Cours arbres

Contents

I) Arbres binaires

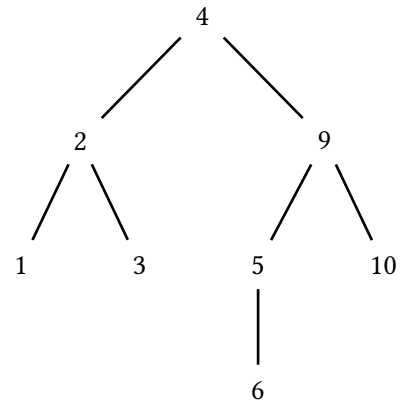
I.1) Définitions

Définition 1 [Arbre binaire]

> Un arbre binaire stockant des données d'un ensemble E est défini par induction comme :
> - Soit l'arbre vide
> - Soit une feuille contenant un élément de E.
> - Soit un noeud interne contenant un ou deux arbres et un élément de E.
>
> On appelle *noeud* soit un noeud interne, soit une feuille.
>
> Les éléments de E sont appelés *étiquette* d'un noeud, et les deux arbres associé à un noeud interne sont communément appelés *fils droit* et *fils gauche*.

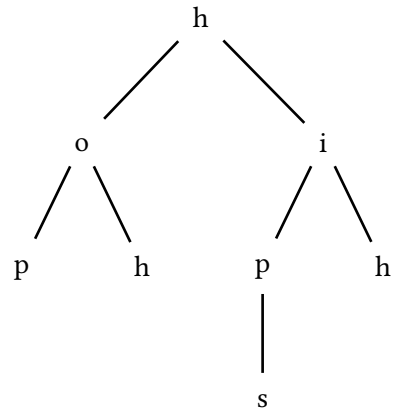
Exemple 2

Arbre binaire de recherche stockant les nombres 1, 2, 3, 4, 5, 6, 9, 10.



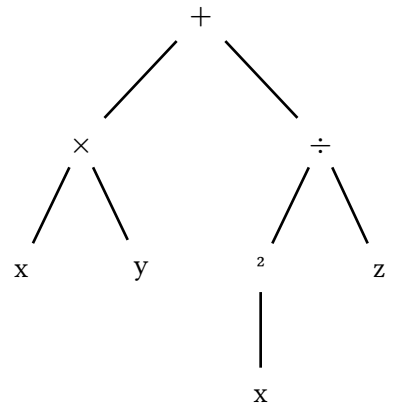
Exemple 3

Arbre préfixe représentant les onomatopées “hop” “hoh” “hip” “hips” et “hih”.



Exemple 4

Arbre d’opération de $x \times y + \frac{x^2}{z}$.



Pour ces trois exemples, la structure de l’arbre est identique, mais les données stockées sont de nature différente

Définition 5 [Arbre binaire strict]

> Un arbre binaire *strict* est un arbre binaire où tous les noeuds internes ont exactement deux fils.
> Attention aux définitions, parfois on appelle “arbre binaire” des arbres binaires strictes.

I.2) Propriétés des arbres

Définition 6 [racine]

> La *racine* d’un arbre est son noeud d’origine (le plus en haut dans la représentation).

Exemple 7

On va donner pour exemple l’arbre de l’ex 2.

4 est sa racine

Définition 8 [profondeur]

> La *profondeur* d’un noeud est sa distance à la racine.

Exemple 9

- 4 est à une profondeur de 0.
- 5 est à une profondeur de 2.

Définition 10 [taille]

> La *taille* d'un arbre est son nombre de noeuds.

Exemple 11

Ici, 8.

Définition 12 [hauteur]

> La hauteur d'un arbre est la profondeur maximum des noeuds de l'arbre.
>
> Par définition, l'arbre vide a une hauteur de -1.

Exemple 13

Ici, 3, car 6 est à une profondeur 3.

Définition 14 [Père, fils, descendant, sous-arbre enraciné]

> - Un noeud est le fils d'un autre si c'est son fils droit ou son fils gauche. Ce dernier est alors le père de son fils.
> - Les descendants d'un noeuds sont ses fils, et les descendants de ses fils.
> - Le sous arbre enraciné en un noeud est composé de ce noeud et de tous ses descendants.

I.3) Et en OCaml ?

On peut définir des arbres binaires de la manière suivante :

```
type arbre_binaire =
    Vide
    |N of arbre_binaire * int * arbre_binaire
```

Un noeud avec un seul fils peut être représenté par un fils vide. Une feuille peut être représenté par deux fils vides.

Les arbres binaires stricts non vides sont le plus souvent représentés par :

```
type arbre_binaire_strict =
    |F of int
    |N of arbre_binaire_strict * int * arbre_binaire_strict
```

C'est la forme la plus courante d'arbre binaires.

Les fonctions sur les propriétés les plus courantes peuvent être définies de la manière suivante :

```
let rec taille a = match a with
    | F _ -> 1
    | N (fg, _, fd) -> (taille fg) + (taille fd)
;;
let rec hauteur a = match a with
    | F _ -> 0
    | N (fg, _, fd) -> max (taille fg) (taille fd)
;;
```

I.4) Preuves sur les arbres binaires

Soit A un arbre binaire. On note $T(A)$ sa taille et $h(A)$ sa hauteur.

Montrer que $T(A) \leq 2^{h(A)+1} - 1$

On va faire une preuve par *induction structurelle* :

- (cas de base) si F est une feuille $T(F) = 1 \leq 2^{h(F)+1} - 1 = 1$
- (induction) soit $A = \big(G, , D\big)$ un arbre. Par induction, on a : $T(G) \leq 2^{h(G)+1} - 1$ et $T(D) \leq 2^{h(D)+1} - 1$

Donc

$$\begin{aligned} T(A) &= T(G) + T(D) + 1 \\ &\leq 2^{h(G)+1} - 1 + 2^{h(D)+1} - 1 + 1 \\ &\leq 2^{\max(h(G),h(D))+2} - 1 \\ &\leq 2^{h(A)-1} - 1 \end{aligned}$$

Cela conclu la preuve.