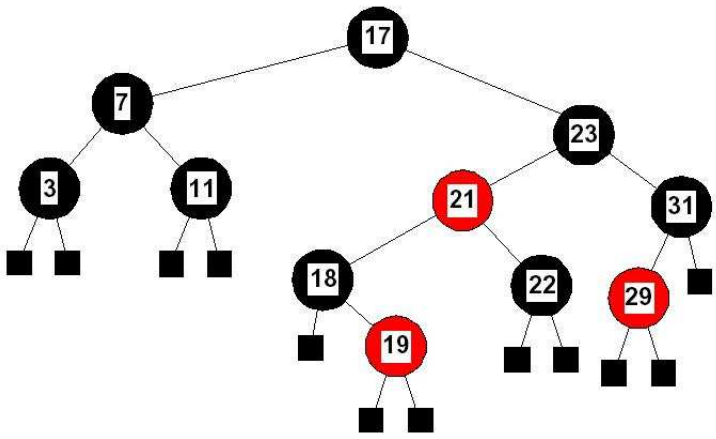
Arbres rouges et noirs

1 Definitions

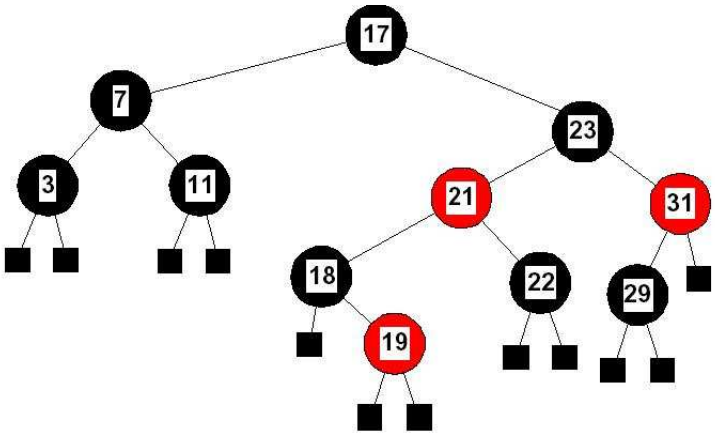
Un arbre rouge et noir est un arbre GRD où chaque nœud est de couleur rouge ou noire de telle sorte que

1. les enfants d'un nœud rouge sont noirs,
2. le nombre de nœuds noirs le long d'une branche de la racine à une feuille est indépendant de la branche. Autrement dit, pour tout nœud de l'arbre, les chemins de ce nœud vers les feuilles (nœud sentinelle) qui en dépendent ont le même nombre de nœuds noirs.

Cet arbre est un arbre rouge et noir :

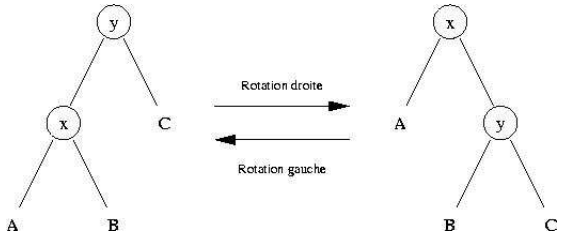


Cet arbre n'est pas un arbre rouge et noir (à cause de la branche droite du nœud 31 qui ne contient pas le même nombre de noirs que les autres branches) :

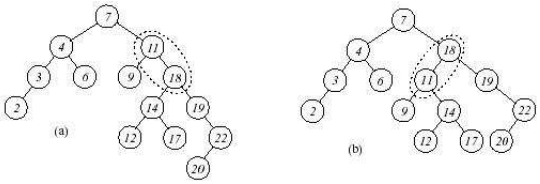


Un arbre rouge et noir est donc un arbre GRD “presque” équilibré
La première condition stipule que les nœuds rouges ne sont pas trop nombreux. La dernière condition est une condition d'équilibre. Elle signifie que si on oublie les nœuds rouges d'un arbre on obtient un arbre binaire parfaitement équilibré. En contrôlant cette information de couleur dans chaque nœud, on garantit qu'aucun chemin ne peut être deux fois plus long que n'importe quel autre chemin, de sorte que l'arbre reste équilibré.

Les rotations Les rotations sont des modifications locales d'un arbre binaire. Elles consistent à échanger un nœud avec l'un de ses fils. Dans la rotation droite, un nœud devient le fils droit du nœud qui était son fils gauche. Dans la rotation gauche, un nœud devient le fils gauche du nœud qui était son fils droit. Les rotations gauche et droite sont inverses l'une de l'autre. Elles sont illustrées par la figure ci-dessous. Dans les figures, les lettres majuscules comme A, B et C désignent des sous-arbres.



La figure ci-dessous montre comment une rotation permet de rééquilibrer un arbre.



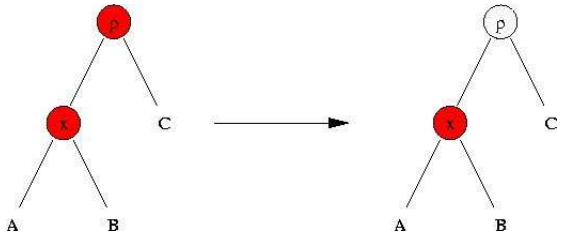
Rééquilibrage d'un arbre par une rotation. Une rotation gauche sur le nœud de clé 11 de l'arbre (a) conduit à un arbre (b) mieux équilibré : la hauteur de l'arbre est passée de 5 à 4.

2 Insertion d'une valeur

L'insertion d'une valeur dans un arbre rouge et noir commence par l'insertion usuelle d'une valeur dans un arbre binaire de recherche. Le nouveau nœud devient rouge de telle sorte que la propriété (3) reste vérifiée. Par contre, la propriété (2) n'est plus nécessairement vérifiée. Si le père du nouveau nœud est également rouge, la propriété (2) est violée. Afin de rétablir la propriété (2), l'algorithme effectue des modifications dans l'arbre à l'aide de rotations. Ces modifications ont pour but de rééquilibrer l'arbre. Soit x le nœud et p son père qui sont tous les deux rouges. L'algorithme distingue plusieurs cas.

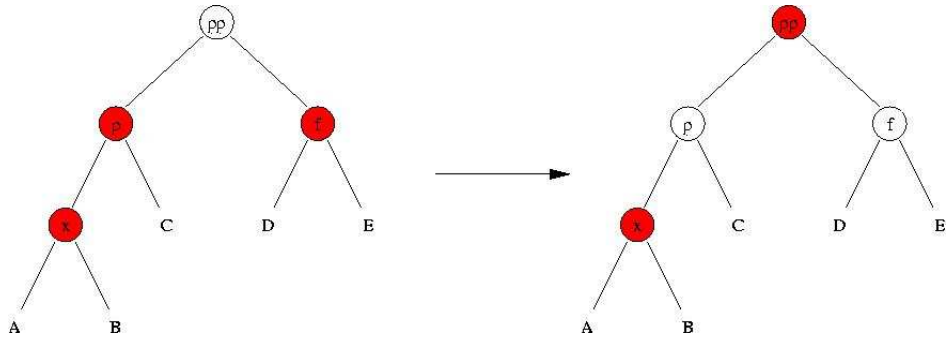
Cas 0 : le nœud père p est la racine de l'arbre

Le nœud père devient alors noir. La propriété (2) est maintenant vérifiée et la propriété (3) le reste. C'est le seul cas où la hauteur noire de l'arbre augmente.



Cas 1 : le frère f de p est rouge

Les nœuds p et f deviennent noirs et leur père pp devient rouge. La propriété (3) reste vérifiée mais la propriété ne l'est pas nécessairement. Si le père de pp est aussi rouge. Par contre, l'emplacement des deux nœuds rouges consécutifs s'est déplacé vers la racine.

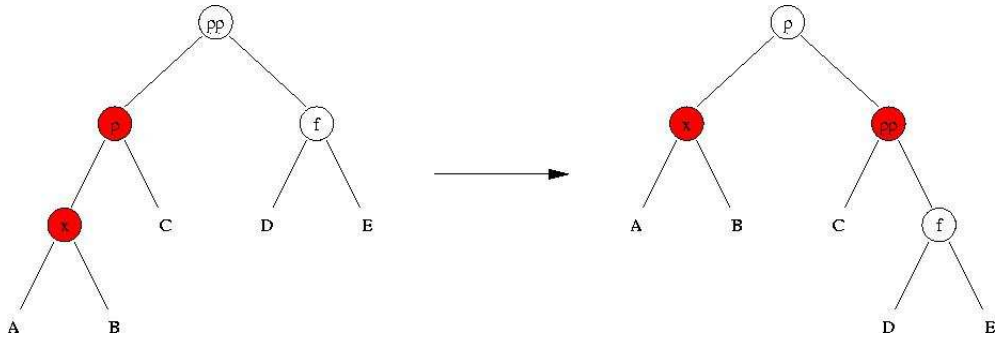


Cas 2 : le frère f de p est noir

Par symétrie on suppose que p est le fils gauche de son père (attention : ne pas oublier de prévoir l'autre cas, celui où p est le fils droit de son père). L'algorithme distingue à nouveau deux cas suivant que x est le fils gauche ou le fils droit de p.

Cas 2a : x est le fils gauche de p.

L'algorithme effectue une rotation droite entre p et pp. Ensuite le nœud p devient noir et le nœud pp devient rouge. L'algorithme s'arrête alors puisque les propriétés (2) et (3) sont maintenant vérifiées.



Cas 2b : x est le fils droit de p.

L'algorithme effectue une rotation gauche entre x et p de sorte que p devienne le fils gauche de x. On est ramené au cas précédent et l'algorithme effectue une rotation droite entre x et pp. Ensuite le nœud x devient noir et le nœud pp devient rouge. L'algorithme s'arrête alors puisque les propriétés (2) et (3) sont maintenant vérifiées.