TD h-équilibré : les AVL [1]

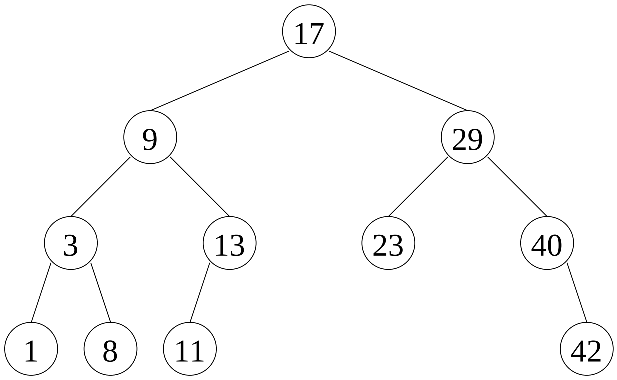

Figure 1 – AVL ?

1 Préliminaires

Exercice 1.1 (Déséquilibre (Balance factor))

1. (a) Qu'est-ce que le *déséquilibre* ?
   Quel est son domaine de définition ?
   (b) Indiquer sur l'arbre de la figure 1 les déséquilibres de tous les nœuds.
2. L'arbre de la figure 1 est-il un AVL ? Pourquoi ?
3. Écrire une fonction qui vérifie si un arbre binaire est *h-équilibré*.

Exercice 1.2 (ABR → AVL)

Écrire une fonction qui construit à partir d'un ABR classique (un arbre binaire sans les déséquilibres) un arbre équivalent au premier (contenant les mêmes valeurs aux mêmes places) mais avec le déséquilibre (un "champ" *bal*) renseigné en chaque nœud.

---

2 Rotations

Exercice 2.1 (Rotations et déséquilibres)

Soit un AVL de hauteur ≥ 2, ayant la structure représentée en figure 2.

1. Le but ici est de déterminer quelle rotation doit être utilisée pour chaque cas de déséquilibre, ainsi que les nouveaux déséquilibres après chacune des rotations.

1. Adelson-Velskii & Landis

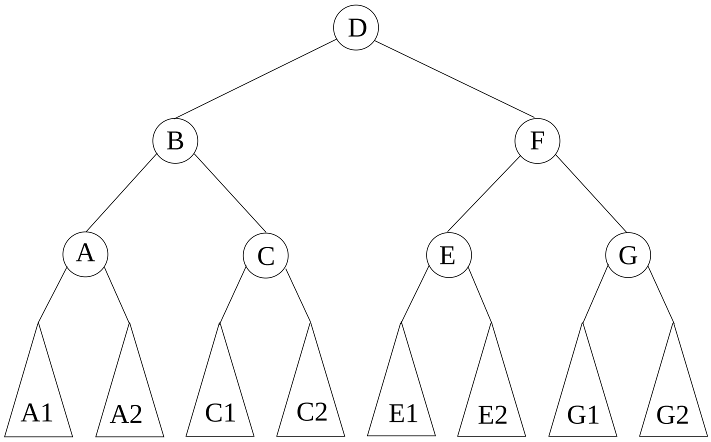

FIGURE 2 – AVL : *les cercles sont des nœuds et les triangles des sous-arbres*

(a) **Rotation gauche :**

i. Dessiner l'arbre obtenu après rotation gauche appliquée sur D.

ii. Quels sont les nœuds dont la valeur de déséquilibre a changé ?

iii. Donner, pour chaque cas de déséquilibre de la racine, les valeurs des déséquilibres avant et après rotation gauche des nœuds concernés. Vous donnerez votre réponse sous forme d'un tableau (voir annexe).

iv. En déduire les cas pour lesquels la rotation gauche permet de rééquilibrer l'arbre.

(b) Étudier de la même manière la **rotation droite**.

(c) En considérant uniquement les cas "intéressants", étudier de la même manière la **rotation gauche-droite**.

(d) En considérant uniquement les cas "intéressants", étudier de la même manière la **rotation droite-gauche** (voir annexe).

Remplir les tables données en annexe (dernière page). La dernière table doit résumer les cas où les rotations sont utiles au ré-équilibrage.

2. Afin de simplifier les algorithmes d'ajout et de suppression, il est intéressant d'intégrer les mises à jour des déséquilibres aux algorithmes des rotations.

(a) Pourquoi les rotations doubles ne peuvent plus être implémentées par les rotations simples ?

(b) Sachant que les rotations ne seront utilisées que dans les cas considérés ici (donner pour chaque rotations les spécifications précises d'utilisation), écrire les quatre fonctions de rotations avec mises à jour des déséquilibres.

Exercice 2.2 (Rotations et hauteur)

Lors d'une rotation, l'arbre concerné peut changer de hauteur. Il est donc nécessaire de savoir dans quel cas, celle-ci change, afin d'indiquer aux nœuds ancêtres que l'un des sous-arbres a changé de hauteur (ce qui modifie les déséquilibres des nœuds ancêtres).

Pour chaque rotation, reprendre les arbres obtenus à l'exercice précédent et répondre aux questions suivantes :

1. Dans quels cas l'arbre a-t'il changé de hauteur ?
2. En fonction de quelles valeurs peut-on déterminer les changement de hauteurs ?
3. **Compléter la table 3 donnée en annexe** en indiquant les variations de hauteur pour chaque cas.

3 Modifications

Exercice 3.1 (Insertion)

On va écrire ici une version récursive de l'ajout d'un élément dans un AVL. Celui-ci sera basé sur le principe de l'ajout en feuilles des arbres binaires de recherche, le rééquilibrage se faisant à la remontée.

1. Après insertion d'un élément, certains sous-arbres peuvent changer de hauteur. Afin de pouvoir mettre à jour les déséquilibres, il faut remonter cette information sur le chemin séparant la nouvelle feuille de la racine.
   (a) En considérant une insertion dans le sous arbre gauche, qui a donc augmenté la hauteur de ce sous-arbre, étudier les différents cas selon que l'arbre de départ avait un déséquilibre de -1, 0 ou 1 (voir figures 3 à 5). Certains cas devront être détaillés (faire des "sous-cas").

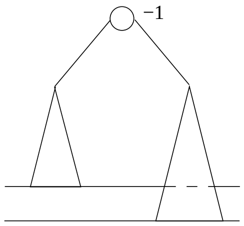

FIGURE 3 – Déséquilibre : -1

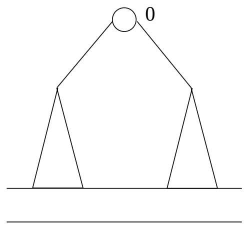

FIGURE 4 – Déséquilibre : 0

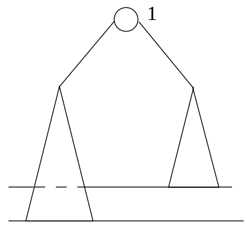

FIGURE 5 – Déséquilibre : 1

   (b) En déduire, en fonction du déséquilibre avant insertion ($deseq_i$), après insertion ($deseq_f$), après rotation ($deseq_r$), les modifications de hauteur de l'arbre.

2. En déduire le principe d'insertion dans les AVL, qui distingue bien la partie "insertion" de la partie "rééquilibrage".

3. Insérer les clés 4, 48, 7, 5 et 6 dans l'arbre de la figure 1.

4. Écrire la fonction d'insertion.

Exercice 3.2 (Suppression)

La suppression se fera sur le même modèle que l'insertion :

- La suppression même se fera sur le modèle de celle vue en td pour un arbre binaire de recherche.
- Le rééquilibrage se fera à la remontée.

1. Étudier, de la même manière que pour l'insertion, les différences de hauteur induites après une suppression et une éventuelle ré-équilibrage dans le sous-arbre gauche (reprendre les schémas des figures 3 à 5).

2. En déduire le principe de suppression dans les AVL, qui distingue bien la partie "suppression" de la partie "réparation".

3. Supprimer les clés 23, 17, 11 et 1 dans l'arbre de la figure 1.

4. Écrire la fonction de suppression.

Les tableaux à remplir

Rotations et déséquilibres

rotation gauche (lr)			
deseq(D)	**deseq(F)**	**deseq'(D)**	**deseq'(F)**
-2	-1		
	0		
	+1		
+2	-1		
	0		
	+1		

TABLE 1 – Valeurs des nouveaux déséquilibres après rotation gauche

rotation droite-gauche (rlr)					
deseq(D)	**deseq(F)**	**deseq(E)**	**deseq'(D)**	**deseq'(F)**	**deseq'(E)**

TABLE 2 – Valeurs des nouveaux déséquilibres après rotation droite-gauche (cas "utiles").

Résumé : Rotations et changements de hauteur

deseq(racine)	***deseq(fils gauche)*** ***deseq(fils droit)***[1]	**rotation**	ΔH
		gauche	
		droite-gauche	
deseq(racine)	***deseq(fils gauche)*** ***deseq(fils droit)***[1]	**rotation**	ΔH
		gauche-droite	
		droite	

[1] *Rayer la mention inutile*

TABLE 3 – Variations de hauteur