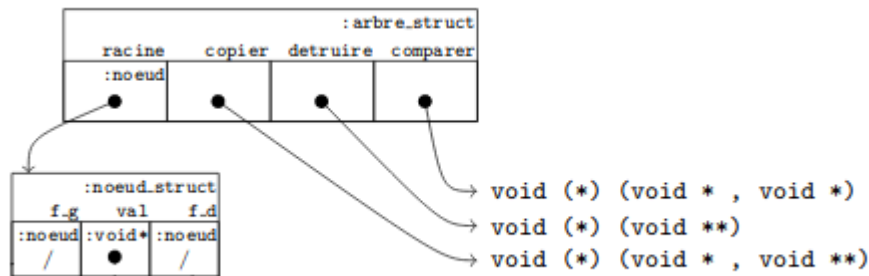


Programmation avancée et Structure de données

LES ARBRES BINAIRES DE RECHERCHES

Binôme : FOURNIER Bastien, ABDOULAYE Hamat

Représentation d'un arbre vide :



Les fonctions d'affichages prennent en argument un pointeur vers une fonction qui permet d'avoir accès à la fonction « afficher » spécialement conçu pour afficher la structure utilisé, ici des Integer et plus tard des Sigles, ce qui permet au programme d'être générique.

La fonction « arbre_rechercher_position » prend en paramètre un arbre et une valeur void*, la fonction est donc générique. La fonction va parcourir l'arbre jusqu'à trouver ou pas la valeur void*, et renvoyer le nœud où elle se trouve.

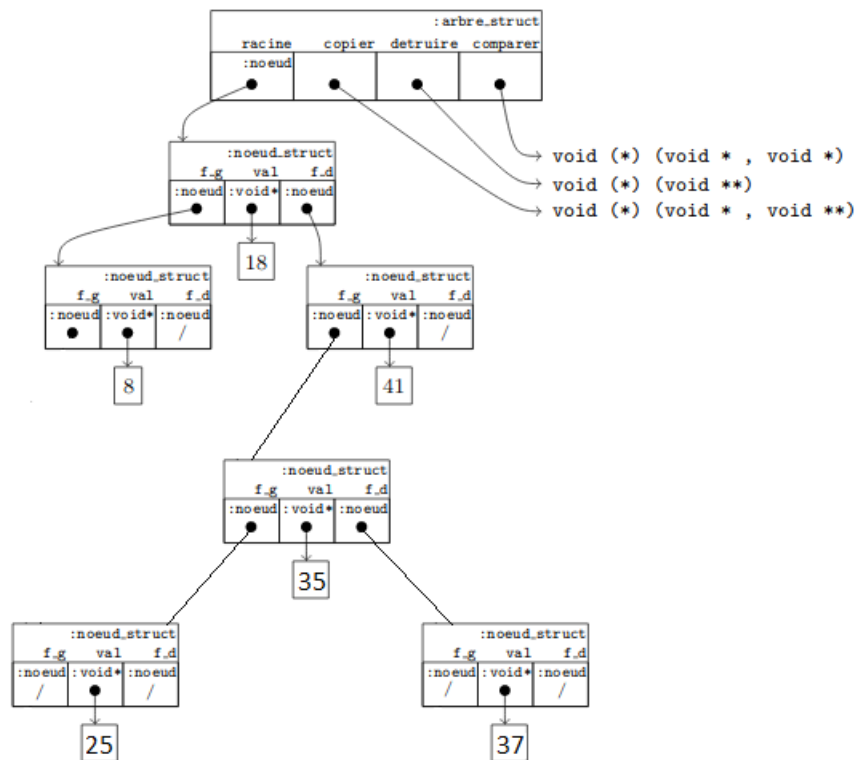
Complexité : La complexité de l'algorithme de suppression est $O(2^n)$ (n étant le nombre de nœud). Lors de la suppression on utilise la fonction « arbre_rechercher_position » qui parcourt l'arbre de la racine au nœud rechercher, qui a une complexité au pire de cas en $O(2^n)$. Ensuite on exécute la fonction « noeud_supprimer_simple » qui à pour complexité $O(2^n)$, jusqu'à arriver sur une feuille. Cela nous oblige donc à parcourir l'arbre, et donc dans le pire cas l'opération aura un temps proportionnel à 2^n .

Algorithme de suppression :

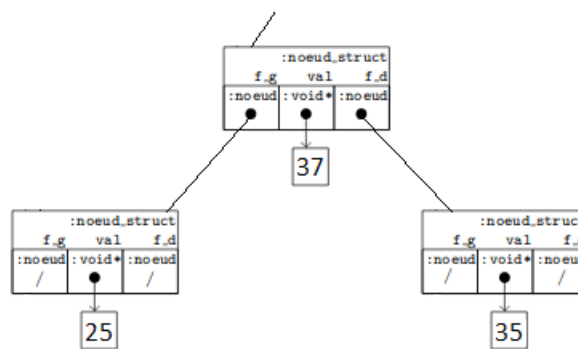
La fonction « arbre_supprimer » prend en argument un arbre et une valeur, elle fait appel à la fonction « arbre_chercher_position » pour trouver l'adresse du nœud correspondant à la valeur. Ensuite, elle fait appel à la fonction « noeud_detruire_simple » qui prend un pointeur de nœud en argument et un pointeur de la fonction « detruire », son but est de supprimer le nœud en argument tout en laissant un état cohérent à l'arbre après la suppression.

Exemple :

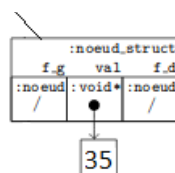
Supprimons la valeur 35 :



Une fois le nœud trouvé par la fonction « `arbre_chercher_position` », on appelle la fonction « `noeud_detruire_simple` » avec le nœud en argument. Ici, le nœud possède deux fils donc priorité au fils droit, un échange de valeur est fait entre les deux, puis on appelle de nouveau la fonction « `noeud_detruire_simple` » avec pour argument le fils droit dont l'on vient de modifier la valeur.



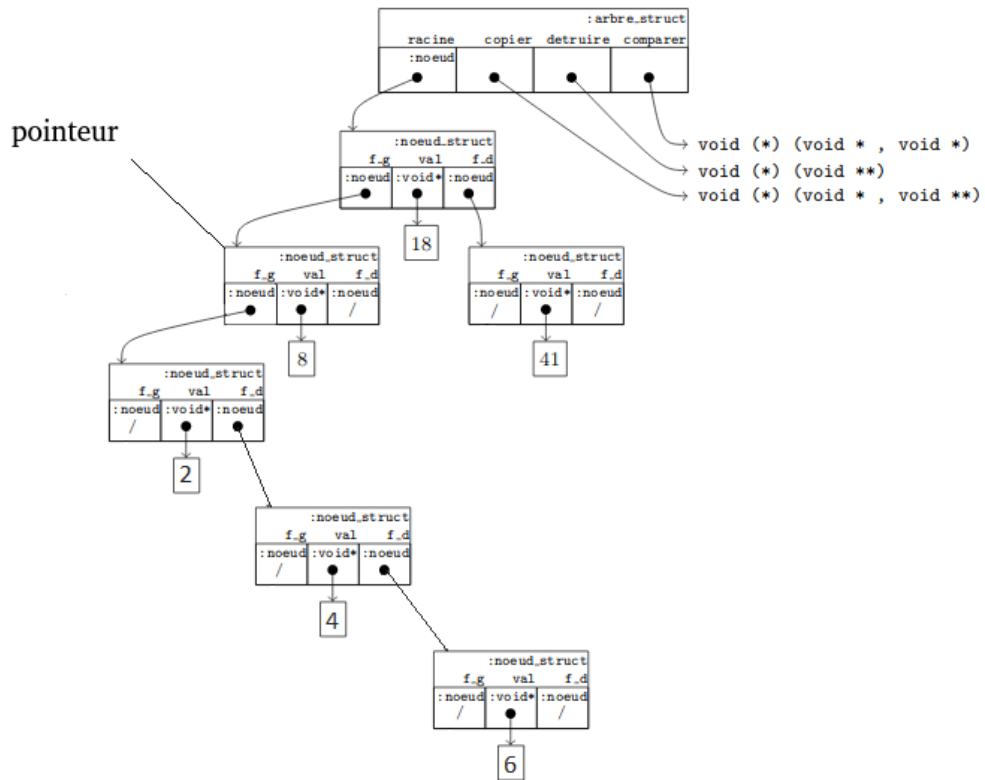
Étape suivante :



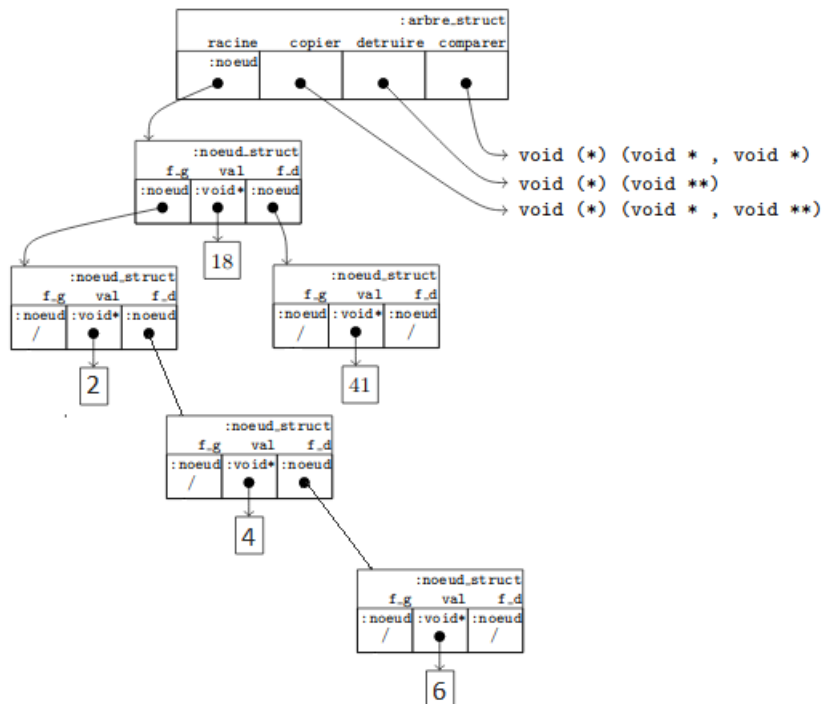
Ici, le nœud n'a pas de fils, il sera donc supprimer en faisant appel à la fonction « `detruire` » pour détruire la valeur pointée et on détruit le nœud.

Autre exemple :

Supprimons la valeur 8 :



Le nœud comportant la valeur 8 ne possède qu'un fils gauche, il suffit alors de le remplacer par celui-ci. On crée un pointeur sur le nœud comportant 8 et on échange 8 avec son fils gauche. On supprime ensuite le nœud avec l'aide du pointeur.



Bilan :

Nous avons terminé le TD dans son ensemble, sans erreur dans valgrind pour les deux exercices. Nous n'avons pas procédé exactement comme dans le sujet du TP pour la suppression d'un nœud possédant un fils droit, nous avons essayé, mais il nous paraissait plus simple de passer par notre méthode, au final le résultat reste le même mais peut-être pas le temps d'exécution.