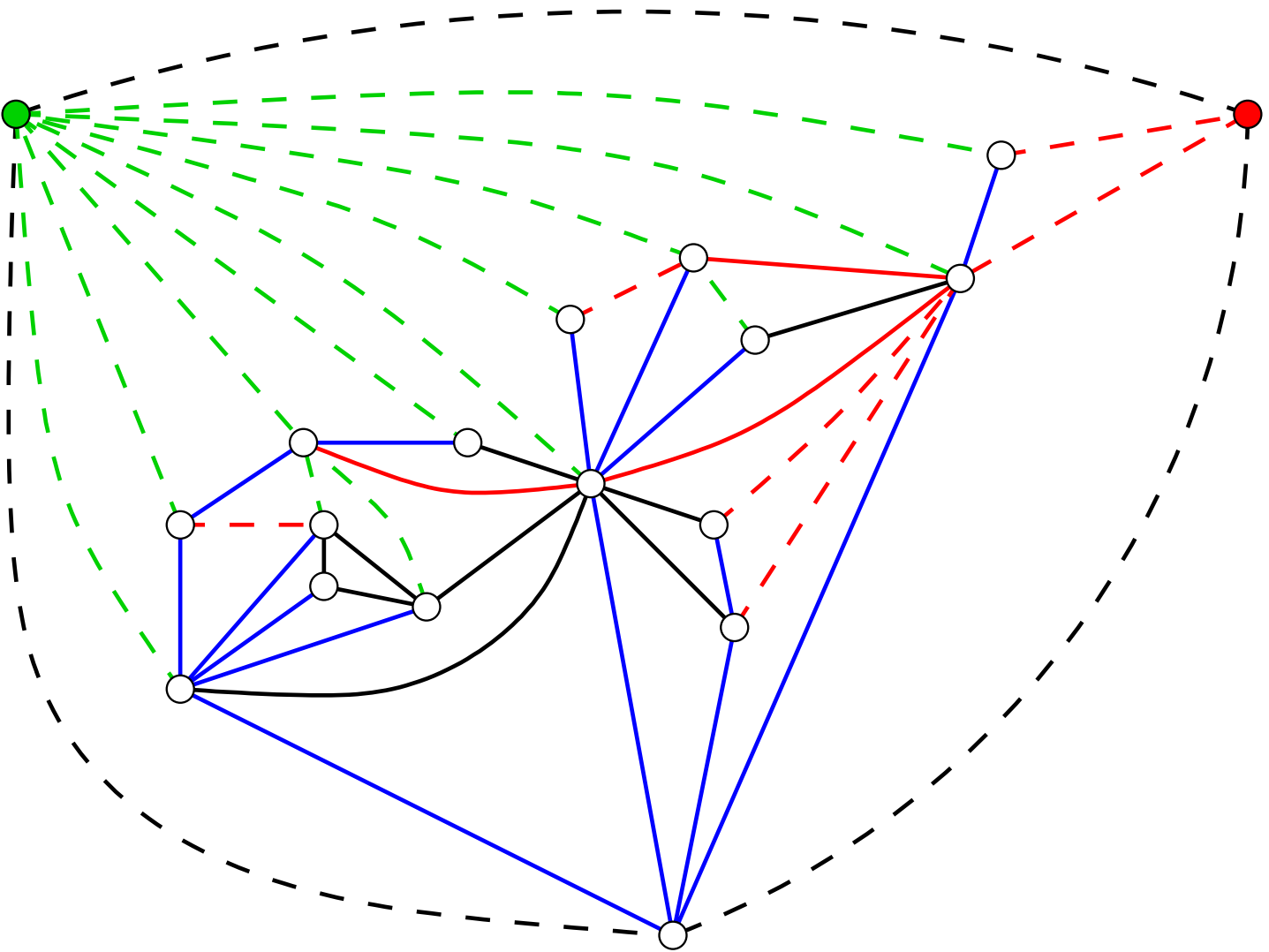
Les arêtes de T_0 sont dans G



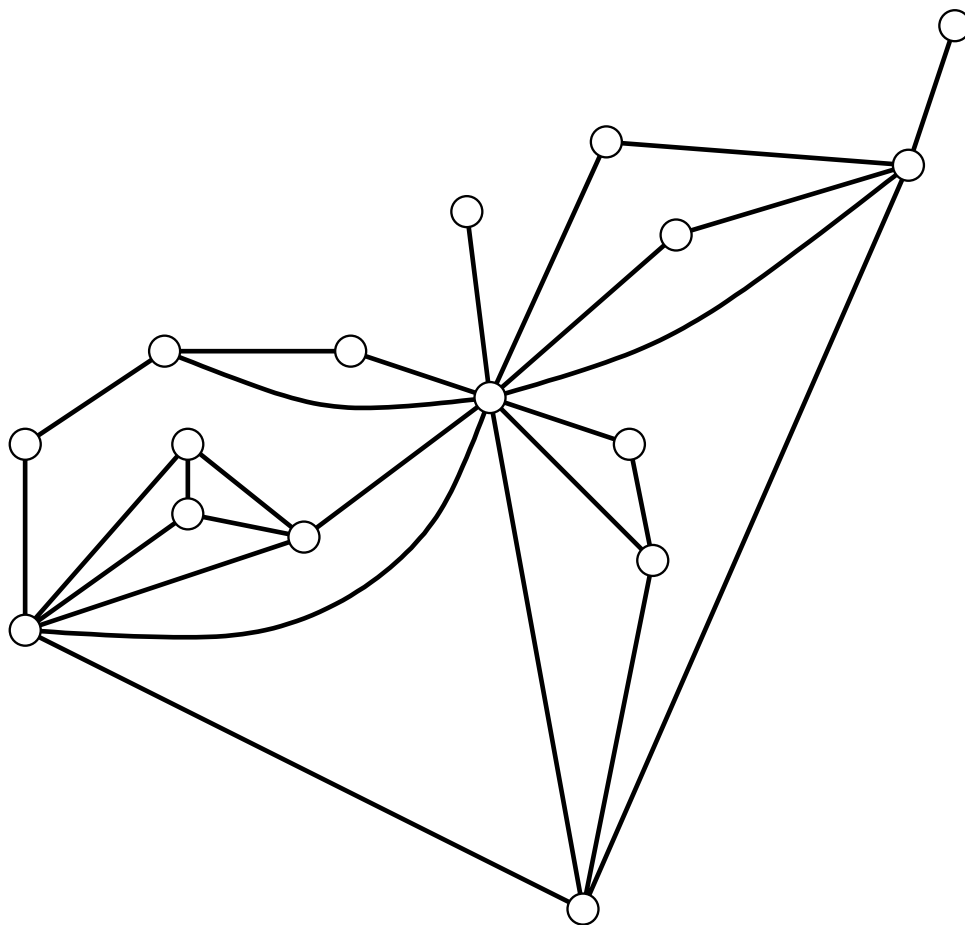
Les arêtes $(v, p_1(v))$, v nœud interne de T_2 , sont dans G



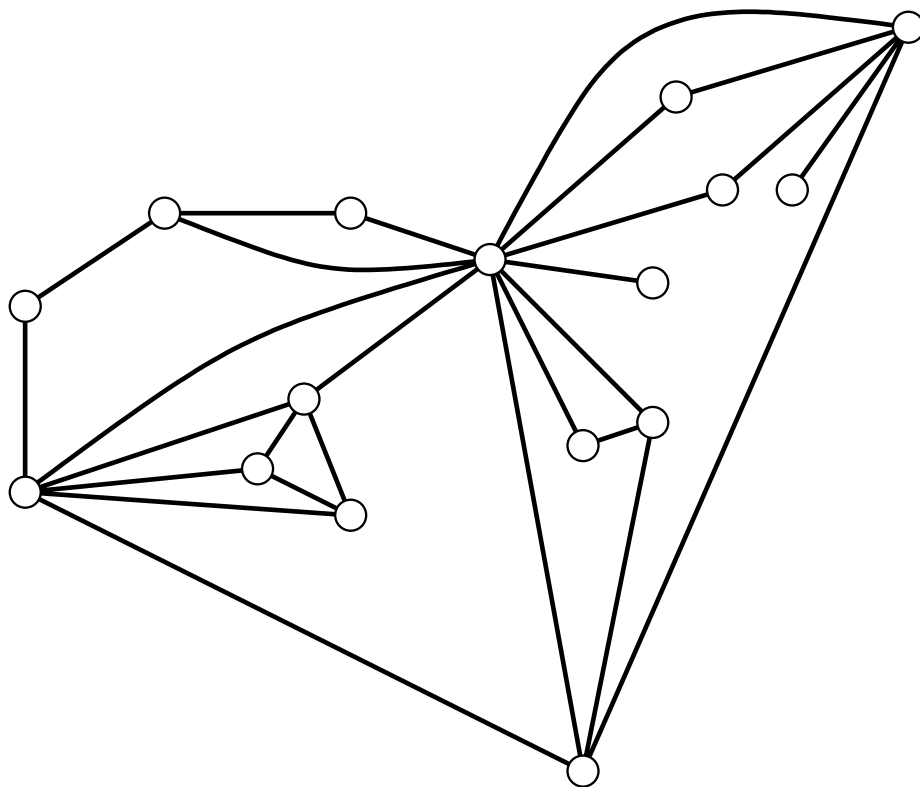
et les arêtes manquantes




C'est G ! (une carte de G)

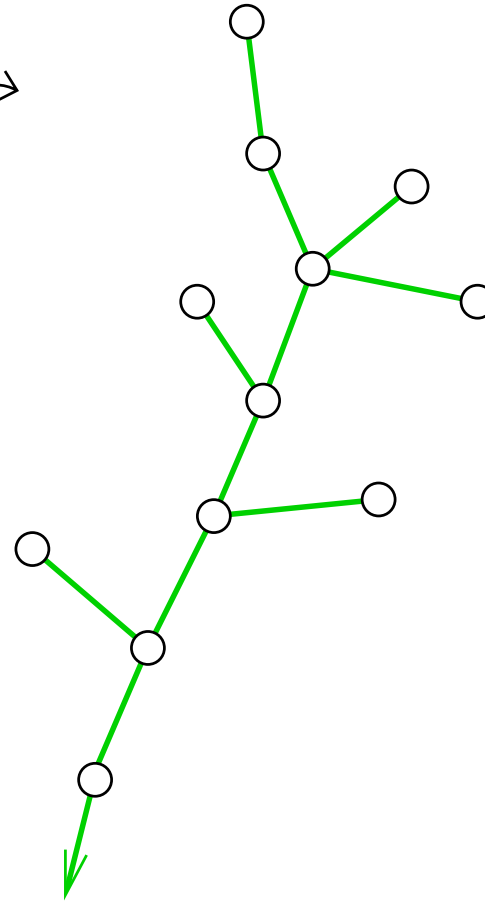


[Dessin d'origine de G]




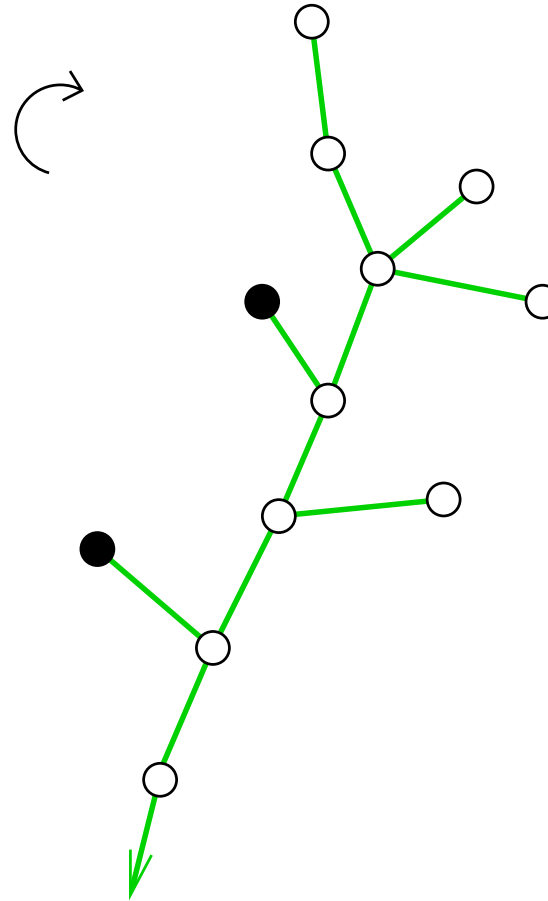
Bourgeons (déf. & propriété)

1 bourgeon = 
= 1 1 0 1




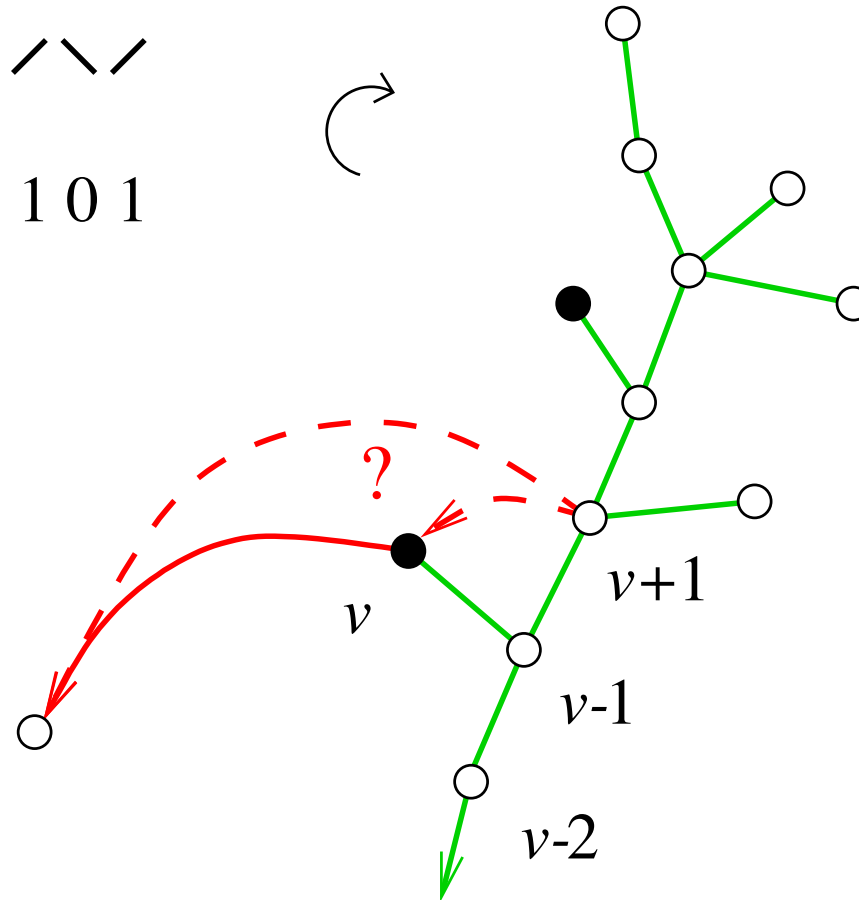
Bourgeons (déf. & propriété)

1 bourgeon = 
= 1 1 0 1



Bourgeons (déf. & propriété)

1 bourgeon = 
= 1 1 0 1



v bourgeon \Rightarrow $p_1(v+1) \in \{v, p_1(v)\}$

4. Comptage

(T_2) $A_n(f, b)$ = nombre d'arbres à n nœuds, f feuilles et b bourgeons

(U_S) $B_n(f, b)$ = nombre maximum d'ensembles U_S pour un T_2

(M_S) $C_n(f)$ = nombre maximum d'ensembles M_S pour une S

Donc,

$$q(n) \leq \sum_{0 \leq b < f < n} A_n(f, b) \cdot B_n(f, b) \cdot C_n(f)$$

$$q(n) \leq n^2 \cdot \max_{0 \leq b < f < n} \{A_n(f, b) \cdot B_n(f, b) \cdot C_n(f)\}$$

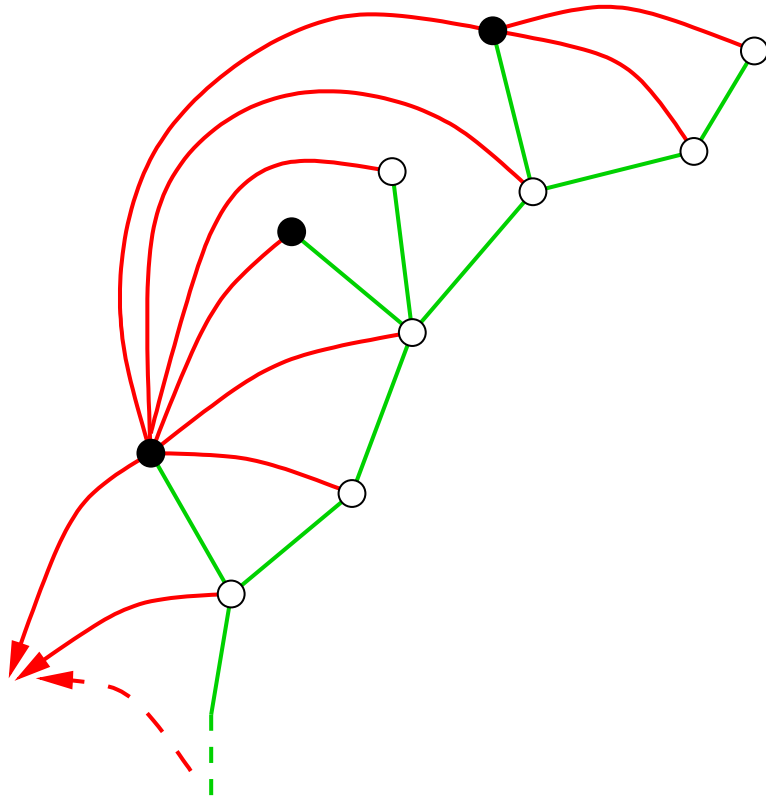
$C_n(f)$?
(arêtes manquantes connaissant S)

$$M_S \subset E(T_2) \cup \{(v, p_1(v)) \mid v \text{ feuille de } T_2\}$$

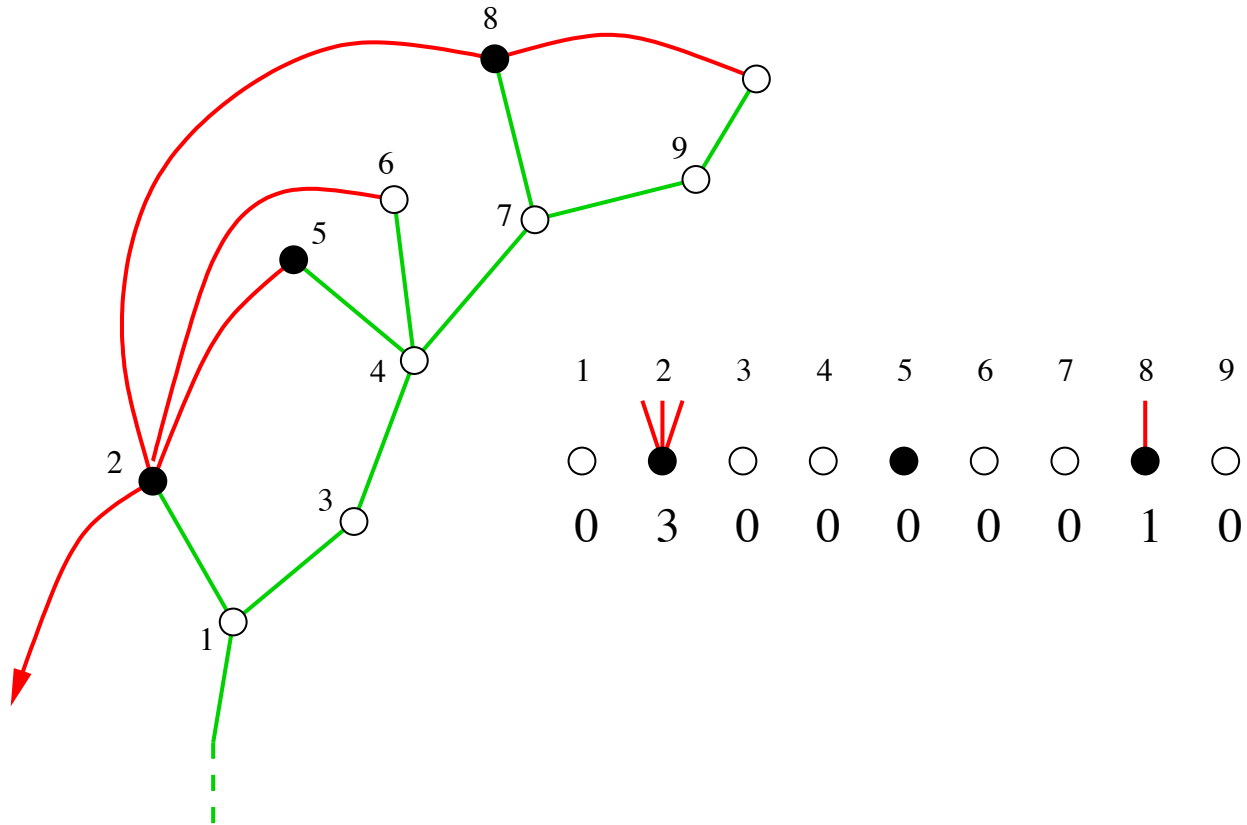
$$|M_S| \leq n + f$$

$$\Rightarrow C_n(f) \leq 2^{n+f}$$

$B_n(f, b) ?$
(arêtes utiles connaissant T_2)

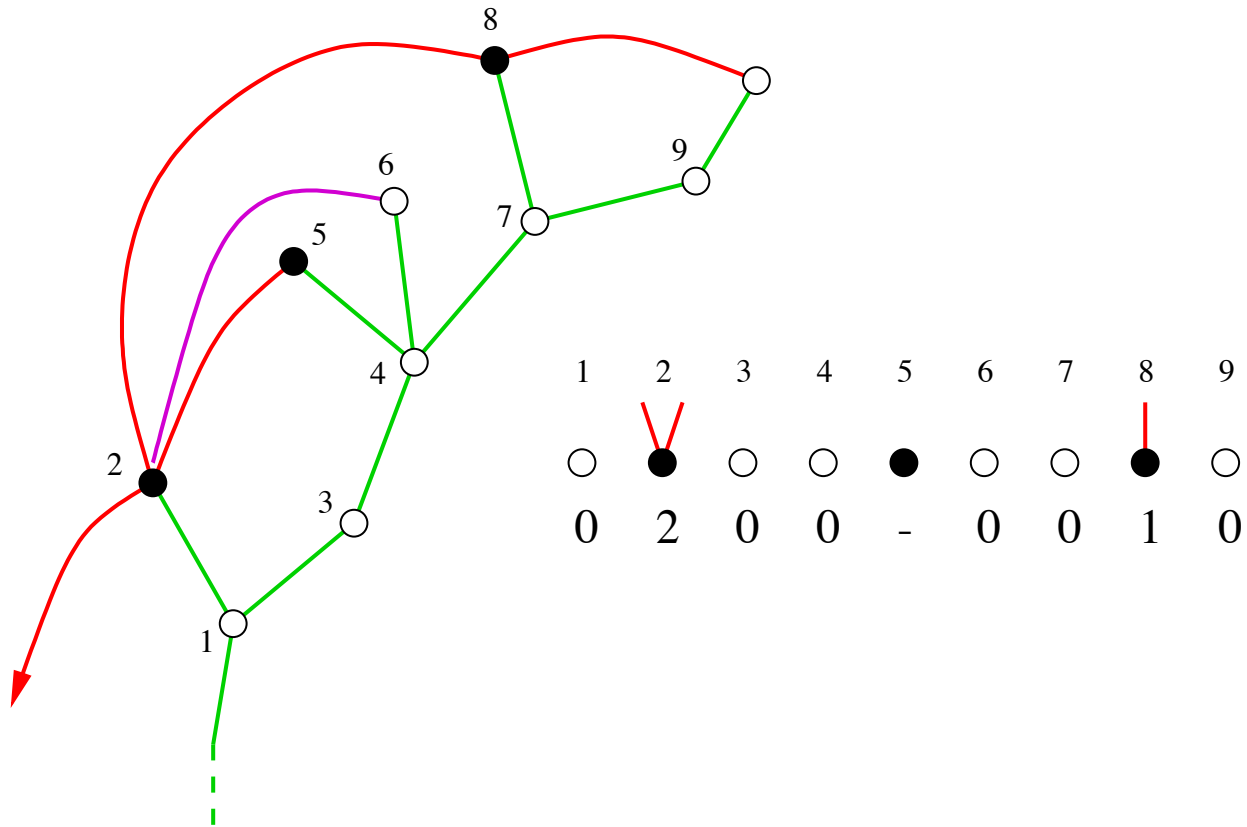


$B_n(f, b) ?$ (arêtes utiles connaissant T_2)



$$B_n(f, b) \leq n^{c_1} \cdot \binom{n+f}{f}$$

$B_n(f, b) ?$ (arêtes utiles connaissant T_2)



$t = \#v \text{ tq}$

$$p_1(v+1) = p_1(v)$$

$$B_n(f, b) \leq n^{c_2} \cdot \max_t \left\{ \binom{b}{t} \cdot \binom{(n-t)+(f-b)}{f-b} \right\}$$

$A_n(f, b)$? (codage de T_2)

$T_2 \in W = \{0, 1\}^{2n+O(1)}$, avec $n = \#0 = \#1$, $f = \#(01)$ et $b = \#(1101)$

$X_p \subset W$ tq $X_p = \{ \text{mots de la forme } (A_i B_i)^p A_{p+1} \}$
avec $A_i = (01)^+$ suite maximale de blocks consecutifs de 01

Ex: $x = \boxed{010101} B_1 \boxed{0101} B_2 \boxed{01} B_3 \cdots B_p \boxed{0101}$, $x \in X_p$

$T_2 \in \cup_p X_p$,

$$A_n(f, b) \leq n^{O(1)} \cdot \max_p |X_p|$$

$$\boxed{010101} B_1 \boxed{0101} B_2 \boxed{01} B_3 \cdots B_p \boxed{0101}$$

Cas 1: $B_i = 1^+$

Ce cas crée exactement :

- un bourgeon, $\boxed{\dots 01} 1 \dots 1 \boxed{01 \dots}$
- un motif 10 (une feuille).

Cas 2: $B_i = 1^*0^+$

Ce cas crée exactement : un motif 10 (une feuille).



Soit w le nombre total de 1 utilisés dans le cas 1,

$$A_n(f, b) \leq n^{c_3} \cdot \max_{p, w} \left\{ \binom{f}{p} \cdot \binom{p}{b} \cdot \binom{w}{b} \cdot \binom{n - w - f + p - b}{p - b} \cdot \binom{n - f}{p - b} \right\}$$

D'où

$$q(n) \leq n^c \cdot \max_{f,b,t,p,w} \{A \cdot B \cdot C\}$$

$$A = \binom{f}{p} \cdot \binom{p}{b} \cdot \binom{w}{b} \cdot \binom{n-w-f+p-b}{p-b} \cdot \binom{n-f}{p-b}$$

$$B = \binom{b}{t} \cdot \binom{n-t+f-b}{f-b}$$

$$C = 2^{n+f}$$

Numériquement,

$$\log_2 q(n)^{1/n} \approx 5.03\dots$$

5.01

Lemme. Si G est maximal ($3n - 6$ arêtes avec un sommet pointé), alors G possède une seule super-triangulation. (Réalisateur sans cw-triangle, [Ossona de Mendez '94], ...)

Comme $(T_2, U_S) \simeq S$,

$$\Rightarrow A \cdot B \leq n^{O(1)} \cdot T_n$$

$$q(n) \leq n^{O(1)} \cdot \max_{f,b,t,p,w} \{ \min \{ A \cdot B, T_n \} \cdot C \}$$

Numériquement,

$$\log_2 q(n)^{1/n} \approx 5.007\dots$$

**Adjacence en temps $O(1)$
avec $5.03n$ bits ?**

La prochaine fois ...