

Arbres de décision binaires

Modélisation et programmation par objets - HAI401I

Les arbres et les arborescences sont utiles pour :

- structurer des informations selon un principe de division,
- effectuer certains traitements de manière efficace (par exemple une recherche d'information).

Pour re-situer ces notions d'un point de vue théorique, ce sont des graphes particuliers, pour rappel :

- Un *graphe orienté* $G = (V, E)$ est un couple composé d'un ensemble de nœuds ou sommets (V) et d'un ensemble d'arcs ($E \subseteq V \times V$). Cet ensemble d'arcs représente une relation entre les éléments de V . Un arc est un couple (s, t) .
- Dans un *graphe non orienté*, V contient des paires $\{s, t\}$ et non des couples et on parle d'arêtes plutôt que d'arcs.
- Un *arbre* est un graphe non orienté, connexe (on peut aller d'un nœud à un autre par une chaîne d'arêtes dans V) et dans lequel on ne trouve pas de cycle (aucune chaîne ne part d'un sommet pour y revenir).
- Un *arbre enraciné* est un arbre dans lequel on distingue l'un des nœuds comme étant sa racine.
- Une *arborescence* est un arbre orienté (tout nœud est accessible depuis la racine par un chemin d'arcs orientés).
- les *feuilles* d'un arbre ou d'une arborescence sont les nœuds qui n'ont qu'un voisin.

Nous allons étudier des arborescences particulières, que sont les *arbres binaires de décision*. Ils vérifient les propriétés suivantes :

- ils sont enracinés,
- il servent à représenter de manière graphique une procédure de classification d'un ensemble d'objets,
- ils modélisent une hiérarchie de tests sur des attributs des objets classés (ces tests ont pour réponse "vrai" ou "faux"),
- les nœuds internes sont étiquetés par les tests,
- les arcs issus d'un nœud correspondent aux réponses possibles,
- les feuilles représentent les décisions (on peut dire aussi que les feuilles représentent les classes dans lesquelles se rangent les objets),
- un nœud interne n'a que deux successeurs, correspondant aux réponses "vrai" ou "faux".

Les arbres binaires de décision peuvent être appris à partir d'exemples, par différentes procédures plus ou moins compliquées¹. Nous donnons ci-dessous un exemple simple avec des animaux et leurs attributs (table 1), à partir duquel on pourrait construire l'arbre de décision de la figure 1.

1. <https://www.cs.waikato.ac.nz/ml/weka/>

	vertébré (squel. interne)	plumes	poils	coquille	<i>classe</i>
autruche	x	x			oiseau
canard	x	x			oiseau
lézard	x				poisson/reptile
musaraigne	x		x		mammifère
coccinelle					arthropode
saumon	x				poisson/reptile
escargot				x	mollusque
crevette					arthropode

TABLE 1 – Une table décrivant des animaux par des attributs

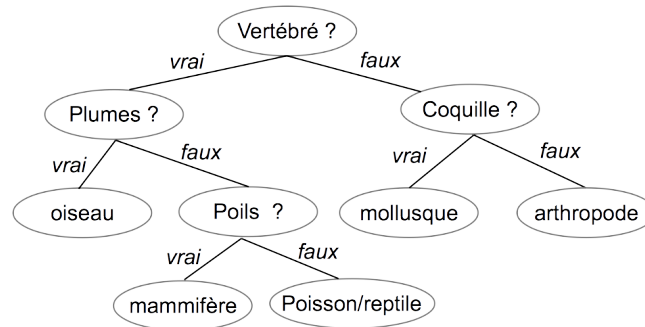


FIGURE 1 – Un arbre de décision pour classer les animaux de la figure 1

Les premiers éléments d'un type abstrait Arbre de Décision Binaire (ADB) sont donnés ci-dessous.

Types

- ADB
- E

Opérations

- composer(E étiquette, ADB vrai, ADB faux) : arbre
- sousArbreVrai() : ADB
- sousArbreFaux() : ADB
- estVide() : boolean
- estFeuille() : boolean
- getEtiquette() : E

Axiomes

- $\text{composer}(e, v, f).\text{getEtiquette}() = e$
- $\text{composer}(e, v, f).\text{sousArbreVrai}() = v$
- $\text{composer}(e, v, f).\text{sousArbreFaux}() = f$
- $\text{composer}(a.\text{getEtiquette}(), a.\text{sousArbreVrai}(), a.\text{sousArbreFaux}()) = a$ (quand $a.\text{estVide}() = \text{false}$)
- $a.\text{estVide}()$ ssi $a.\text{getEtiquette}()$ n'a pas de valeur
- $a.\text{estFeuille}()$ ssi $a.\text{sousArbreVrai}() = \text{null}$ and $a.\text{sousArbreFaux}() = \text{null}$ (quand $a.\text{estVide}() = \text{false}$)
- $a.\text{sousArbreVrai}() = \text{null}$ ssi $a.\text{sousArbreFaux}() = \text{null}$

Parcourir un arbre, comme beaucoup d'autres opérations, se fera le plus souvent de manière récursive. Parcourir consiste à énumérer les étiquettes dans un certain ordre :

- un parcours en largeur énumère par niveaux
Vertebre ?-Plumes ?-Coquille ?-oiseau-Poils ?-mollusque-arthropode-mammifere-poisson/reptile-
- un parcours en profondeur énumère une branche avant de passer à la suivante suivant un ordre :
 - préfixe (étiquette, gauche, droite)
Vertebre ?-Plumes ?-oiseau-Poils ?-mammifere-poisson/reptile-Coquille ?-mollusque-arthropode-
 - suffixe (gauche, droite, étiquette)
oiseau-mammifere-poisson/reptile-Poils ?-Plumes ?-mollusque-arthropode-Coquille ?-Vertebre ?-
 - infixé (gauche, étiquette, droite)
oiseau-Plumes ?-mammifere-Poils ?-poisson/reptile-Vertebre ?-mollusque-Coquille ?-arthropode-

QUESTION 1 *Créer une première interface générique, paramétrée par le type des étiquettes des nœuds, pour représenter les arbres binaires de décision, munie des opérations prévues dans le type abstrait et d'une méthode de parcours préfixe (appelée `toString`). Vous la complèterez ensuite.*

QUESTION 2 *Créer une ou plusieurs classes implémentant cette interface (par exemple, arbre vide, arbre avec une étiquette, arbre réduit à une feuille, arbre avec des sous-arbres).*

QUESTION 3 *Créer dans un `main` un arbre de décision correspondant à la figure 1.*

QUESTION 4 *Créer dans un `main` un arbre de décision pour déterminer si un nombre correspond à une année bissextile. Une année est bissextile (elle aura 366 jours) : si elle est divisible par 4 et non divisible par 100, ou si elle est divisible par 400. 1900 n'est pas bissextile car divisible par 4 et par 100 et non divisible par 400. 2000 est bissextile car divisible par 400. 2008 est bissextile car divisible par 4 et non divisible par 100.*

Vous testerez toutes les méthodes suivantes sur un arbre vide, une feuille et les arbres créés aux questions précédentes.

QUESTION 5 *Rechercher si une étiquette apparaît dans un arbre binaire.*

QUESTION 6 *Compter le nombre de feuilles de l'arbre.*

QUESTION 7 *Compter le nombre de nœuds internes de l'arbre.*

QUESTION 8 *Ecrire une méthode qui recherche la classe d'un objet en parcourant l'arbre et en posant des questions à chaque nœud.*

QUESTION 9 *Ecrire une méthode inspirée par le parcours préfixe et qui affiche l'arbre avec un décalage et des indentations, par exemple :*

Vertebre ?

Plumes ?

oiseau

Poils ?

mammifere

poisson/reptile

Coquille ?

mollusque

arthropode

QUESTION 10 *Ecrire les méthodes permettant d'effectuer respectivement un parcours infixe, un parcours postfixe et un parcours en largeur.*