Arbres Rouge - Noir

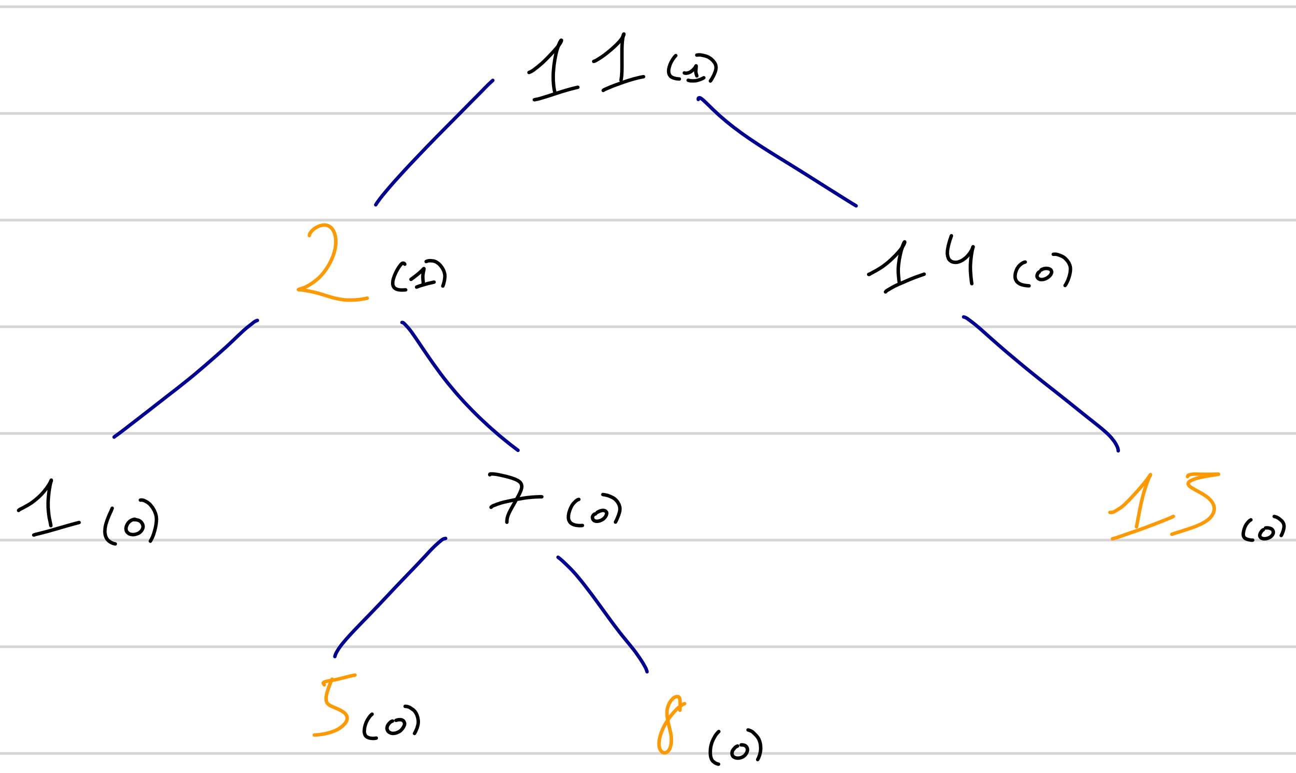
ABR $\left\{ \begin{array}{l} * \text{ Recherche} \\ * \text{ Minimum / Maximum / Successeur / ...} \\ * \text{ Insertion / Suppression} \end{array} \right\} O(h)$

Équilibré : $h = O(\log n)$

ABR rouge-noir : ABR dont chaque noeud se voit associer une couleur rouge ou noir et vérifiant :

→ Si un noeud est rouge, tous ses enfants éventuels sont noirs.

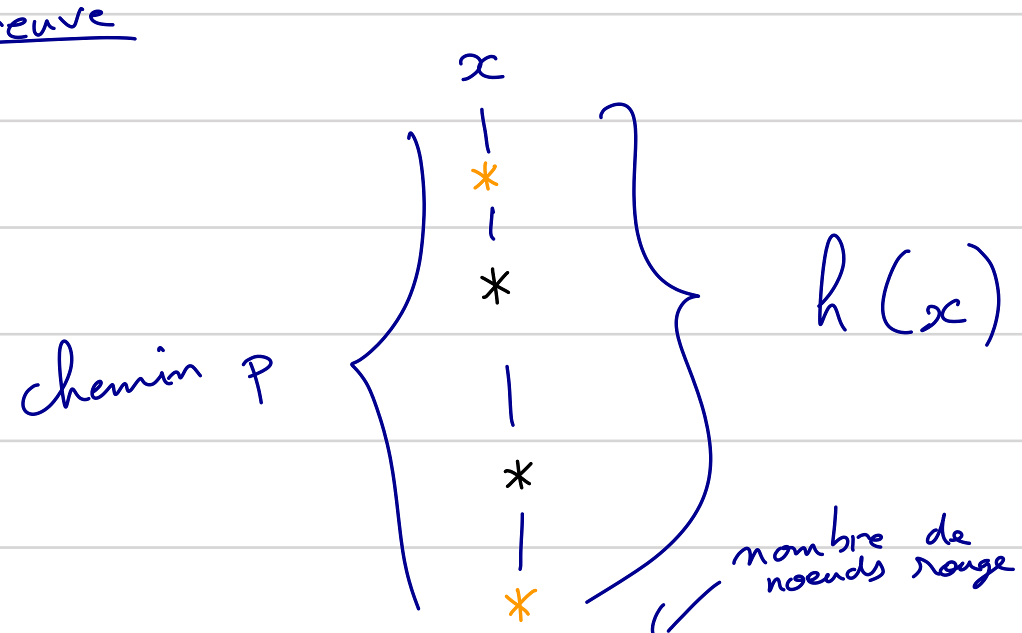
→ Même nombre de noeuds noirs sur tout chemin d'un noeud x de l'arbre à ses descendants (x inclus) ayant 0 ou 1 enfant.



<u>Définition</u> La <u>hauteur noire</u> (h_n) d'un noeud x (notée $h_n(x)$) est le nombre (unique) de noeuds noir sur tout chemin du noeud x à une feuille de l'arbre, le noeud x exclu.

__Proposition__ $\forall x,\ h_n(x) \geqslant \dfrac{h(x)-1}{2}$ (On a toujours bien sûr $h_n(x) \leqslant h(x)$)

__Preuve__



nombre de noeuds rouge sur p

$$h(x) = h_n(x) + \underbrace{nr(p)}_{\leqslant h_n(x)+1}$$

$$h(x) \leqslant 2h_n(x) + 1 \iff h_n(x) \geqslant \frac{h(x)-1}{2}. \quad \square$$

__Prop.__ Dans un ABR rouge-noir à n noeuds et de racine r, $n \geqslant 2^{h_n(r)}$.

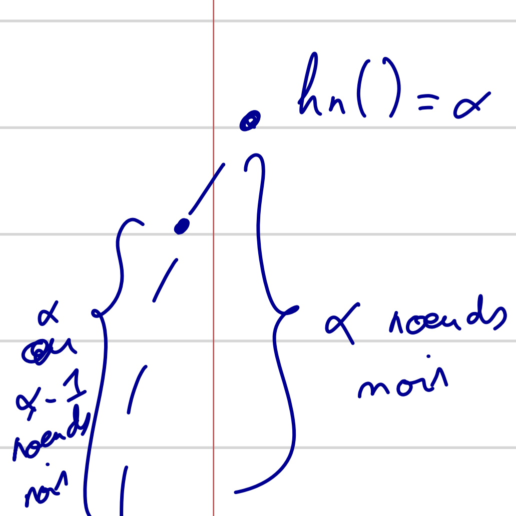
__Preuve__. Récurrence sur la hauteur h de l'arbre rouge-noir.

__Cas de base :__ $h = 0$ $n = 1$ $1 \geqslant 2^0$
$h_n(r) = 0$

__Cas inductif__

On suppose $n \geqslant 2^{h_n(r)}$ vérifié pour tous les arbres rouge-noir de hauteur $\leqslant h$. On considère un arbre de hauteur $h+1$.

- Si $h_n(r) = 0$ alors $n \geqslant 2^0 = 1$
- On suppose $h_n(r) > 0$. Cela signifie que la racine a deux enfants (à cause de la 2ème propriété des arbres rouge-noir).



Ces enfants sont racine d'arbres de hauteur $\leqslant h$ et de hauteur noire $\geqslant h_n(r) - 1$

$$n \geqslant 1 + 2^{h_n(r)-1} \times 2 = 1 + 2^{h_n(r)} \geqslant 2^{h_n(r)} \quad \square$$

Thm Les arbres rouge-noir sont des ABRE.

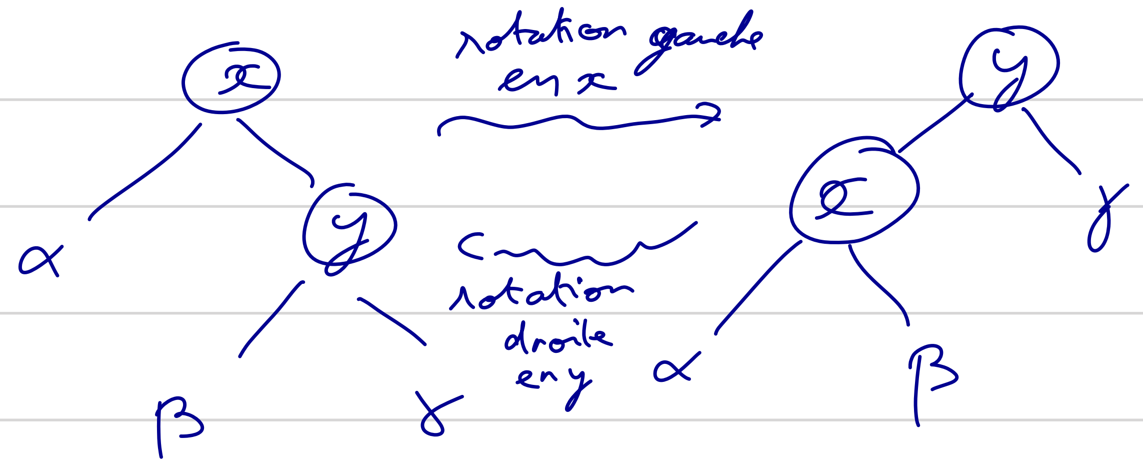
Preuve : $n \underset{\substack{2^{\text{ème}}\text{ prop.}\\ \downarrow}}{\geq} 2^{hn(x)} \underset{\substack{1^{\text{ère}}\text{ prop.}\\ \downarrow}}{\geq} 2^{\frac{h-1}{2}}$

$$\iff \log_2 n \geq \frac{h-1}{2} \iff h \leq 2(\log_2 n) + 1$$

Autrement dit $h = O(\log n)$. $\square$

Rotation { gauche, droite

Maintienment des ABR mais pas forcément les propriétés rouge-noir



Insertion dans un arbre rouge-noir

→ Insertion dans l'arbre à son endroit naturel (cf. recherche dans les ABR) z, colorié en rouge.

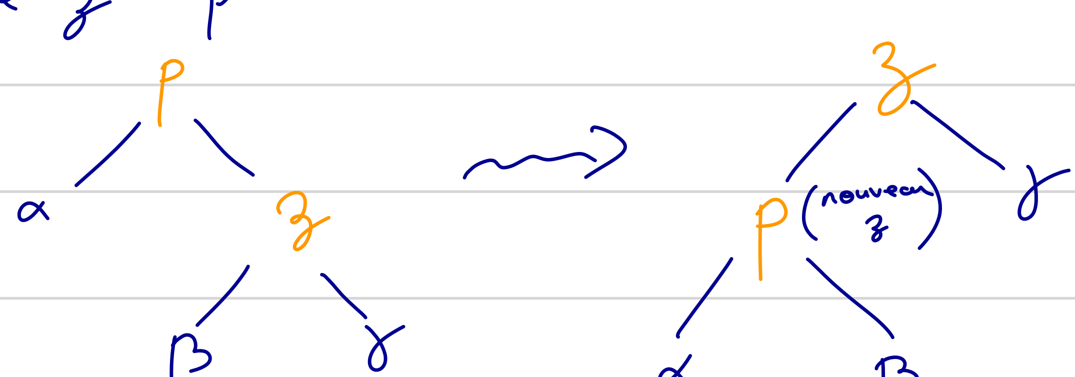
→ Si le père p de z est rouge, il faut corriger la violation des propriétés rouge-noir

1/ Si p est la racine, on colorie p en noir.

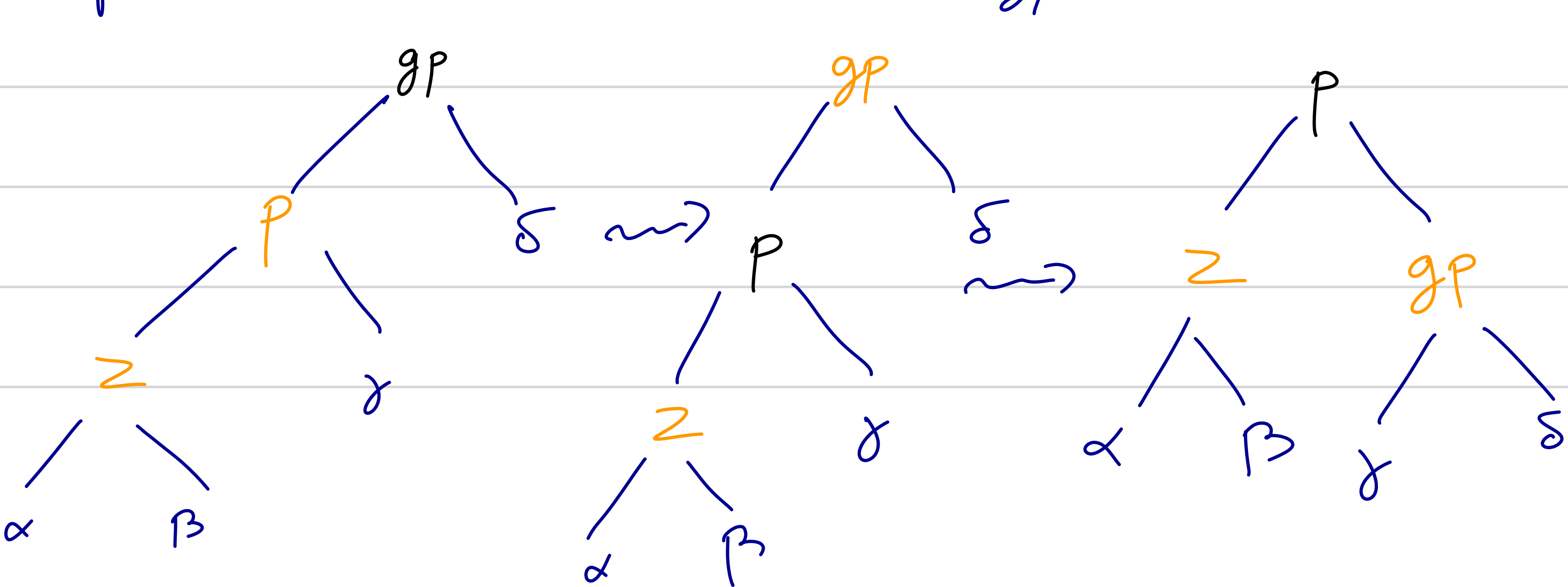
2/ On traite le cas où p est un enfant gauche d'un noeud gp (le cas où p est enfant droit de gp est symétrique).

3/ Si l'oncle w de z (s'il existe) est rouge, on colorie w et p en noir et on colorie gp en rouge. On continue récursivement en posant $z := gp$.

4/ Sinon, si z est un enfant droit, on fait une rotation gauche sur p et on considère le nouvel enfant gauche de z pour le cas suivant.



5/ z est un enfant gauche. On colorie son parent p en noir, son grand-parent gp en rouge, puis on fait une rotation droite sur gp. Fini.



Insérer dans l'arbre vide : 41, 38, 31, 12, 19, 8