

Dans ce TD, on étudie des arbres binaires, qui ne sont pas des arbres binaires de recherche.

## 1 L'algorithme des nœuds chapeaux

On modélise un calendrier par un arbre binaire, dans lequel une feuille représente une période élémentaire, et un nœud interne représente la période couverte par tous ses descendants. On définit les nœuds chapeaux d'une période, comme l'ensemble minimal de nœuds représentant cette période [1]. Un exemple est donné Figure 1.

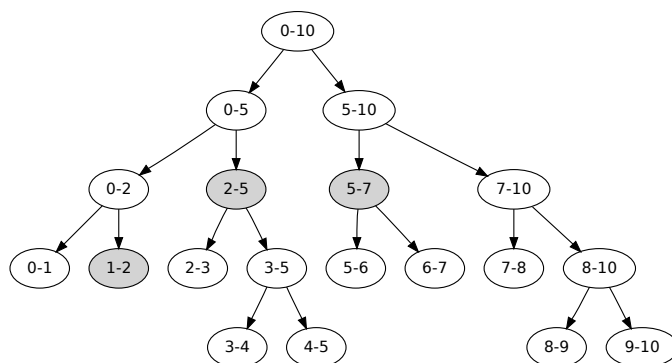


FIGURE 1 – Arbre binaire représentant un calendrier, pour une période 0 – 10. Les nœuds chapeaux de la période 1 – 7 apparaissent en gris.

**Question 1.** Donner la déclaration pour un type `struct` `nœud`, permettant de représenter un calendrier. Spécifier ce type.

**Question 2.** Écrire une fonction `nouveau_calendrier`, paramétrée par deux entiers  $t_0$  et  $t_1$ , qui retourne un calendrier couvrant la période  $[t_0, t_1[$ . Les feuilles doivent toutes être de durée 1.

### 1.1 Calcul des nœuds chapeaux d'une période

Prenons l'exemple de la Figure 1. La période est  $[t_0, t_1[ = [1, 7[$ . On commence par déterminer le plus petit nœud contenant la période. Sur l'exemple, il s'agit du nœud  $N_0 = [0, 10[$ . On détermine ensuite le nœud-chapeau  $N_G$  le plus à gauche. Il s'obtient en parcourant l'arbre jusqu'à trouver un nœud  $N$  tel que  $t_0(N) = t_0$  et  $t_1(N) \leq t_1$ . Le nœud-chapeau  $N_D$  le plus à droite s'obtient d'une manière similaire. Les autres nœuds chapeaux de la période se déduisent de  $N_G$  et  $N_D$ , en remontant l'arbre à partir de ces deux nœuds, jusqu'à  $N_0$  :

- pendant la remontée depuis  $N_G$ , si un nœud  $N \neq N_0$  est tel que  $t_1(N)$  est différent du champ  $t_1$  du dernier nœud-chapeau produit, alors le fils droit de  $N$  est lui-même un nœud-chapeau ;
- pendant la remontée depuis  $N_D$ , si un nœud  $N \neq N_0$  est tel que  $t_0(N)$  est différent du champ  $t_0$  du dernier nœud-chapeau produit, alors le fils gauche de  $N$  est lui-même un nœud-chapeau.

Sur l'exemple, en remontant depuis le nœud-chapeau  $N_G = [1, 2[$ , on considère  $N = [0, 2[$ , puis  $N = [0, 5[$ . Pour le premier nœud, on a  $t_1(N) = t_1(N_G) = 2$  : on ne produit pas de nouveau nœud-chapeau. Pour le second nœud,  $t_1(N) = 5$  est différent de  $t_1(N_G) = 2$ . On en déduit que  $[2, 5[$  est un nœud-chapeau. En remontant depuis le nœud-chapeau  $N_D = [5, 7[$ , on ne considère que le nœud  $N = [5, 10[$  mais, comme  $t_0(N) = t_0(N_D) = 5$ , on ne produit aucun nouveau nœud-chapeau.

**Question 3.** Donner une déclaration C pour une structure permettant de représenter l'ensemble des nœuds chapeaux d'une période. Spécifier cette structure.

**Question 4.** Écrire en C une fonction `noeuds_chapeaux`, paramétrée par un calendrier, une période, et qui calcule l'ensemble des nœuds chapeaux de la période.

**Question 5.** Écrire en C une fonction `dot_calendrier`, paramétrée par un calendrier, l'ensemble des nœuds chapeaux d'une période, et qui affiche le calendrier en grisant les nœuds chapeaux, à la façon de la Figure 1. Voici un extrait du fichier à produire :

```
digraph G {
    "0-10" -> "0-5";
    "0-5" -> "0-2";
    ...
    "8-10" -> "8-9";
    "8-10" -> "9-10";
    "1-2"[style=filled];
    "2-5"[style=filled];
    "5-7"[style=filled];
};
```

## Références

- [1] Wikipedia. Algorithme des nœuds chapeaux. [fr.wikipedia.org/wiki/Algorithme\\_des\\_noeuds\\_chapeaux](http://fr.wikipedia.org/wiki/Algorithme_des_noeuds_chapeaux).