Composantes fortement connexes

Algorithmique – L3

François Laroussinie

15 novembre 2010

Plan

Plan

1. Définitions
2. La définition des coefficients
3. Calcul des coefficients

Composantes fortement connexes

$G = (S, A)$: un graphe orienté.

Définition : Une composante fortement connexe (CFC) $\mathcal{C}$ de G est un **sous-ensemble maximal** de sommets de G tel que :
si $u, v \in \mathcal{C}$, alors $u \rightarrow_G^* v$ et $v \rightarrow_G^* u$.

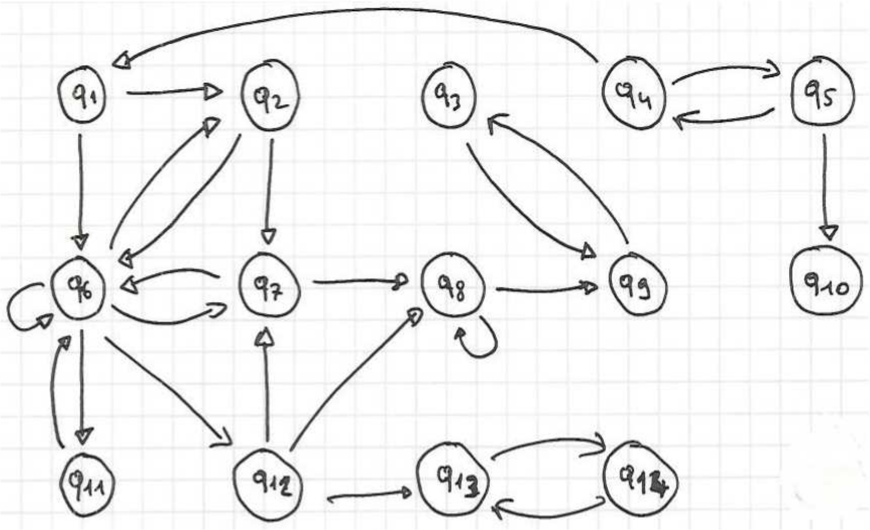
Composantes fortement connexes

$G = (S, A)$: un graphe orienté.

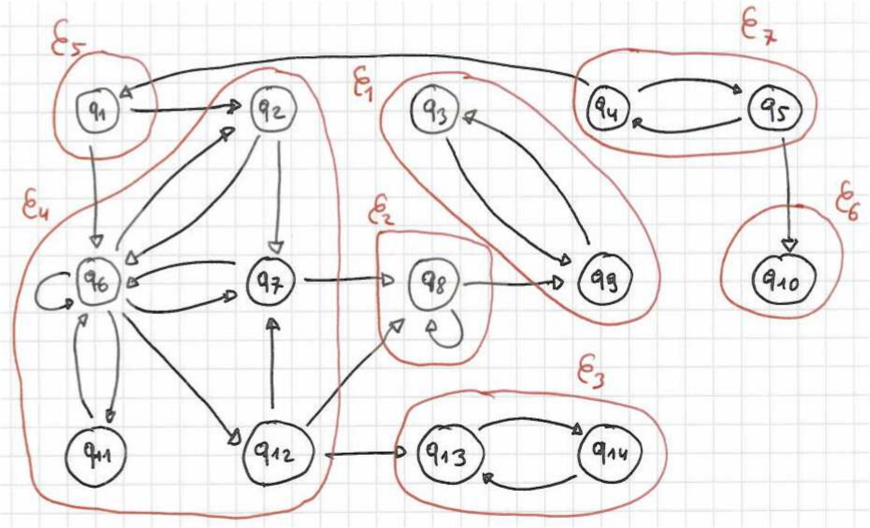
Définition : Une composante fortement connexe (CFC) $\mathcal{C}$ de G est un **sous-ensemble maximal** de sommets de G tel que :
si $u, v \in \mathcal{C}$, alors $u \rightarrow_G^* v$ et $v \rightarrow_G^* u$.

Algorithme de Tarjan (SIAM Journal of Computing, Vol. 1, No. 2, June 1972)

Exemple : le graphe G



Exemple : les CFC de G



Idée de l'algorithme

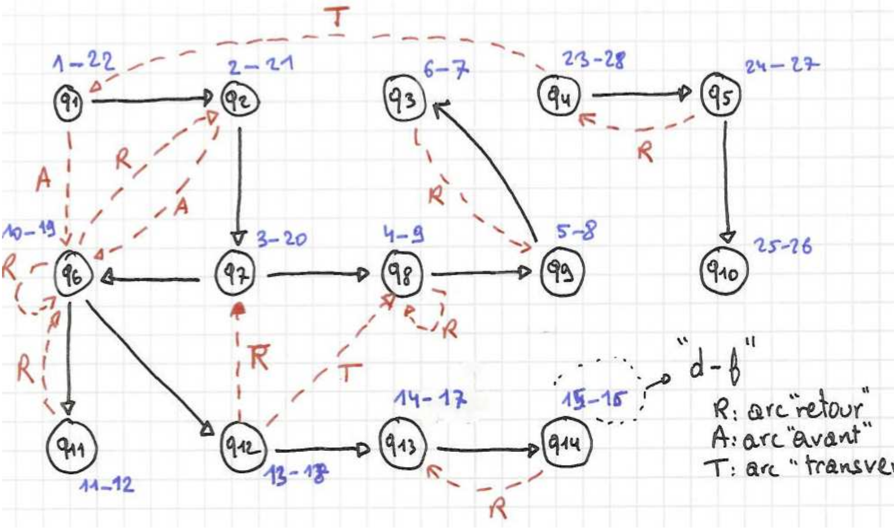
L'algorithme de recherche des CFC est basé sur les dates d[$-$] et la forêt G_Π calculées lors d'un parcours en profondeur.

Idée de l'algorithme

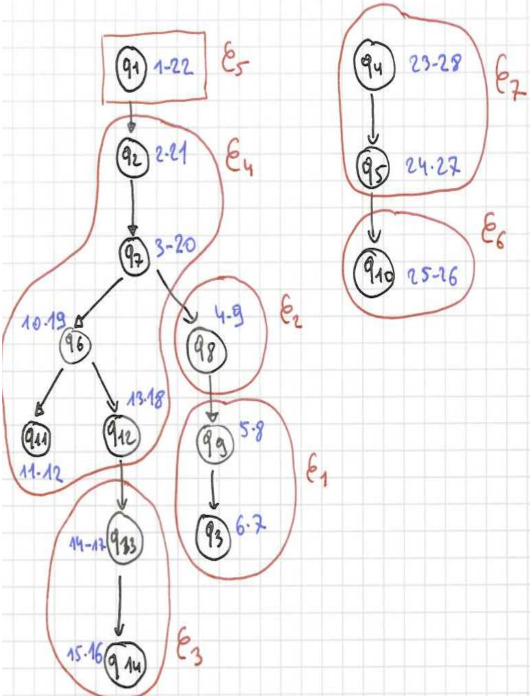
L'algorithme de recherche des CFC est basé sur les dates d[$-$] et la forêt G_Π calculées lors d'un parcours en profondeur.

exemple. . .

Exemple : un parcours en profondeur de G



On en déduit la forêt G_Π...



q1
1-22
E5
q2
2-21
E4
q7
3-20
10-19
q6
4-9
q8
E2
13-18
q11
q12
11-12
q9
5-8
E1
q3
6-7
14-17
q13
q14
15-16
E3
q4
23-28
E7
q5
24-27
q10
25-26
E6

Observations

Les sommets d'une même CFC n'apparaissent pas n'importe comment dans la forêt G_{Π}...

Est-ce un hasard ?

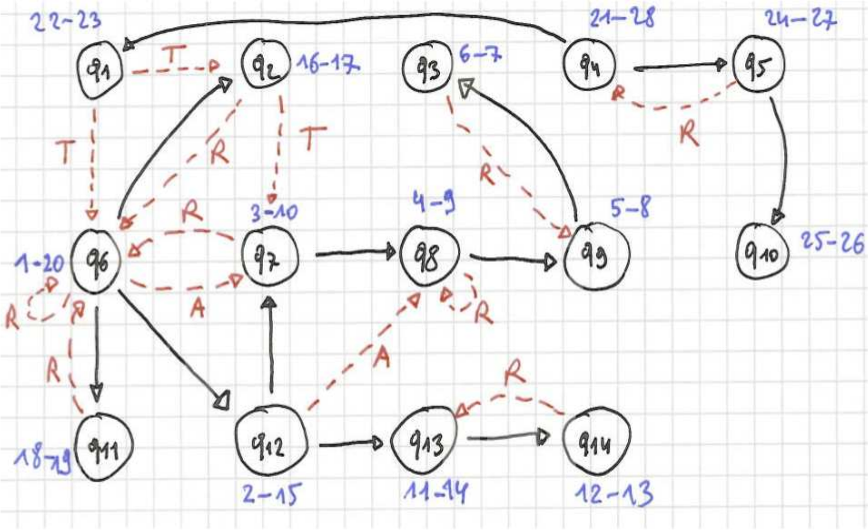
Observations

Les sommets d'une même CFC n'apparaissent pas n'importe comment dans la forêt G_{Π}...

Est-ce un hasard ?

Autre essai...

Autre parcours

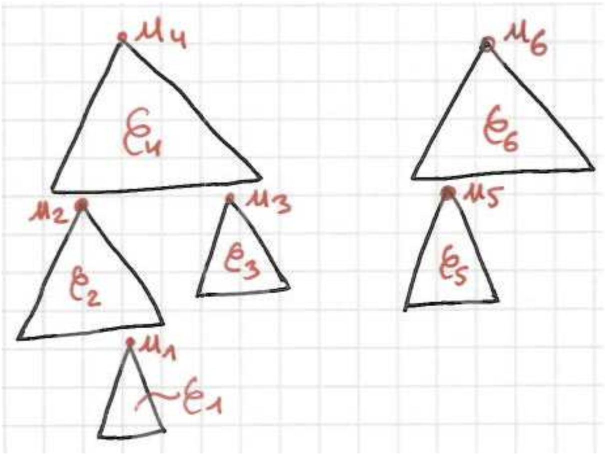


Propriétés - 1

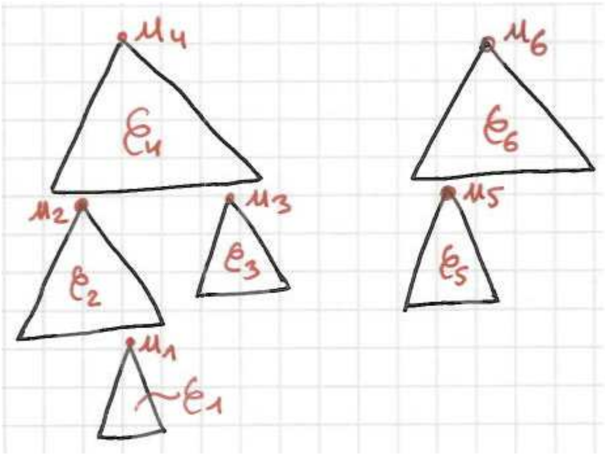
Propriété 21 Soient $v, w \in S$ tels que $v \leftrightarrow^*_G w$,

1. alors v et w ont un ancêtre commun dans G_Π ; et
2. soit u le sommet tel que (1) $u \rightarrow^*_{G_\Pi} v$, (2) $u \rightarrow^*_{G_\Pi} w$ et (3) $\mathrm{d}[u]$ est maximal, alors on a : $v \leftrightarrow^*_G u$ et $u \leftrightarrow^*_G w$.

Place des CFC dans la forêt G_Π



Place des CFC dans la forêt G_Π



Corollaire :

Soit $\mathcal{C}$ une CFC de G.

Alors l'arborescence G_Π restreinte aux sommets de $\mathcal{C}$ est un arbre (couvrant $\mathcal{C}$).

Idée générale de l'algorithme

$\Rightarrow$ calculer un coefficient r[x] pour chaque $x \in S$ à partir des dates d[$-$] de manière à ce que $\forall x \in S$, on ait :
r[x] $=$ d[x] ssi x est la racine de sa CFC

Def : la racine d'une CFC $\mathcal{C}$ est le premier sommet de $\mathcal{C}$ à être découvert dans le parcours en profondeur.

Idée générale de l'algorithme

$\Rightarrow$ calculer un coefficient r[x] pour chaque $x \in S$ à partir des dates d[$-$] de manière à ce que $\forall x \in S$, on ait :
r[x] $=$ d[x] ssi x est la racine de sa CFC

Def : la racine d'une CFC $\mathcal{C}$ est le premier sommet de $\mathcal{C}$ à être découvert dans le parcours en profondeur.

Les coefficients r[$-$] vous nous permettre de trouver les racines des CFC, puis de trouver les autres sommets de ces CFC.

Plan

Définition des coefficients r

$\forall x \in S$, on définit le coefficient r$[x]$ par :

$$\mathrm{r}[x] \stackrel{\text{def}}{=} \min\Big(\{\mathrm{d}[x]\} \cup \{\mathrm{d}[w] \mid (1)\; x \rightarrow^*_{G_\Pi} \xrightarrow{T/R} w \wedge$$

$$(2)\; \exists u \in S.u \rightarrow^*_{G_\Pi} x \wedge u \rightarrow^*_{G_\Pi} w \;\wedge\; (3) u \leftrightarrow^*_{G} w\}\Big)$$

Définition des coefficients r

$\forall x \in S$, on définit le coefficient $\mathrm{r}[x]$ par :

$$\mathrm{r}[x] \stackrel{\text{def}}{=} \min\Big(\{\mathrm{d}[x]\} \cup \{\mathrm{d}[w] \mid (1)\ x \rightarrow^*_{G_\Pi} \xrightarrow{T/R} w \wedge$$
$$(2)\ \exists u \in S.u \rightarrow^*_{G_\Pi} x \wedge u \rightarrow^*_{G_\Pi} w \ \wedge\ (3) u \leftrightarrow^*_G w\}\Big)$$

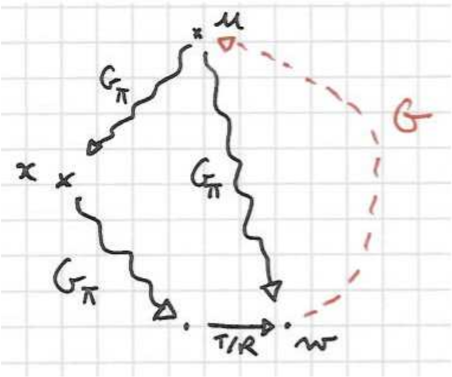
$\mathrm{r}[x]$ est le minimum entre $\mathrm{d}[x]$ et les dates de découverte des sommets w qui :

1. sont accessibles depuis x par un chemin de G_Π prolongé par **un** arc “retour” ou un arc “transverse”, et tels que
2. x et w ont un ancêtre commun dans G_Π, et
3. u et w sont dans la même CFC.

Définition des coefficients r

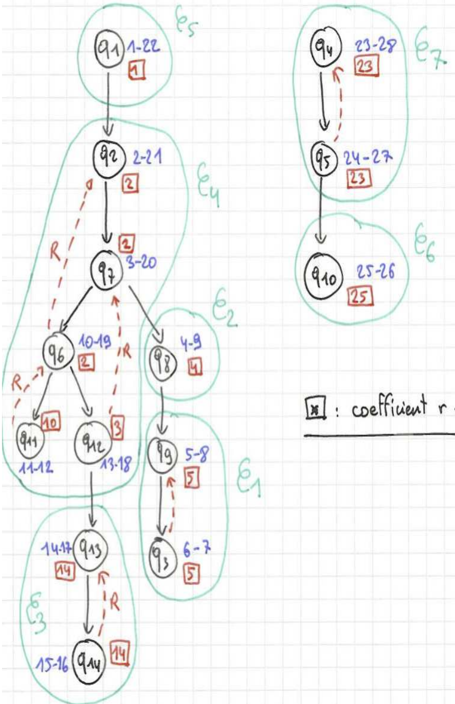
$$\mathrm{r}[x] \stackrel{\text{def}}{=} \min\Big(\{\mathrm{d}[x]\} \cup \{\mathrm{d}[w] \mid (1)\ x \to^*_{G_\Pi} \xrightarrow{T/R} w \land$$

$$(2)\ \exists u \in S.u \to^*_{G_\Pi} x \land u \to^*_{G_\Pi} w\ \land\ (3) u \leftrightarrow^*_G w\}\Big)$$



Autre définition

r[x] est le minimum entre d[x] et les d[w] tel que w est un sommet de la CFC de x accessible depuis x par un chemin de la forme $x \rightarrow^{*}_{G_\Pi} \xrightarrow{T/R} w$.



$\boxed{x}$: coefficient r du nœud.

Propriétés des coefficients r

Comme w, u et x sont dans la même CFC, on a :

Propriété 22

Pour tout sommet x d'une CFC $\mathcal{C}$ de racine $u_{\mathcal{C}}$, on a :

$$\mathrm{d}[u_{\mathcal{C}}] \ \leq \ \mathrm{r}[x] \ \leq \ \mathrm{d}[x]$$

Propriétés des coefficients r

Comme w, u et x sont dans la même CFC, on a :

Propriété 22

Pour tout sommet x d'une CFC $\mathcal{C}$ de racine $u_{\mathcal{C}}$, on a :

$$\mathrm{d}[u_{\mathcal{C}}] \ \leq \ \mathrm{r}[x] \ \leq \ \mathrm{d}[x]$$

Propriété 23

Un sommet x est la racine d'une composante fortement connexe **si et seulement si** $\mathrm{r}(x) = \mathrm{d}[x]$.

Plan

Algorithme de Tarjan

On modifie l'algorithme de parcours en profondeur...

Les nouvelles variables utilisées sont :

- *nbcfc* : pour compter le nombre de CFC trouvées ;
- *NumCFC*$[-]$: un tableau donnant pour chaque sommet le numéro de sa CFC ;
- $\mathrm{r}_A[-]$: un tableau pour stocker les coefficients r ;
- P est une pile pour stocker des sommets de G.

Algorithme de Tarjan

Procédure `Tarjan-CFC`(G)
temps $:= 0$;
nbcfc $:= 0$;
$P :=$ `Pile vide`;
pour chaque $x \in S$ **faire**
    Couleur[x] $:=$ blanc;
    NumCFC[x] $:=$ `undef`;
pour chaque $x \in S$ **faire**
    si *Couleur*[x] $=$ *blanc* **alors** `CFC`(x);
retourner *NumCFC*[]

```
Procédure CFC(x)
temps++; d[x] := temps; Couleur[x] := gris;
r_A[x] := d[x];
P.Empiler(x);
pour chaque (x, y) ∈ A faire
    si Couleur[y] = blanc alors
        CFC(y);
        r_A[x] := min(r_A[x], r_A[y]);
    sinon
        si d[y] < d[x] ∧ y ∈ P alors
            r_A[x] := min(r_A[x], d[y]);
temps++; f[x] := temps; Couleur[x] := noir;
si r_A[x] = d[x] alors
    nbcfc++;
    tant que P ≠ ∅ ∧ d[P.Tete()] ≥ d[x] faire
        y := P.Tete();
        P.Depiler();
        NumCFC[y] := nbcfc;
```

Comment ça marche ?

Il faut montrer que les coefficients $r_A[-]$ calculés par l'algorithme sont égaux aux $r[-]$ définis précédemment.

Quelques éléments. . .

Comment ça marche ?

Il faut montrer que les coefficients $\mathrm{r}_A[-]$ calculés par l'algorithme sont égaux aux $\mathrm{r}[-]$ définis précédemment.

Quelques éléments. . .

- P contient les sommets dont la racine de leur CFC est encore grise. Ils ont des ancêtres communs dans G_Π.
  Les sommets de $\mathcal{C}$ ne sont extraits de P que lorsque $u_\mathcal{C}$ est coloriée en noir et que les instructions 14-18 de `CFC` sont exécutées.

Comment ça marche ?

Il faut montrer que les coefficients $\mathrm{r}_A[-]$ calculés par l'algorithme sont égaux aux $\mathrm{r}[-]$ définis précédemment.

Quelques éléments...

- P contient les sommets dont la racine de leur CFC est encore grise. Ils ont des ancêtres communs dans G_Π.
  Les sommets de $\mathcal{C}$ ne sont extraits de P que lorsque $u_\mathcal{C}$ est coloriée en noir et que les instructions 14-18 de `CFC` sont exécutées.
- l'instruction 8 permet de faire "remonter" les valeurs de $\mathrm{r}_A[y]$ à son père x dans G_Π.

Comment ça marche ?

Il faut montrer que les coefficients $\mathsf{r}_A[-]$ calculés par l'algorithme sont égaux aux $\mathsf{r}[-]$ définis précédemment.

Quelques éléments. . .

- P contient les sommets dont la racine de leur CFC est encore grise. Ils ont des ancêtres communs dans G_Π.
  Les sommets de $\mathcal{C}$ ne sont extraits de P que lorsque $u_\mathcal{C}$ est coloriée en noir et que les instructions 14-18 de `CFC` sont exécutées.
- l'instruction 8 permet de faire "remonter" les valeurs de $\mathsf{r}_A[y]$ à son père x dans G_Π.
- le test de la ligne 10 ("$\mathsf{d}[y] < \mathsf{d}[x] \;\wedge\; y \in P$") caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow^*_G y$? . . .

le test "$\mathrm{d}[y] < \mathrm{d}[x] \ \wedge \ y \in P$" caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow^*_G y$.

le test “$\mathsf{d}[y] < \mathsf{d}[x] \ \wedge \ y \in P$” caractérise bien les arcs (x, y) de type “retour” ou “transverse” t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow^*_G y$.

- (x, y) : $\notin G_\Pi$ (y n'est pas blanc) et pas arc avant ($\mathsf{d}[y] < \mathsf{d}[x]$) ;

le test "$d[y] < d[x] \ \wedge \ y \in P$" caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow^*_G y$.

- $(x, y) : \notin G_\Pi$ (y n'est pas blanc) et pas arc avant ($d[y] < d[x]$) ;
- $x, y \in P \ \Rightarrow$ ils ont des ancêtres communs dans G_Π ;

le test "$\mathrm{d}[y] < \mathrm{d}[x] \ \wedge \ y \in P$" caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow^*_G y$.

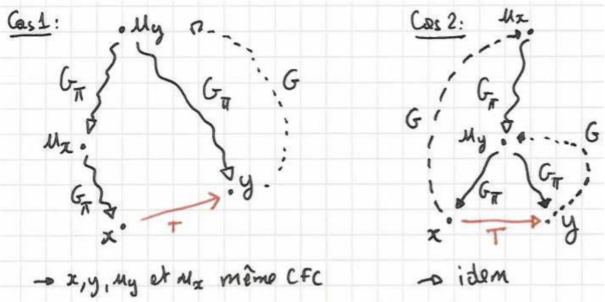
- $(x, y) : \notin G_\Pi$ (y n'est pas blanc) et pas arc avant ($\mathrm{d}[y] < \mathrm{d}[x]$) ;
- $x, y \in P \ \Rightarrow$ ils ont des ancêtres communs dans G_Π ;
- $x \leftrightarrow^*_G y$:

le test "$d[y] < d[x] \ \wedge \ y \in P$" caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow^*_G y$.

- $(x, y) : \notin G_\Pi$ (y n'est pas blanc) et pas arc avant ($d[y] < d[x]$) ;
- $x, y \in P \ \Rightarrow$ ils ont des ancêtres communs dans G_Π ;
- $x \leftrightarrow^*_G y$:
  - direct si (x, y) est un arc retour...

le test "$\mathrm{d}[y] < \mathrm{d}[x] \ \wedge \ y \in P$" caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow^*_G y$.

- $(x, y) : \notin G_\Pi$ (y n'est pas blanc) et pas arc avant ($\mathrm{d}[y] < \mathrm{d}[x]$) ;
- $x, y \in P \ \Rightarrow$ ils ont des ancêtres communs dans G_Π ;
- $x \leftrightarrow^*_G y$:
  - direct si (x, y) est un arc retour...
  - si (x, y) est un arc transverse, soient u_x et u_y les racines (en gris car dans P) de leurs CFC...



Correction de l'algorithme de Tarjan

Théorème

L'algorithme de Tarjan, vérifie les propriétés suivantes :

1. à la date $\mathrm{f}[x]$, on a : $\mathrm{r}_A[x] = \mathrm{r}[x]$,
2. à la fin du traitement $\mathtt{CFC}(u_{\mathcal{C}})$, tous les sommets y de $\mathcal{C}$ sont retirés de la pile P et vérifient : $NumCFC[y] = NumCFC[u_{\mathcal{C}}]$,
3. à la fin de $\mathtt{Tarjan\text{-}CFC}(G)$, chaque CFC a reçu un numéro différent : pour deux CFC distinctes $\mathcal{C}$ et $\mathcal{C}'$, on a $NumCFC[u_{\mathcal{C}}] \neq NumCFC[u_{\mathcal{C}'}]$.