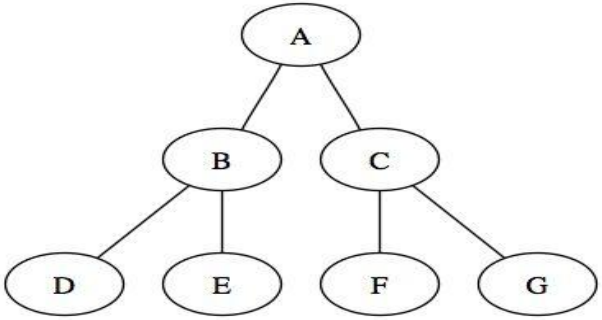
Chapitre III: Arbres Equilibrés

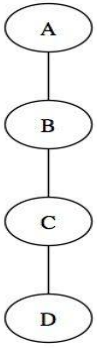
Chapitre III: Arbres Equilibrés

Complexité

Les opérations d'insertion, de recherche et de suppression dans un ABR ont une complexité en : O $log2n$ dans les cas où l'arbre n'est pas trop déséquilibré, O n lorsque l'arbre est dégénéré en liste.



Arbre complet



Arbre dégénéré

Chapitre III: Arbres Equilibrés

Solution

➢ Pour pallier une telle perte de performance, une manière de procéder consiste à assurer qu'un arbre de recherche est toujours équilibré, c.-à-d. qu'après toute opération modifiant l'arbre, un rééquilibrage est effectué si cette opération a conduit à un déséquilibre.

➢ Ce rééquilibrage consiste en une ou plusieurs rotation appliquées à un ou plusieurs sous arbre bien choisis de l'arbre.

Chapitre III: Arbres Equilibrés

Equilibre d'un ABR

- L'équilibre d'un arbre binaire est un entier qui vaut 0 si l'arbre est vide et la déférence des hauteurs des sous-arbres gauche et droit de l'arbre sinon.
- Un arbre binaire est équilibré lorsque l'équilibre de chacun de ses sous-arbres non vides n'excède pas 1 en valeur absolue.

Chapitre III: Arbre Equilibrés

Objectifs d'équilibrage des ABR

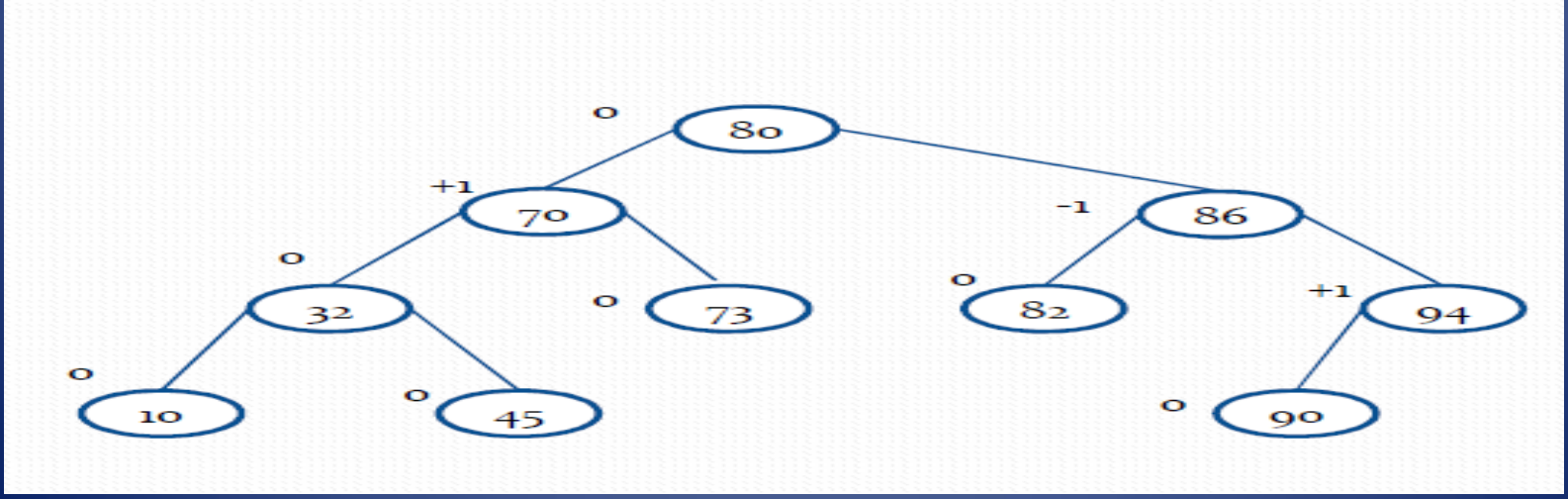
- Eviter de construire des arbres dits *dégénérés* (*arbres peignes*), dans lesquels certains chemins d'accès aux données sont d'une longueur disproportionnée.

- Très avantageux lorsqu'on réutilise l'arbre ainsi construit : minimisation considérable de temps de recherche

Chapitre III: Arbre Equilibrés

Facteur d'équilibrage

Chaque nœud d'un arbre binaire contient un entier appelé *facteur d'équilibrage* qui permet de déterminer si il est nécessaire de rééquilibrer l'arbre.



Chapitre III: Arbre Equilibrés

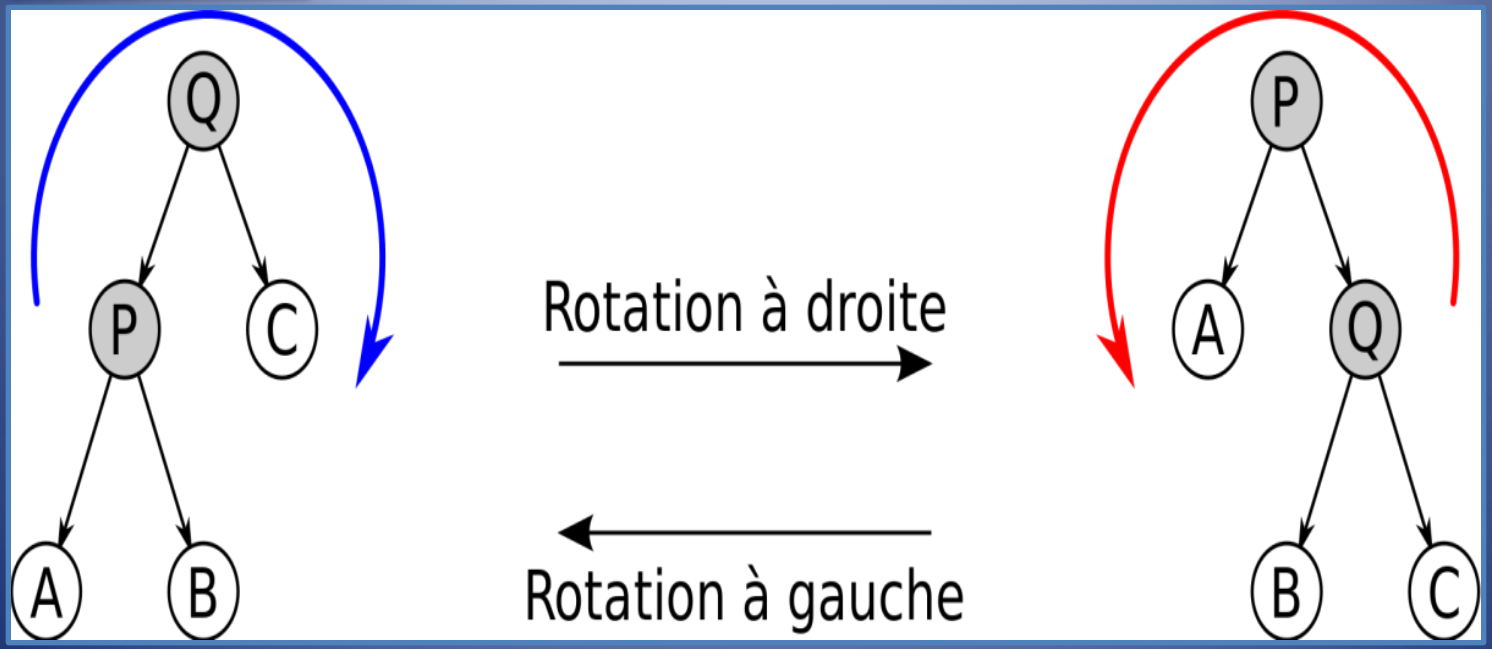
Rotation

- Une rotation consiste en remonter un nœud dans l'arbre et à en faire redescendre un autre. Elle permet de changer la structure d'un ABR sans invalider l'ordre des éléments

- Une rotation droite autour du sommet y d'un arbre binaire de recherche consiste à faire descendre le sommet y et à faire remonter son fils gauche x sans invalider l'ordre des éléments. L'opération inverse s'appelle rotation gauche autour du sommet x

Chapitre III: Arbre Equilibrés

Exemple



Chapitre III: Arbre Equilibrés

Rotation (Contraintes)

- la première contraintes à respectée lors d'une rotation est de Garder l'ordre des feuilles de l'arbre lorsqu'elles sont lues de gauche à droite par exemple, de tel sorte que lors d'un parcours en profondeur , l'ordre de visite des feuilles est le même qu'avant l'opération.

.

- L'autre contrainte est la propriété principale d'un ABR , c'est-à-dire que le fils gauche est plus petit que son père et que le fils droit est plus grand que son père

Chapitre III: Arbres Equilibrés

Rotation (objectifs)

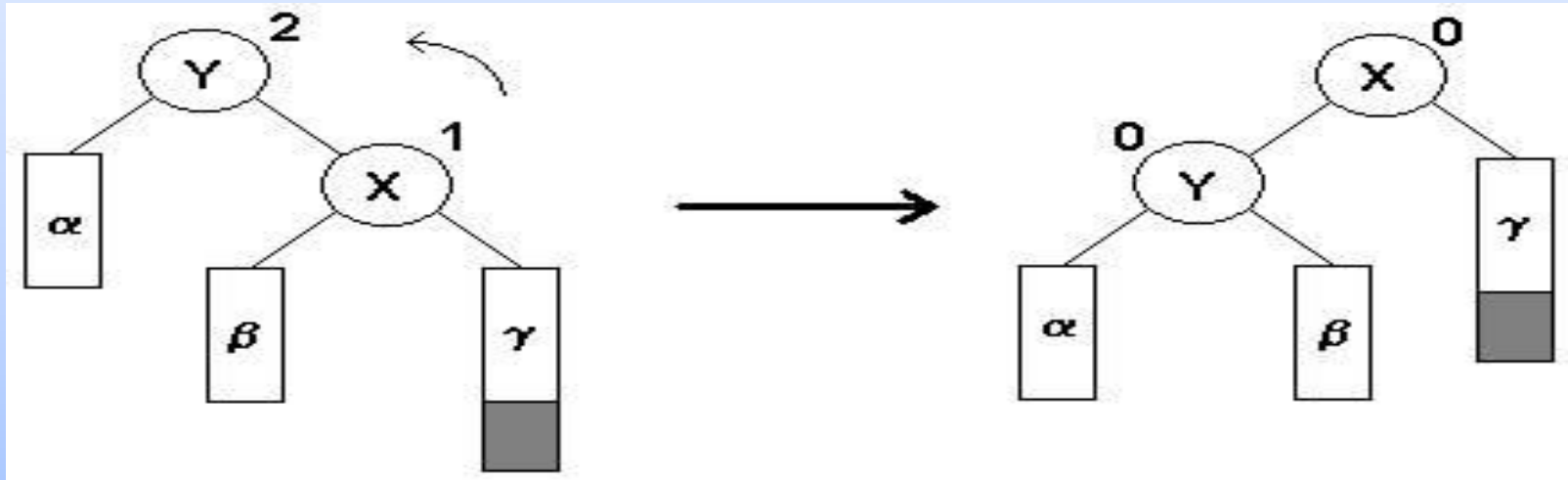
- L'objectif principal d'une rotation d'arbre est de garder un ABR équilibrés après chaque opération de suppression ou d'insertion d'un élément dans l'arbre.

- Le sens et le nombres de rotations nécessaires pour rééquilibrer un ABR dépend des valeurs des facteurs d'équilibrage d'un nœud et ainsi que ces fils droit et gauche.

Chapitre III: Arbre Equilibrés

Rotation simple gauche

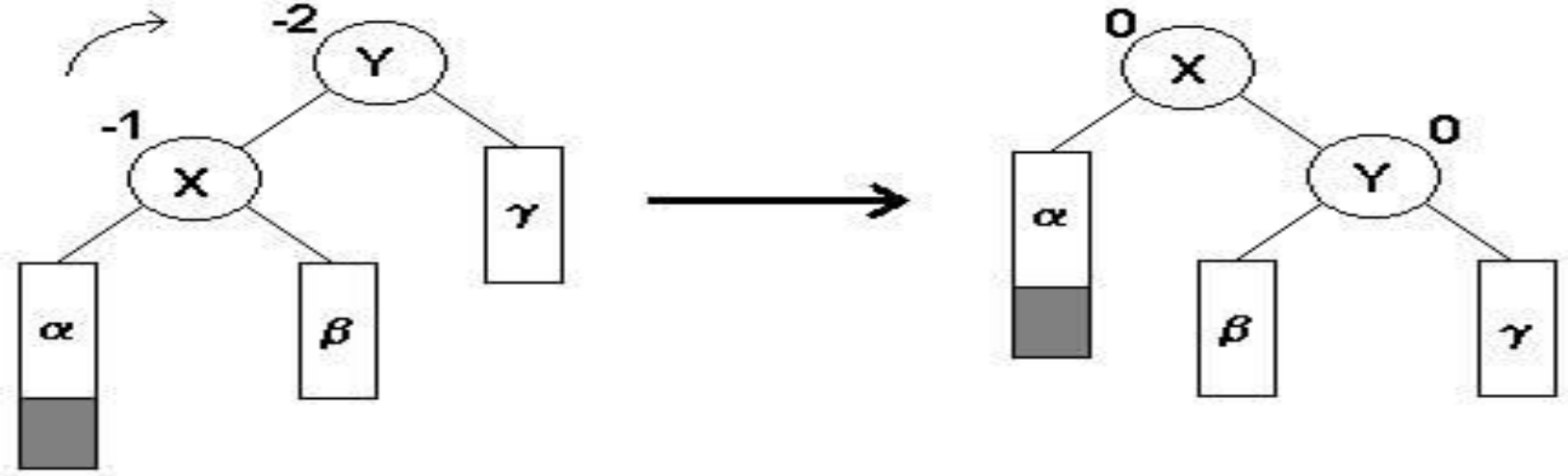
Cas où le nœud a une différence de profondeur des sous-arbre de 2 et le fils droit une différence de 1



Chapitre III: Arbre Equilibrés

Rotation simple droite

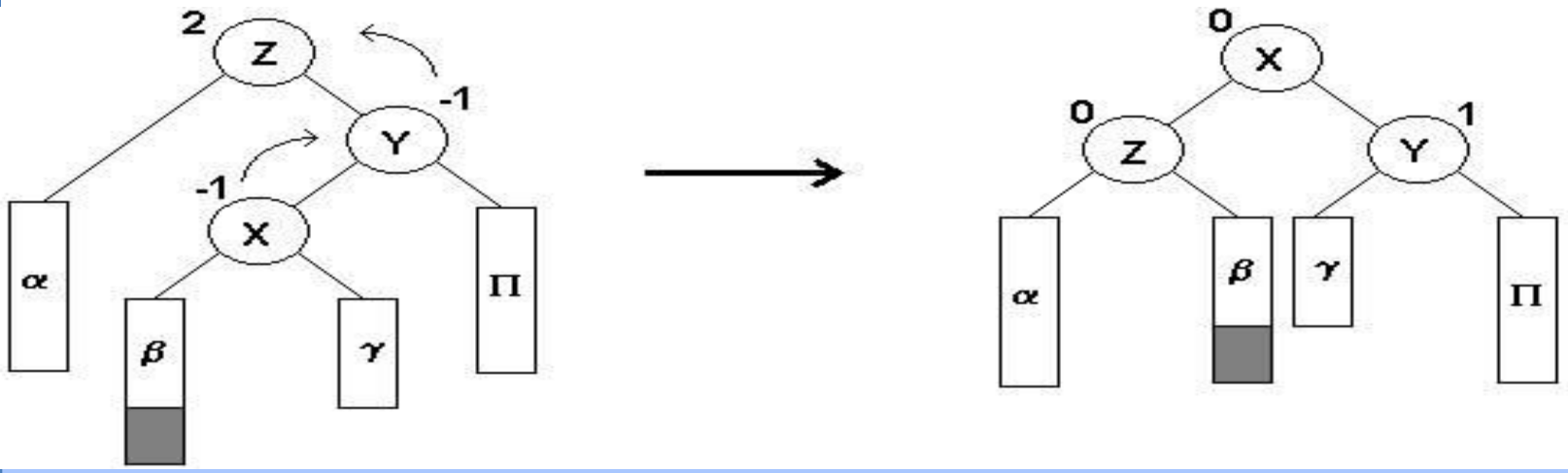
Cas où le nœud a une différence de profondeur des sous-arbre de -2 et le fils gauche une différence de -1



Chapitre III: Arbre Equilibrés

Rotation double droite gauche

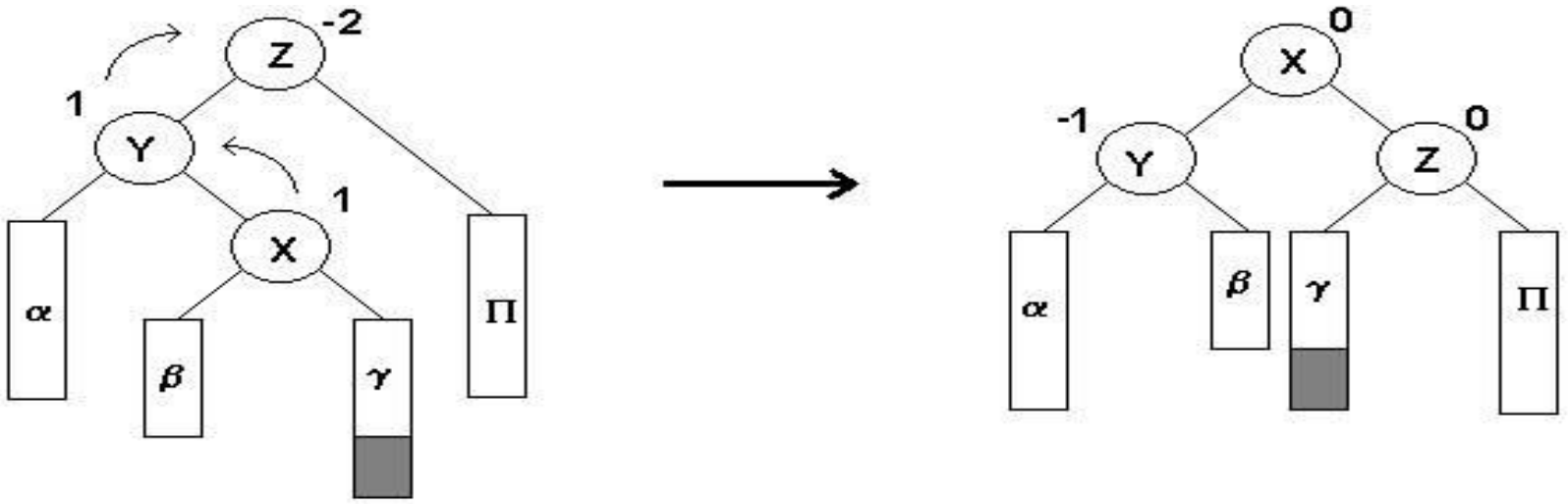
Cas où le nœud a une différence de profondeur des sous-arbre de 2 et le fils droit une différence de -1



Chapitre III: Arbre Equilibrés

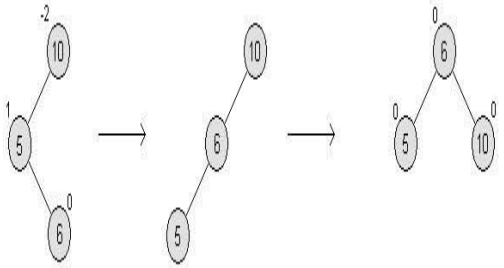
Rotation double gauche droite

Cas où le nœud a une différence de profondeur des sous-arbre de -2 et le fils gauche une différence de 1

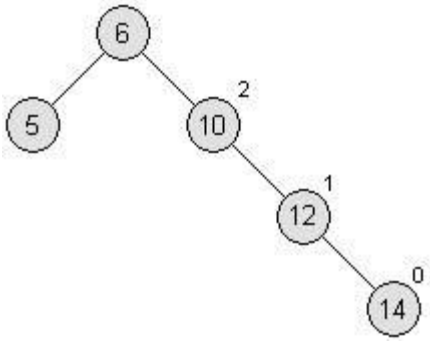


Chapitre III: Arbre Equilibrés

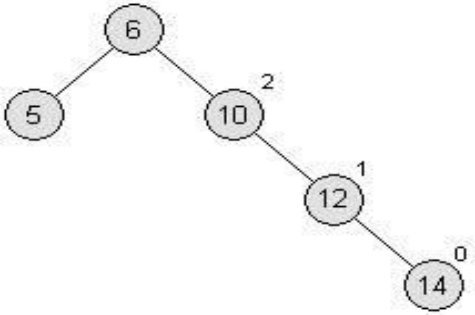
Cas –2, 1 : Rotation double gauche droite



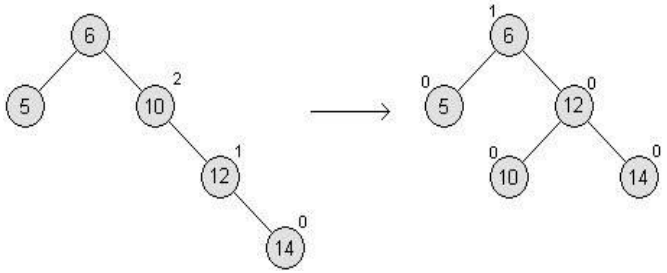
Ajout de la valeur 12



Ajout de la valeur 14

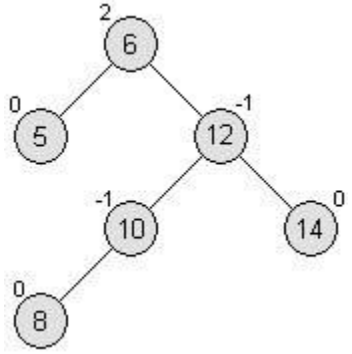


Cas 2, 1 : Rotation simple gauche

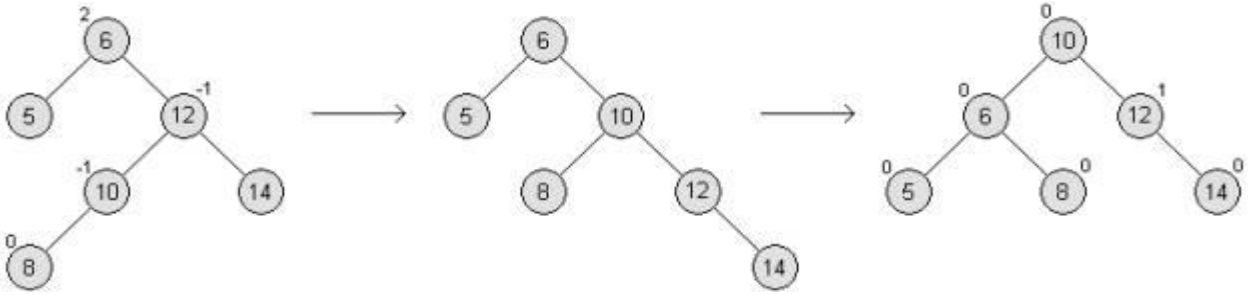


Chapitre III: Arbre Equilibrés

Ajout de la valeur 8



Cas 2, -1 : Rotation double droite gauche



Chapitre III: Arbre Equilibrés

Implémentation

- Quand un sous-arbre subit une rotation, la hauteur de l'un de ses côtés augmente et celle de l'autre côté diminue. Ceci rend les rotations utiles pour équilibrer un arbre.

- Algorithme de rotation ou de rééquilibrage d'une ABR nécessite d'ajouter un champs supplémentaire pour stocker la hauteur de chaque nœud .
- A chaque opération (insertion ou suppression) recalcule le facteur d'équilibre de chaque sous arbres pour .

Chapitre III: Arbre Equilibrés

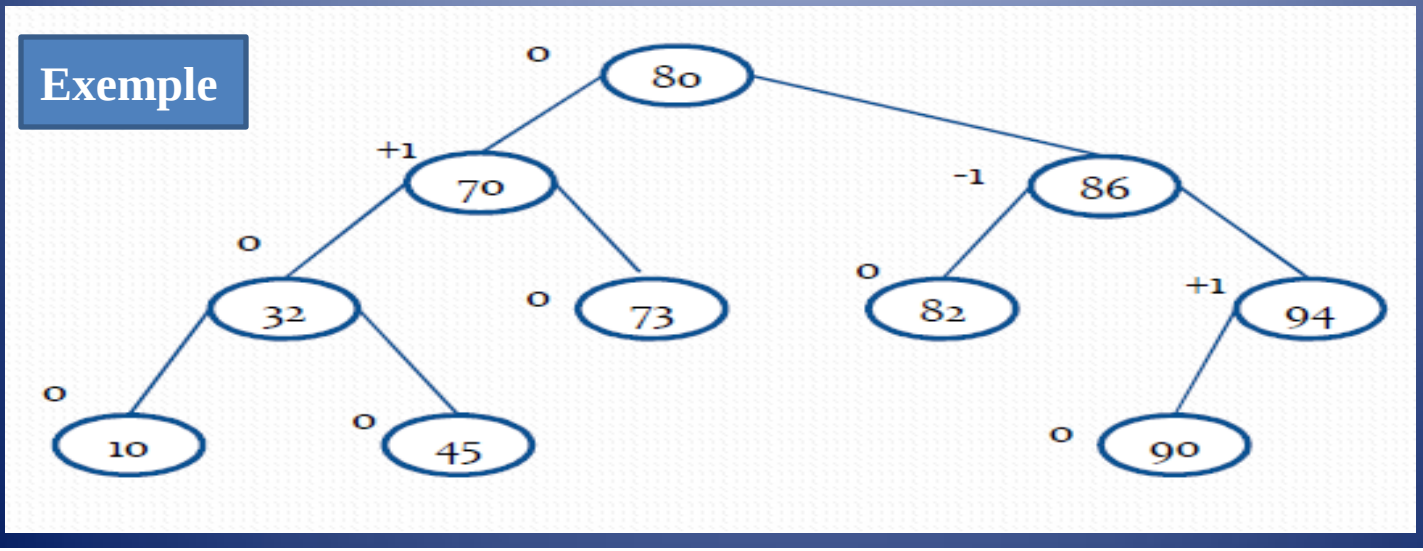
Les Arbres Equilibrés AVL

Chapitre III: Arbre Equilibrés

Arbres binaires équilibrés AVL

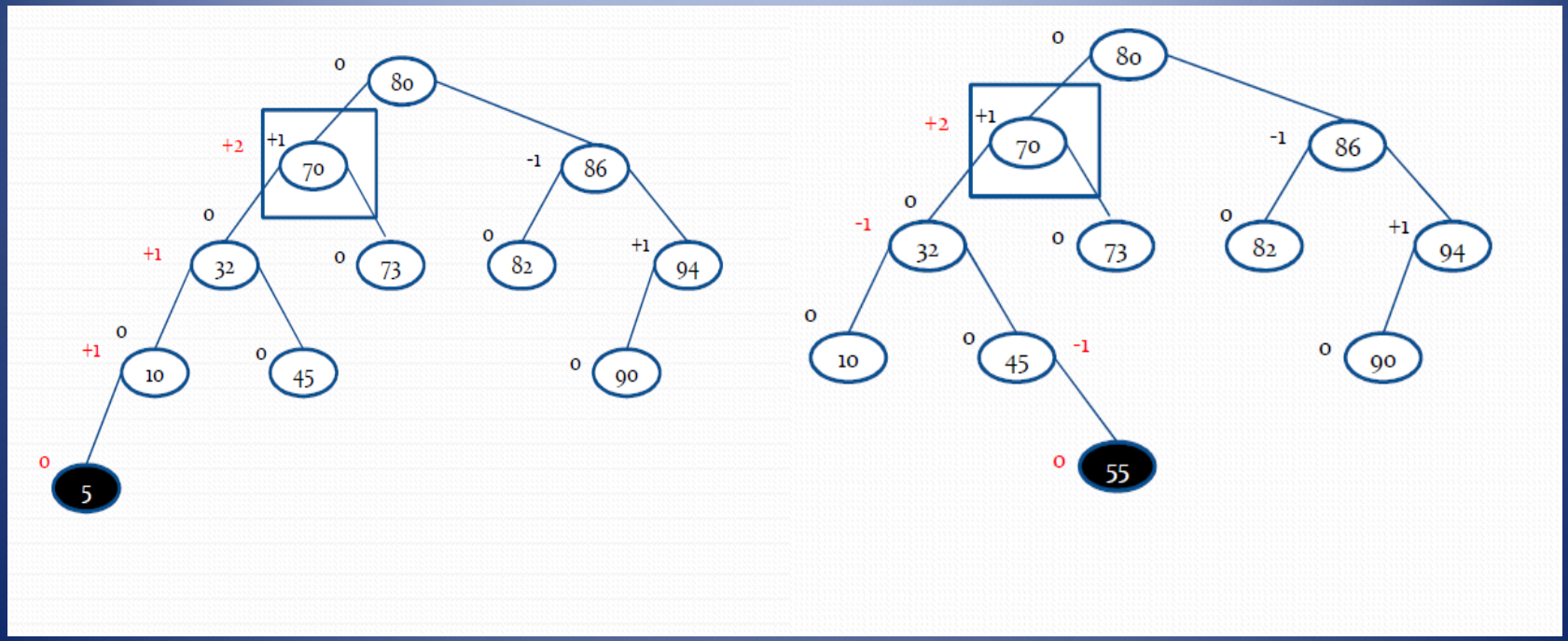
- Arbres AVL Un arbre AVL est un arbre de recherche binaire équilibré :
  | Profondeur(fg(n))– Profondeur(fd(n)) | <= 1
- Ajouter un champ balance (facteur d'équilibrage) au niveau de chaque nœud

Exemple



Chapitre III: Arbre Equilibrés

Arbres AVL (Cas de déséquilibre)



Chapitre III: Arbre Equilibrés

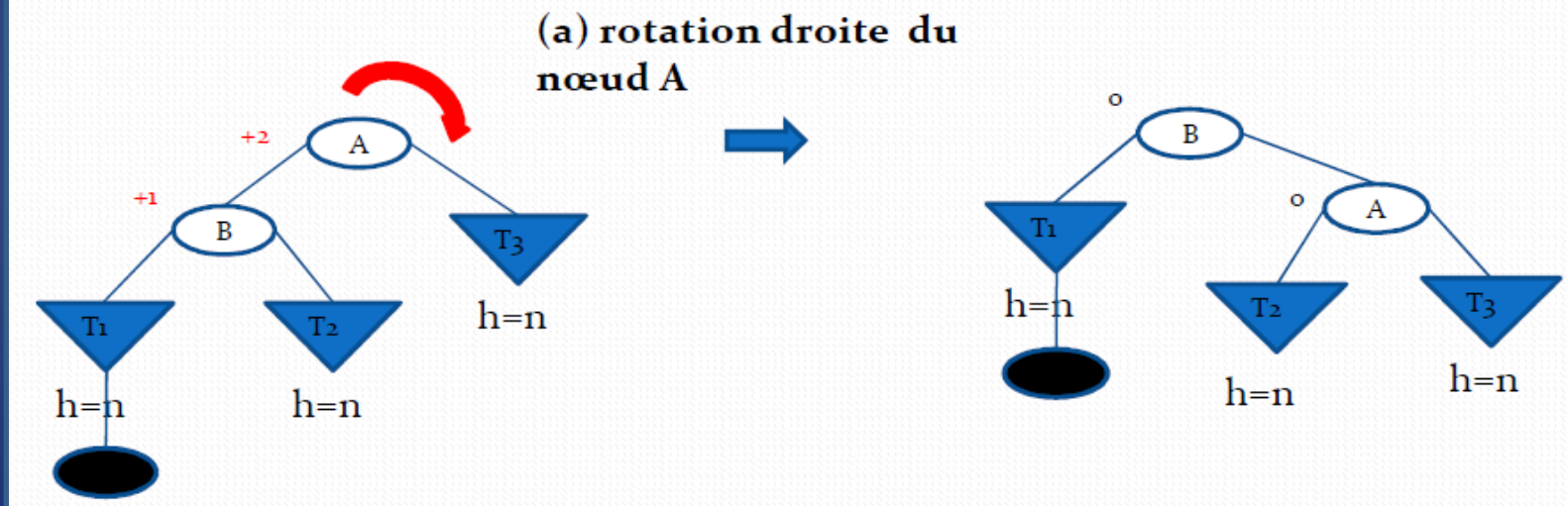
Techniques d'équilibrage des AVL

- Examinons un sous arbre de racine le plus jeune antécédent qui devient non équilibré suite à une insertion .

- Transformer l'arbre de telle sorte que:
  - l'ordre soit préservé
  - l'arbre transformé soit équilibré

Chapitre III: Arbre Equilibrés

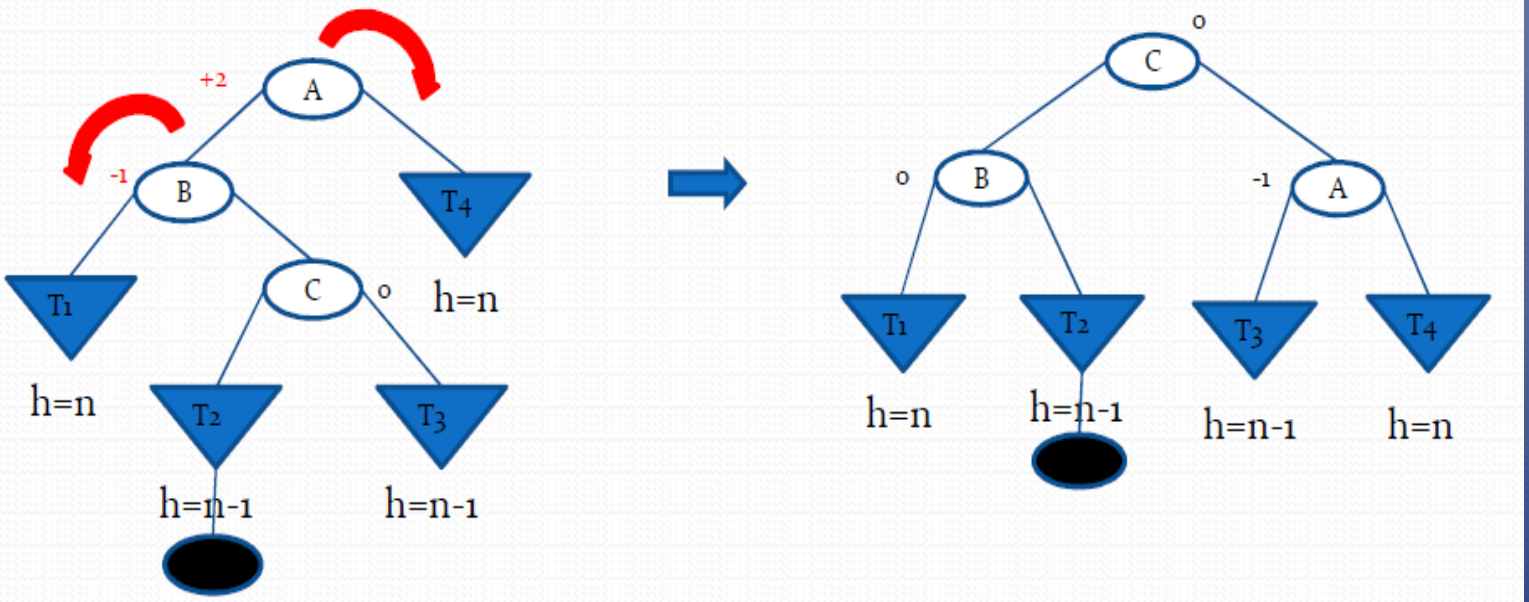
Cas où le facteur d'équilibrage est +1



Le nouveau nœud est inséré dans le sous arbre gauche de B. Donc f(B) devient 1 et f(A) devient 2

Chapitre III: Arbre Equilibrés

Cas où le facteur d'équilibrage est +1



Le nouveau nœud est inséré dans le sous arbre droit de B. f(B) devient -1 et f(A) devient 2.

Chapitre III: Arbre Equilibrés

Algorithme MAJ dans un AVL

comme dans un arbre de recherche binaire ordinaire :

- ***Etape1***:Insérer (ou supprimer)la clé dans l'arbre sans tenir compte du facteur d'équilibrage mais garder la trace du plus jeune antécédent, qui devient non équilibré

- ***Etape 2*** : Fait la transformation à partir de Y (les rotation suivant les cas vus dans le chapitre précédent).

Chapitre III: Arbre Equilibrés

Arbres AVL (Analyse théorique)

- la profondeur maximale d'un arbre binaire équilibré est 1.44*Log2n La recherche dans un tel arbre n'exige jamais plus de 44% de plus de comparaisons que pour un arbre binaire complet ;

- Operations de maintenance :
  - Restructuration = 1 rotation ou double rotation
  - Insertion : au plus 1 restructuration
  - suppression : au plus Log2 (N) restructurations

Chapitre III: Arbre Equilibrés

Arbres bicolores (Rouge – Noir)

Chapitre III: Arbre Equilibrés

Arbre Rouge Noir

Définition

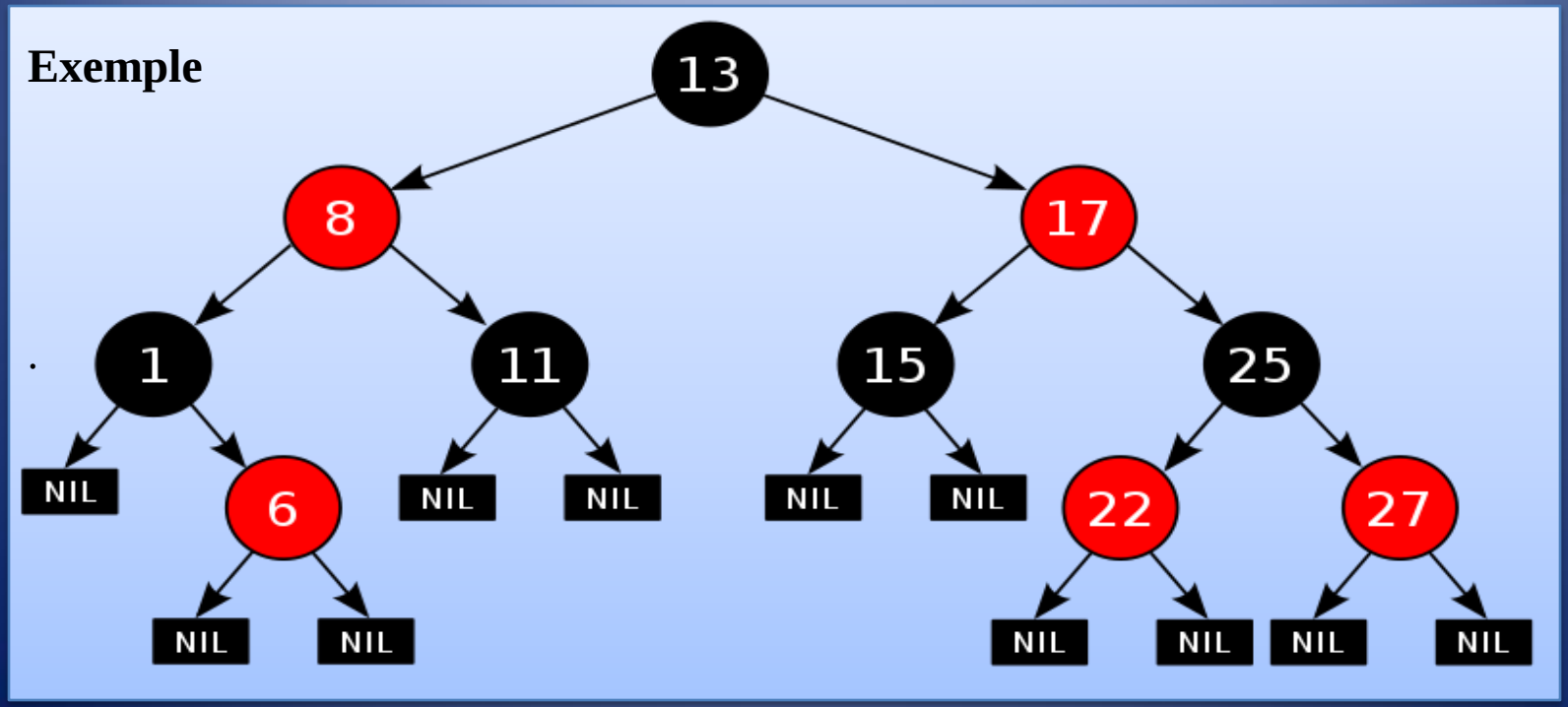
Un arbre rouge et noir est un arbre binaire de recherche qui satisfait les propriétés suivantes :

1. Chaque nœud est soit rouge, soit noir.
2. Chaque feuille (Nil) est noire.
3. Si un nœud est rouge alors ses deux fils sont noirs.
4. Tous les chemins descendants qui relie un nœud donné à une feuille (du sous-arbre dont il est la racine) contiennent le même nombre de nœuds noirs.

Chapitre III: Arbre Equilibrés

Arbre Rouge Noir

Exemple



Chapitre III: Arbre Equilibrés

Opérations sur les arbres Rouge noir

Toute opération (suppression /insertion) l'arbre sur un arbre rouge noire on doit respectée que l'arbre reste:

- un arbre binaire de recherche
- un arbre rouge noir

En plus des techniques de rééquilibrage utilisées pour les ABR on doit avoir des méthode de correction de couleurs pour que l'arbre reste un arbre rouge et noir après chaque manipulation.

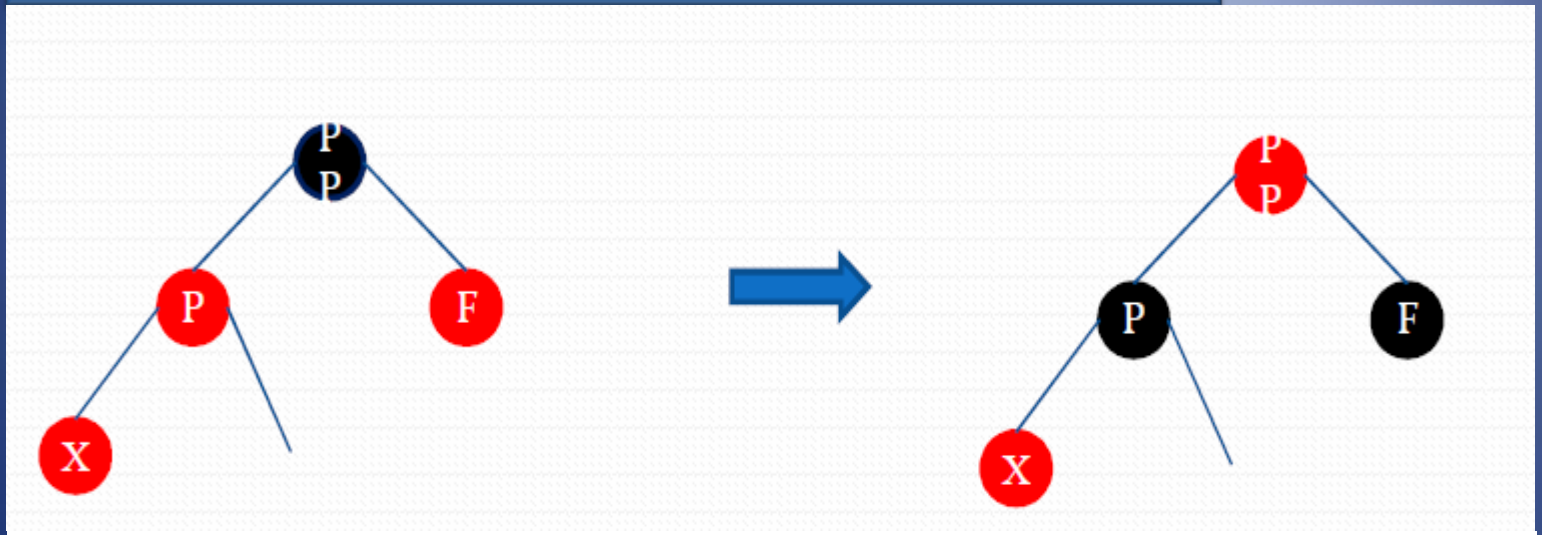
Chapitre III: Arbre Equilibrés

Insertion d'un nœud dans arbres Rouge noir

- L'insertion comme dans un arbre de recherche binaire.
- Le nœud inséré est toujours une feuille On lui attribue la couleur rouge
- Si son père est aussi rouge, un algorithme de maintenance est appliqué

Chapitre III: Arbre Equilibrés

CAS 1: le frère F de P est rouge

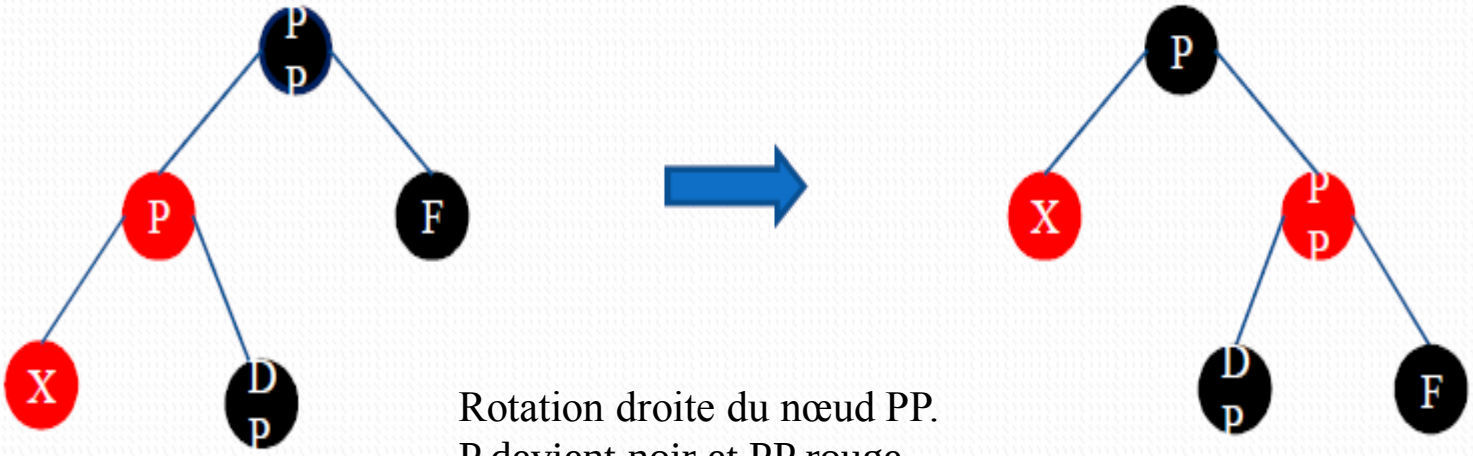


x : est le nœud introduit

Les nœuds P et F deviennent noirs et leur père PP devient rouge

Chapitre III: Arbre Equilibrés

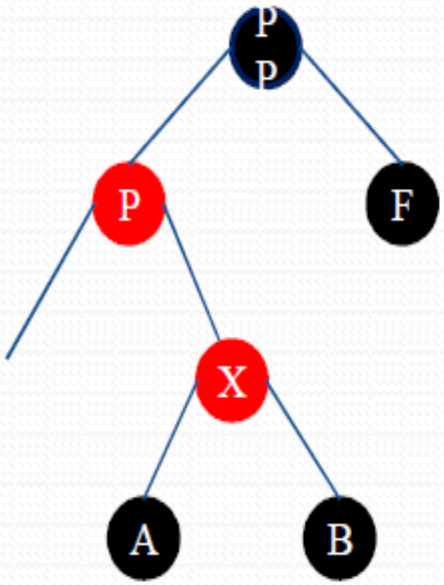
CAS 2: le frère F de P est noir et X est le fils gauche de P.



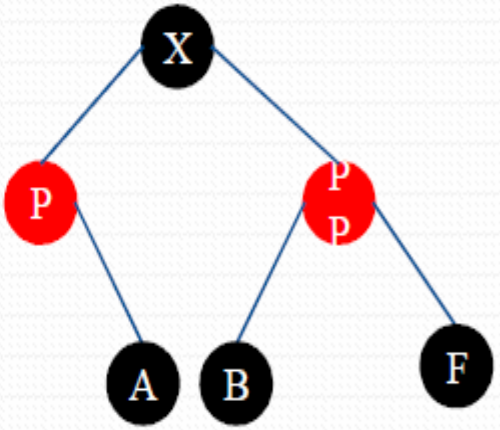
Rotation droite du nœud PP.
P devient noir et PP rouge

Chapitre III: Arbre Equilibrés

CAS 3: le frère F de P est noir et X est le fils droit de P



Rotation gauche du nœud P +
rotation droite du nœud PP.
X devient noir et PP rouge.



Chapitre III: Arbre Equilibrés

CAS 4: le nœud père P est la racine de l'arbre

Le nœud père devient noir

C'est le seul cas où la hauteur noire de l'arbre augmente.

Chapitre III: Arbre Equilibrés

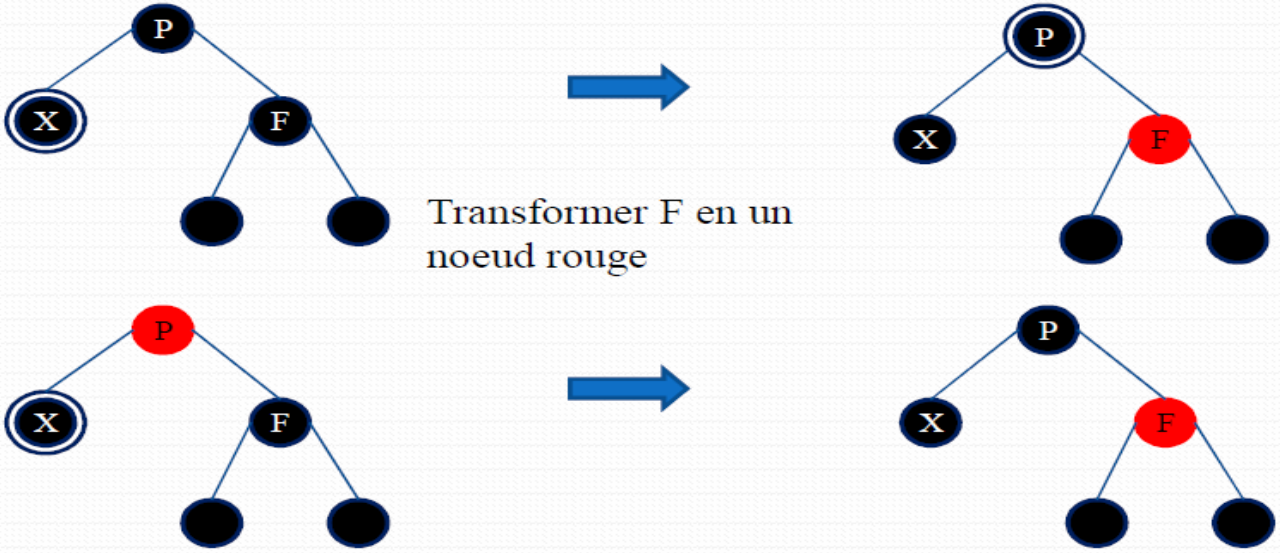
suppression d'un nœud d'un arbres Rouge noir

- Suppression comme dans un arbre de recherche binaire.
- Si le nœud physiquement supprimé est noir, un algorithme de maintenance est appliqué.
- On considère que le nœud qui remplace le nœud supprimé porte une couleur noire en plus.
- Ceci signifie qu'il devient noir s'il est rouge et qu'il devient doublement noir s'il est déjà noir.
- L'algorithme de maintenance a donc pour rôle de supprimer ce nœud doublement noir.

Chapitre III: Arbre Equilibrés

CAS 1: Le frère F de X est noir et a deux fils noirs

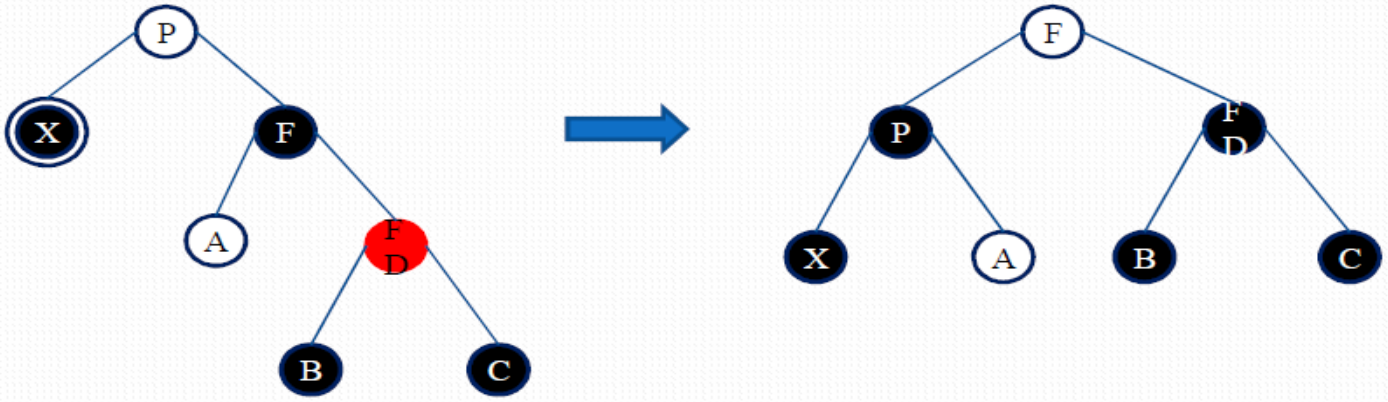
Soit X le noeud doublement noir (Supposé ici comme un fils gauche) et P son père



Si le parent P est noir, le processus continue en remontant dans l'arbre. X devient le nouveau noeud doublement noir.

Chapitre III: Arbre Equilibrés

CAS 2: Le frère F du nœud X est noir et a un fils droit rouge (FD)

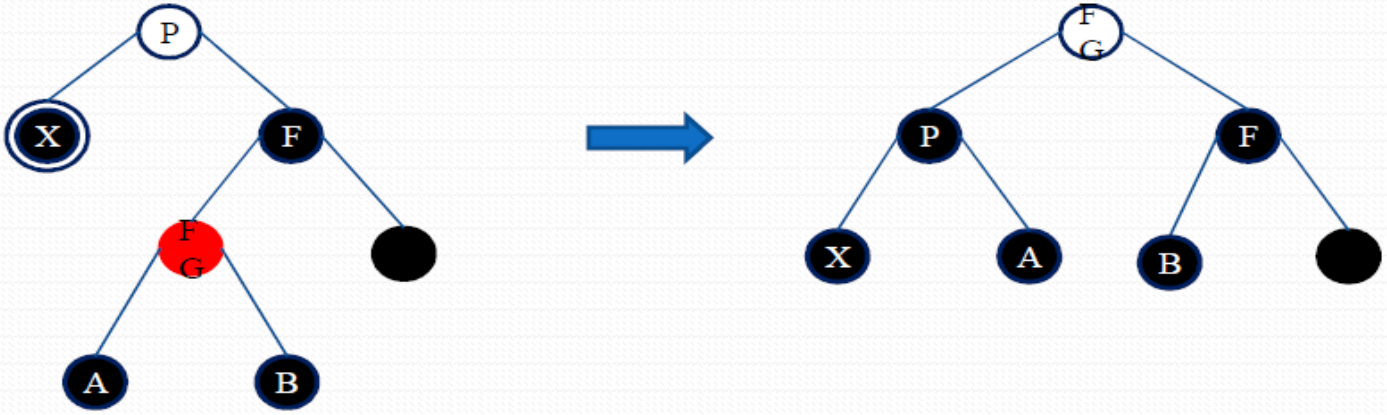


Rotation gauche de P

Recolorer : P et FD deviennent noirs et la couleur de F est celle de P avant la transformation.

Chapitre III: Arbre Equilibrés

CAS 3: si le frère F du nœud X est noir et a un fils gauche rouge(FG)



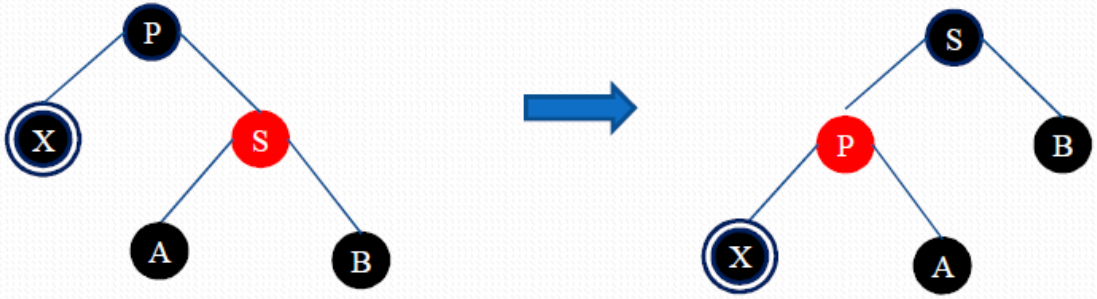
Rotation droite (F) + rotation gauche (P).

Recolorer : P devient noir et la couleur de FG est celle de P avant la transformation.

Le processus se termine

Chapitre III: Arbre Equilibrés

CAS 4: Le frère F de X est rouge



Rotation droite(P)

Le processus continue selon le cas 1, 2 ou 3

Chapitre III: Arbre Equilibrés

CAS 0: X est la racine de l'arbre

Soit X le noeud doublement noir

X devient simplement un noeud noir.

C'est le seul cas ou la hauteur de l'arbre diminue. Le processus se termine.

Chapitre III: Arbre Equilibrés

Arbre Rouge et Noir (analyse théorique)

- Operations de maintenance :
- Restructuration et coloration.
- Insertion : au plus 1 restructuration et au plus Log2 (n) colorations.
- Suppression : au plus 2 restructurations et au plus Log2 (n) colorations.

N : nombre d'éléments insérés.

la profondeur maximale d'un arbre binaire équilibré est 2*Log2 (n) Recherche, insertion et suppression : O(Log2 (n))