

Algorithmes Combinatoires (2)

Décomposition en coupes - familles bipartitives

Christophe PAUL
(CNRS - LIRMM)

November 19, 2009

Familles bipartitives

Familles des coupes (splits)

Théorème de représentation

Décomposition en coupes (splits)

Graph Labeled Trees (GLT)

Graphes totalement décomposables

Algorithme incrémental de décomposition en split

Graphes de cercles

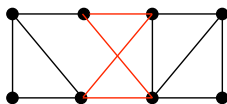
Décomposition en coupe des graphes de cercle

Décomposition et largeur de rang

Coupes (splits)

Une bipartition (A, B) des sommets d'un graphe $G = (V, E)$ est une **coupe (split)** ssi

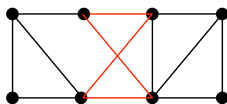
- ▶ $|A| \geq 2$, $|B| \geq 2$;
- ▶ pour $x \in A$ et $y \in B$, $xy \in E$ ssi $x \in N(B)$ et $y \in N(A)$.



Coupes (splits)

Une bipartition (A, B) des sommets d'un graphe $G = (V, E)$ est une **coupe (split)** ssi

- ▶ $|A| \geq 2$, $|B| \geq 2$;
- ▶ pour $x \in A$ et $y \in B$, $xy \in E$ ssi $x \in N(B)$ et $y \in N(A)$.



Exemples de splits

- ▶ toute bipartition non-triviale de la clique
- ▶ toute bipartition non-triviale de $K_{1,n}$

Familles bipartitive

Théorème

La famille des splits d'un graphe est une **famille bipartitive** : si (A_1, A_2) et (B_1, B_2) sont deux splits tels que A_1 et B_1 se chevauchent, alors

- ▶ $(A_1 \cap B_1, A_2 \cup B_2)$ est un split;

Familles bipartitive

Théorème

La famille des splits d'un graphe est une **famille bipartitive** : si (A_1, A_2) et (B_1, B_2) sont deux splits tels que A_1 et B_1 se chevauchent, alors

- ▶ $(A_1 \cap B_1, A_2 \cup B_2)$ est un split;
- ▶ $(A_1 \cap B_2, A_1 \cup B_1)$, $(A_2 \cap B_1, A_1 \cup B_2)$, $(A_2 \cap B_2, A_1 \cup B_1)$ sont des splits;

Familles bipartitive

Théorème

La famille des splits d'un graphe est une **famille bipartitive** : si (A_1, A_2) et (B_1, B_1) sont deux splits tels que A_1 et B_1 se chevauchent, alors

- ▶ $(A_1 \cap B_1, A_2 \cup B_2)$ est un split;
- ▶ $(A_1 \cap B_2, A_1 \cup B_1)$, $(A_2 \cap B_1, A_1 \cup B_2)$, $(A_2 \cap B_2, A_1 \cup B_1)$ sont des splits;
- ▶ $(A_1 \Delta B_1, A_1 \Delta B_2)$ est un split.

Familles bipartitive

Théorème

La famille des splits d'un graphe est une **famille bipartitive** : si (A_1, A_2) et (B_1, B_2) sont deux splits tels que A_1 et B_1 se chevauchent, alors

- ▶ $(A_1 \cap B_1, A_2 \cup B_2)$ est un split;
- ▶ $(A_1 \cap B_2, A_1 \cup B_1)$, $(A_2 \cap B_1, A_1 \cup B_2)$, $(A_2 \cap B_2, A_1 \cup B_1)$ sont des splits;
- ▶ $(A_1 \Delta B_1, A_1 \Delta B_2)$ est un split.

Une bipartition (A_1, A_2) est **forte** si elle ne chevauche aucune autre bipartition de la famille.

Représentation arborée

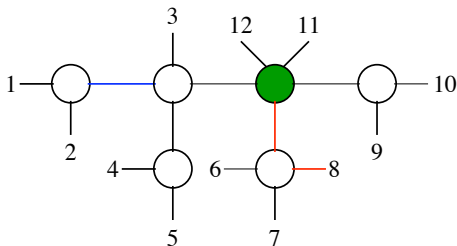
Théorème Soit \mathcal{F} une famille bipartitive sur l'ensemble X . Il existe un arbre T tel que

- ▶ les **feuilles** de T sont en bijection avec X
- ▶ les **arêtes** de T sont en bijection avec les **bipartitions fortes**

Représentation arborée

Théorème Soit \mathcal{F} une famille bipartitive sur l'ensemble X . Il existe un arbre T tel que

- ▶ les **feuilles** de T sont en bijection avec X
- ▶ les **arêtes** de T sont en bijection avec les **bipartitions fortes**

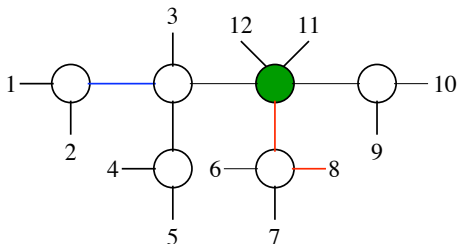


$$\mathcal{F} = \{(\{1, 2\}, -), (\{1, 2, 3\}, -), (\{3, 4, 5\}, -), (\{1, 2, 4, 5\}, -), \\ (\{1, 2, 3, 4, 5\}, -), (\{6, 7\}, -), (\{7, 8\}, -), (\{6, 8\}, -), \\ (\{6, 7, 8\}, -), (\{9, 10\}, -)\}$$

Représentation arborée

Théorème Soit \mathcal{F} une famille bipartitive sur l'ensemble X . Il existe un arbre T tel que

- ▶ les **feuilles** de T sont en bijection avec X
- ▶ les **arêtes** de T sont en bijection avec les **bipartitions fortes**



Il est possible d'étiqueter les nœuds de T **premier** ou **dégénéré** tq :

- ▶ pour toute bipartition (A_1, A_2) non-forte, il existe un nœud dégénéré u de T tel que $A_1 = \bigcup_{v \in C} L(T_v)$ pour $C \subset N_T(u)$.

Familles bipartitives

Familles des coupes (splits)

Théorème de représentation

Décomposition en coupes (splits)

Graph Labeled Trees (GLT)

Graphes totalement décomposables

Algorithme incrémental de décomposition en split

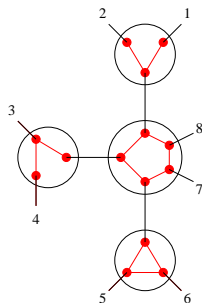
Graphes de cercles

Décomposition en coupe des graphes de cercle

Décomposition et largeur de rang

Graph-labeled trees (GLT)

- ▶ une paire (T, \mathcal{F}) , où T un arbre et \mathcal{F} un ensemble de graphes
- ▶ chaque nœud v de T tq $d(v) = k$ est étiqueté par un graphe $G_v \in \mathcal{F}$ tq $|V(G_v)| = k$
- ▶ il y a une **bijection** ρ_v entre les arêtes de T incidentes à v et les sommets de G_v (\rightarrow **sommets marqueurs**)



Graphe d'accessibilité d'un GLT

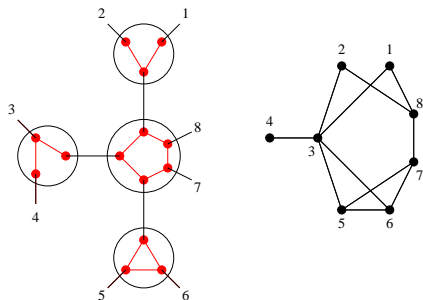
Le **graphe d'accessibilité** $G_S(T, \mathcal{F})$ d'un GLT (T, \mathcal{F}) , est le graphe dont les sommets sont les feuilles de T et les arêtes sont :

- ▶ $xy \in E(G_S(T, \mathcal{F}))$ ssi $\rho_v(uv)\rho_v(vw) \in E(G_v)$,
 \forall arête uv, vw du chemin entre x et y dans T

Graphe d'accessibilité d'un GLT

Le **graphe d'accessibilité** $G_S(T, \mathcal{F})$ d'un GLT (T, \mathcal{F}) , est le graphe dont les sommets sont les feuilles de T et les arêtes sont :

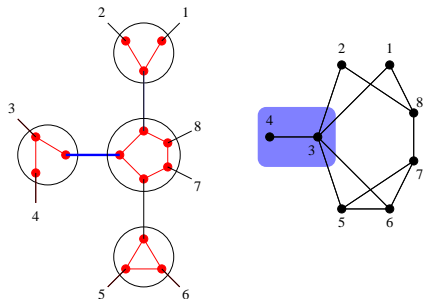
- ▶ $xy \in E(G_S(T, \mathcal{F}))$ ssi $\rho_v(uv)\rho_v(vw) \in E(G_v)$,
 \forall arête uv, vw du chemin entre x et y dans T



Graphe d'accessibilité d'un GLT

Le **graphe d'accessibilité** $G_S(T, \mathcal{F})$ d'un GLT (T, \mathcal{F}) , est le graphe dont les sommets sont les feuilles de T et les arêtes sont :

- ▶ $xy \in E(G_S(T, \mathcal{F}))$ ssi $\rho_v(uv)\rho_v(vw) \in E(G_v)$,
 \forall arête uv, vw du chemin entre x et y dans T

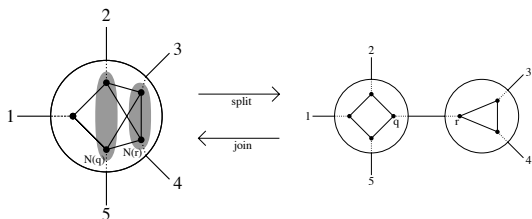


Lemme Toute arête uv de T définit une coupe de $G_S(T, \mathcal{F})$.

Graphe d'accessibilité d'un GLT

Le **graphe d'accessibilité** $G_S(T, \mathcal{F})$ d'un GLT (T, \mathcal{F}) , est le graphe dont les sommets sont les feuilles de T et les arêtes sont :

- ▶ $xy \in E(G_S(T, \mathcal{F}))$ ssi $\rho_v(uv)\rho_v(vw) \in E(G_v)$,
 \forall arête uv, vw du chemin entre x et y dans T



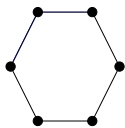
Lemme Toute arête uv de T définit une coupe de $G_S(T, \mathcal{F})$.

Théorème de décomposition en coupes

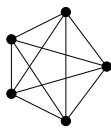
Théorème [Cunningham'82]

Pour tout graphe connexe G , il existe un unique GLT (T, \mathcal{F}) ayant un minimum de nœuds tq

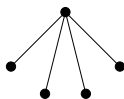
1. $G = G_S(T, \mathcal{F})$,
2. tout graphe de \mathcal{F} est premier ou dégénéré pour la décomposition en split



Prime



Degenerate

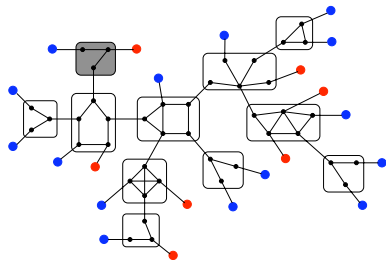


Théorème de décomposition en coupes

Théorème [Cunningham'82]

Pour tout graphe connexe G , il existe un unique GLT (T, \mathcal{F}) ayant un minimum de nœuds tq

1. $G = G_S(T, \mathcal{F})$,
2. tout graphe de \mathcal{F} est premier ou dégénéré pour la décomposition en split



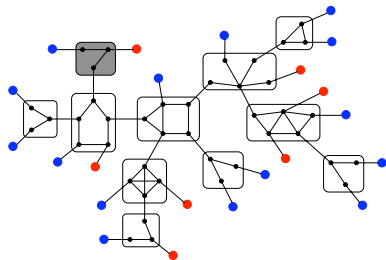
→ Ce GLT, noté $(T, \mathcal{F}) = ST(G)$, est appelé **split tree**

Théorème de décomposition en coupes

Théorème [Cunningham'82]

Pour tout graphe connexe G , il existe un unique GLT (T, \mathcal{F}) ayant un minimum de nœuds tq

1. $G = G_S(T, \mathcal{F})$,
2. tout graphe de \mathcal{F} est premier ou dégénéré pour la décomposition en split



→ Ce GLT, noté $(T, \mathcal{F}) = ST(G)$, est appelé **split tree**

Remarque : Tout split de G est représenté soit par une arête de $ST(G)$ ou par une union d'arête incidente à un nœud dégénéré.

Graphes totalement décomposables

Théorème

Un graphe G est **totalemment décomposable** pour la décomposition en split ssi

- ▶ $ST(G)$ ne contient que des nœuds dégénérés (clique ou star)
- ▶ G est **distance héréditaire**
- ▶ G admet un ordre d'élimination des sommets vrai/faux jumeaux et pendants
- ▶ G ne contient pas de **trou**, de **diamand**, de **maison** comme sous-graphe induit

Graphes totalement décomposables

Théorème

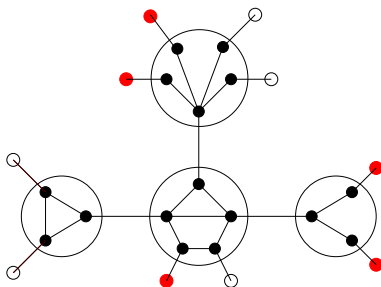
Un graphe G est **totalemt décomposable** pour la décomposition en split ssi

- ▶ $ST(G)$ ne contient que des nœuds dégénérés (clique ou star)
- ▶ G est **distance héréditaire**
- ▶ G admet un ordre d'élimination des sommets vrai/faux jumeaux et pendants
- ▶ G ne contient pas de **trou**, de **diamand**, de **maison** comme sous-graphe induit

→ nombreux algorithmes de reconnaissance en temps linéaire

Caractérisation incrémentale du split-tree

Un sommet x est ajouté au graphe $G = (V, E)$ avec pour voisinage $S \subseteq V$.
(S est en rouge)



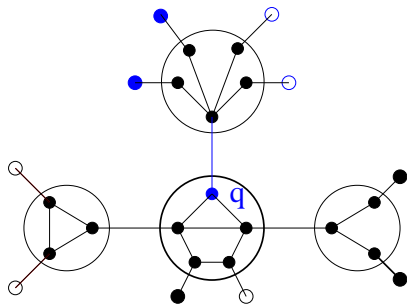
► Comment calculer le split-tree $ST(G + (x, S))$ depuis $ST(G)$?

Caractérisation incrémentale du split-tree

Etat des sommets marqueurs

Soit q un sommet marqueur (ou une feuille)

Soit $L(q)$ l'ensemble des feuilles l tq le chemin de l à q contient l'arête de T incidente à q .



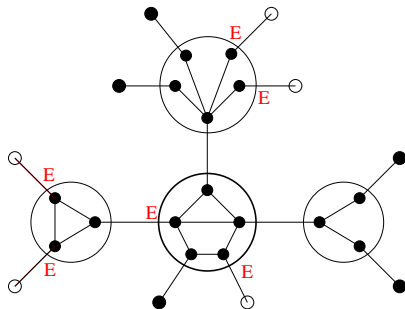
Caractérisation incrémentale du split-tree

Etat des sommets marqueurs

Soit q un sommet marqueur (ou une feuille)

Soit $L(q)$ l'ensemble des feuilles l tq le chemin de l à q contient l'arête de T incidente à q .

- ▶ q est **vide** si $L(q) \cap S = \emptyset$



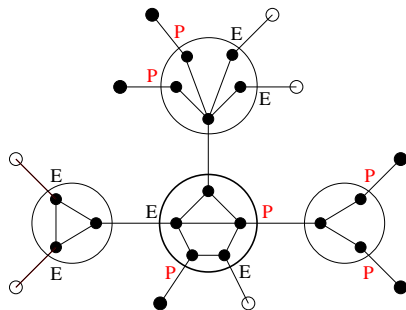
Caractérisation incrémentale du split-tree

Etat des sommets marqueurs

Soit q un sommet marqueur (ou une feuille)

Soit $L(q)$ l'ensemble des feuilles l tq le chemin de l à q contient l'arête de T incidente à q .

- ▶ q est **vide** si $L(q) \cap S = \emptyset$
- ▶ q est **parfait** si $L(q) \cap S = \text{Accessible}(q)$



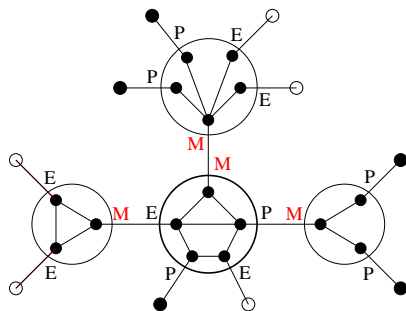
Caractérisation incrémentale du split-tree

Etat des sommets marqueurs

Soit q un sommet marqueur (ou une feuille)

Soit $L(q)$ l'ensemble des feuilles l tq le chemin de l à q contient l'arête de T incidente à q .

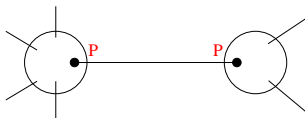
- ▶ q est **vide** si $L(q) \cap S = \emptyset$
- ▶ q est **parfait** si $L(q) \cap S = \text{Accessible}(q)$
- ▶ q est **mixte** sinon



Caractérisation incrémentale du split-tree

Le split-tree de $G + (x, S)$ est obtenu à partir de $ST(G)$ selon 3 cas :

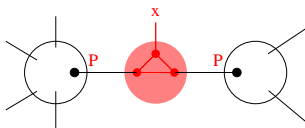
- ▶ 1 - T possède un arête e dont les extrémités sont parfaites



Caractérisation incrémentale du split-tree

Le split-tree de $G + (x, S)$ est obtenu à partir de $ST(G)$ selon 3 cas :

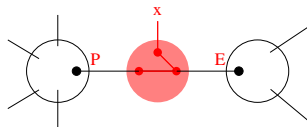
- ▶ 1 - T possède un arête e dont les extrémités sont parfaites



Caractérisation incrémentale du split-tree

Le split-tree de $G + (x, S)$ est obtenu à partir de $ST(G)$ selon 3 cas :

- ▶ 1 - T possède un arête e dont les extrémités sont parfaites
- ▶ 2 - T possède un arête e , dont une extrémité est parfaite et l'autre vide

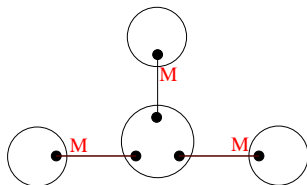


Caractérisation incrémentale du split-tree

Le split-tree de $G + (x, S)$ est obtenu à partir de $ST(G)$ selon 3 cas :

- ▶ 1 - T possède un arête e dont les extrémités sont parfaites
- ▶ 2 - T possède un arête e , dont une extrémité est parfaite et l'autre vide
- ▶ 3 - T possède un nœud **totalelement-mixte** u

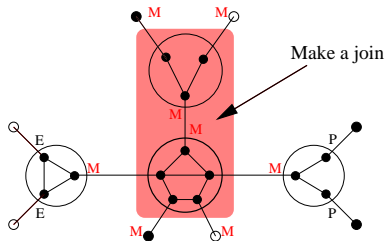
tout sommet marqueur voisin de u est mixte



Caractérisation incrémentale du split-tree

Le split-tree de $G + (x, S)$ est obtenu à partir de $ST(G)$ selon 3 cas :

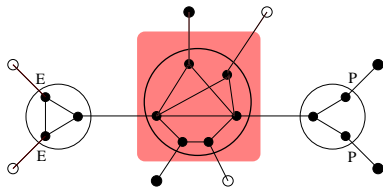
- ▶ 1 - T possède un arête e dont les extrémités sont parfaites
- ▶ 2 - T possède un arête e , dont une extrémité est parfaite et l'autre vide
- ▶ 3 - T possède un nœud **totalemment-mixte** u (l'ensemble des nœuds totalement-mixtes forme un sous-arbre de T)



Caractérisation incrémentale du split-tree

Le split-tree de $G + (x, S)$ est obtenu à partir de $ST(G)$ selon 3 cas :

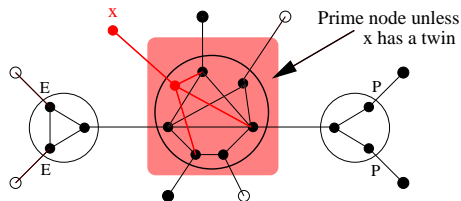
- ▶ 1 - T possède un arête e dont les extrémités sont parfaites
- ▶ 2 - T possède un arête e , dont une extrémité est parfaite et l'autre vide
- ▶ 3 - T possède un nœud **totalemment-mixte** u (l'ensemble des nœuds totalement-mixtes forme un sous-arbre de T)



Caractérisation incrémentale du split-tree

Le split-tree de $G + (x, S)$ est obtenu à partir de $ST(G)$ selon 3 cas :

- ▶ 1 - T possède un arête e dont les extrémités sont parfaites
- ▶ 2 - T possède un arête e , dont une extrémité est parfaite et l'autre vide
- ▶ 3 - T possède un nœud **totale-ment-mixte** u (l'ensemble des nœuds totale-ment-mixtes forme un sous-arbre de T)



Caractérisation incrémentale du split-tree

Théorème: La complexité de l'algorithme incrémental de split-decomposition est

$$O((n + m)\alpha(n, m))$$

Ingrédients pour l'analyse de complexité

- ▶ ajout des sommets selon un ordre **LexBFS**
- ▶ analyse fine de complexité amortie basée sur une **technique de déchargement**

Résultats connus

- ▶ [Cunningham 1982] $O(nm)$
- ▶ [Ma, Spinrad 1994] $O(n^2)$
- ▶ [Dahlhaus 2000] $O(n + m)$ (difficile !)
- ▶ [Charbit, De Montgolfier, Raffinot 2009] $O(n + m)$

Familles bipartitives

Familles des coupes (splits)

Théorème de représentation

Décomposition en coupes (splits)

Graph Labeled Trees (GLT)

Graphes totalement décomposables

Algorithme incrémental de décomposition en split

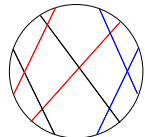
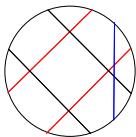
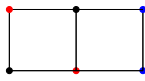
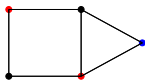
Graphes de cercles

Décomposition en coupe des graphes de cercle

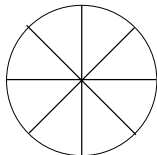
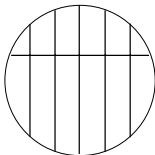
Décomposition et largeur de rang

Graphes de cercle

- ▶ Un graphe est un **graphe de cercle** ssi c'est le graphe d'intersection de cordes dans un cercle.

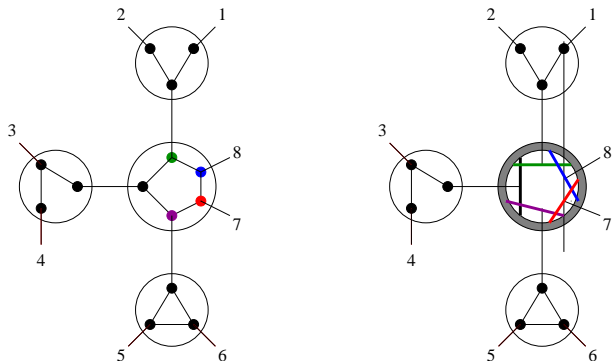


Question : Quels sont ces graphes ?



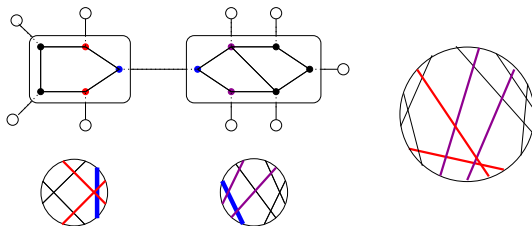
Graphes de cercle

- ▶ Un graphe est un **graphe de cercle** ssi c'est le graphe d'intersection de cordes dans un cercle.
- ▶ Un graphe est un **graphe de cercle** ssi tous les nœuds premiers de son split-tree sont des graphes de cercle



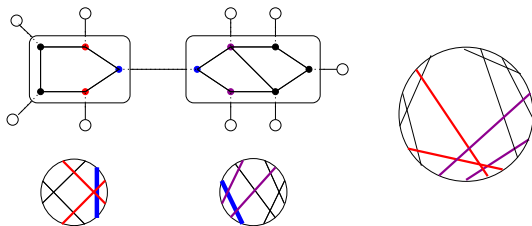
Graphes de cercle

- ▶ Un graphe est un **graphe de cercle** ssi c'est le graphe d'intersection de cordes dans un cercle.
- ▶ Un graphe est un graphe de cercle ssi tous les nœuds premiers de son split-tree sont des graphes de cercle
- ▶ Un graphe de cercle est **premier** ssi il possède un unique diagramme de corde (à miroir près)



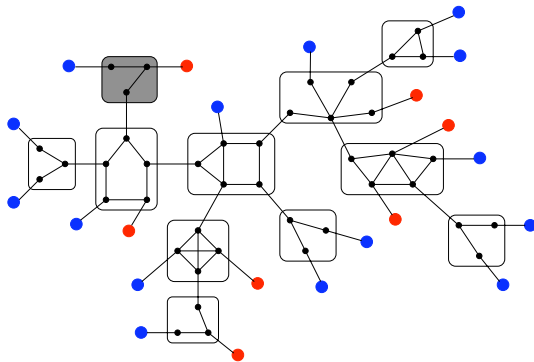
Graphes de cercle

- ▶ Un graphe est un **graphe de cercle** ssi c'est le graphe d'intersection de cordes dans un cercle.
- ▶ Un graphe est un graphe de cercle ssi tous les nœuds premiers de son split-tree sont des graphes de cercle
- ▶ Un graphe de cercle est **premier** ssi il possède un unique diagramme de corde (à miroir près)



Graphes de cercle et décomposition en coupes

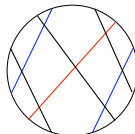
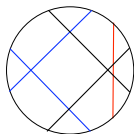
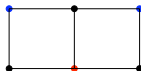
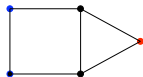
Question : est-ce que le graphe correspondant est un graphe de cercle ?



Reconnaissance des graphes de cercle

Sommet - corde extrême

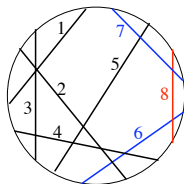
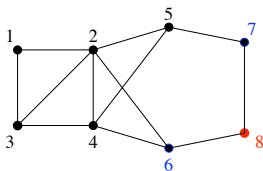
Un sommet x est **extrême** si sa corde $c(x)$ coupe le diagramme de sorte que les cordes de tous les non-voisins de x sont soit toutes à droite ou toutes à gauche de $c(x)$.



Reconnaissance des graphes de cercle

Premier Lemme LexBFS

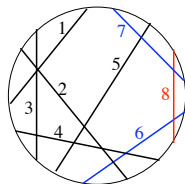
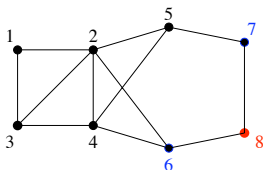
Soient G un graphe decercle et σ un ordre LexBFS de G terminant en x . Alors il existe un diagramme de cordes pour G dans lequel $c(x)$ est extrême.



Reconnaissance des graphes de cercle

Premier Lemme LexBFS

Soient G un graphe decercle et σ un ordre LexBFS de G terminant en x . Alors il existe un diagramme de cordes pour G dans lequel $c(x)$ est extrême.

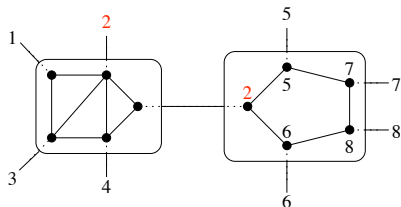
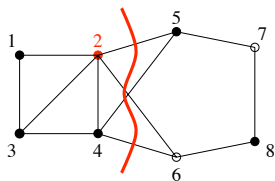


- ▶ **Conséquence:** en temps constant, nous savons où insérer $c(x)$ dans un diagramme de cordes de G .
- ▶ **Mais** beaucoup de diagrammes sont possibles, sauf si G est premier.

Reconnaissance des graphes de cercle

Un second Lemme LexBFS

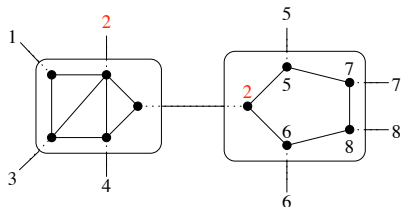
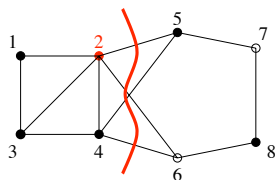
Soit σ un ordre LexBFS de $G = G_S(T, \mathcal{F})$. L'ordre σ_u "induit" par les sommets marqueurs de G_u est un ordre LexBFS.



Reconnaissance des graphes de cercle

Un second Lemme LexBFS

Soit σ un ordre LexBFS de $G = G_S(T, \mathcal{F})$. L'ordre σ_u "induit" par les sommets marqueurs de G_u est un ordre LexBFS.

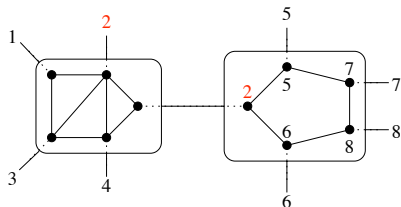
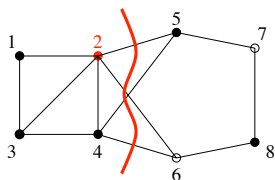


- **conséquence:** il est possible d'appliquer le lemme précédent sur les nœuds du split-tree.

Reconnaissance des graphes de cercle

Un second Lemme LexBFS

Soit σ un ordre LexBFS de $G = G_S(T, \mathcal{F})$. L'ordre σ_u "induit" par les sommets marqueurs de G_u est un ordre LexBFS.



- ▶ **conséquence:** il est possible d'appliquer le lemme précédent sur les nœuds du split-tree.
- ▶ **Que reste-t-il à faire ?**
Effectuer les opérations de join nécessaires sur les nœuds totalement mixtes.

Reconnaissance des graphes de cercle

1. Insertion des sommets selon un ordre LexBFS.
2. Maintien du split-tree en représentant les nœuds premiers par leur diagramme de corde.
3. Faire les opérations de join sur les diagrammes des nœuds totalement mixtes en utilisant les lemmes LexBFS.
⇒ nécessite quelques astuces pour manipuler la représentation des diagrammes

Reconnaissance des graphes de cercle

1. Insertion des sommets selon un ordre LexBFS.
2. Maintien du split-tree en représentant les nœuds premiers par leur diagramme de corde.
3. Faire les opérations de join sur les diagrammes des nœuds totalement mixtes en utilisant les lemmes LexBFS.
⇒ nécessite quelques astuces pour manipuler la représentation des diagrammes

Théorème :

La reconnaissance des graphes de cercle est résolue en temps

$$O((n + m)\alpha(n, m))$$

→ Complexité connue : $O(n^2)$ [Spinrad, J. of Alg. (16), 1994]

Familles bipartitives

Familles des coupes (splits)

Théorème de représentation

Décomposition en coupes (splits)

Graph Labeled Trees (GLT)

Graphes totalement décomposables

Algorithme incrémental de décomposition en split

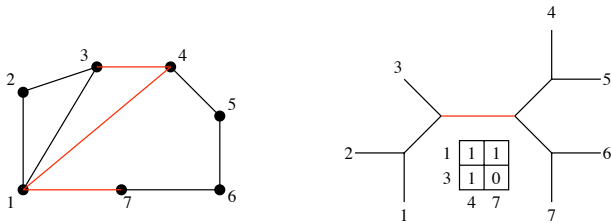
Graphes de cercles

Décomposition en coupe des graphes de cercle

Décomposition et largeur de rang

Largeur de rang

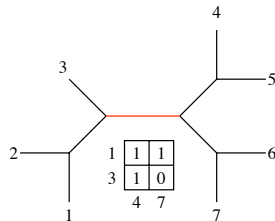
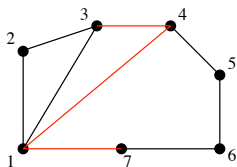
Soient $G = (V, E)$ un graphe et T un arbre ternaire dont les feuilles sont en bijection avec V



- Chaque arête e de T définit une bipartition (A, B) de V

Largeur de rang

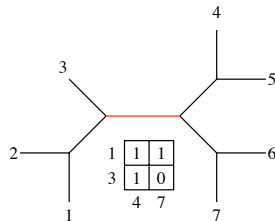
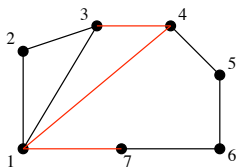
Soient $G = (V, E)$ un graphe et T un arbre ternaire dont les feuilles sont en bijection avec V



- ▶ Chaque arête e de T définit une bipartition (A, B) de V
- ▶ Largeur de e : $rk(e) = rk(A \times B)$

Largeur de rang

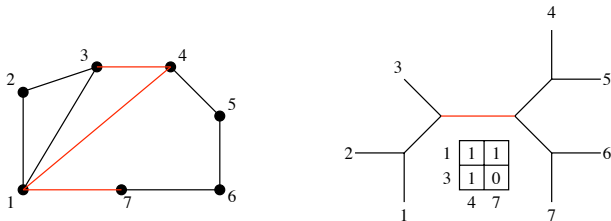
Soient $G = (V, E)$ un graphe et T un arbre ternaire dont les feuilles sont en bijection avec V



- ▶ Chaque arête e de T définit une bipartition (A, B) de V
- ▶ Largeur de e : $rk(e) = rk(A \times B)$
- ▶ Largeur de T : $rk(T) = \max_{e \in T} rk(e)$

Largeur de rang

Soient $G = (V, E)$ un graphe et T un arbre ternaire dont les feuilles sont en bijection avec V



- ▶ Chaque arête e de T définit une bipartition (A, B) de V
- ▶ Largeur de e : $rk(e) = rk(A \times B)$
- ▶ Largeur de T : $rk(T) = \max_{e \in T} rk(e)$
- ▶ Largeur de G : $rw(G) = \min_T rk(T)$

Largeur de rang

Lemme

Soit $ST(G)$ le split-tree d'un graphe G et \mathcal{F} l'ensemble des graphes (premiers) étiquetant $ST(G)$. Alors

$$rw(G) = \max_{H \in \mathcal{F}} rw(H)$$

Preuve : il suffit de montrer que la largeur d'un split est 1.

Largeur de rang

Lemme

Soit $ST(G)$ le split-tree d'un graphe G et \mathcal{F} l'ensemble des graphes (premiers) étiquetant $ST(G)$. Alors

$$rw(G) = \max_{H \in \mathcal{F}} rw(H)$$

Preuve : il suffit de montrer que la largeur d'un split est 1.

Lemme

Les graphes de largeur de rang 1 sont exactement les graphes totalement décomposable par la split-decomposition

Preuve : il suffit de montrer que les cliques et les graphes $K_{1,n}$ ont une largeur de rang 1