Introduction à l’informatique : Arbres

Table des matières

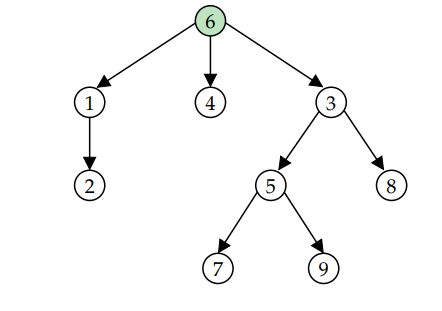
[Définitions basiques 3](#_Toc530222715)

[Arbres binaires 3](#_Toc530222716)

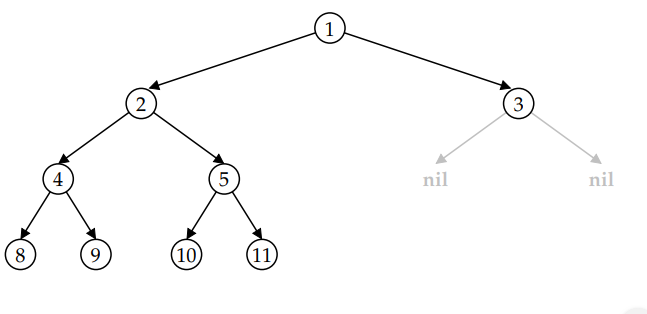
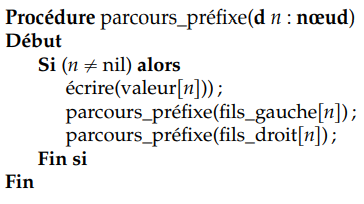
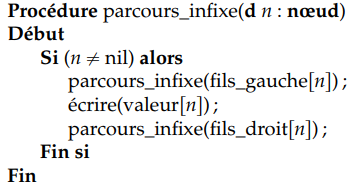
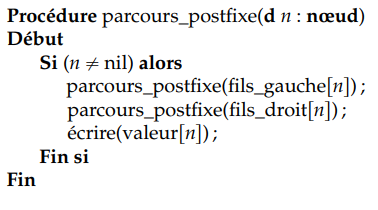
[Arbres binaires de recherche 5](#_Toc530222717)

[Autre application : arbres de décision 6](#_Toc530222718)

# Définitions basiques

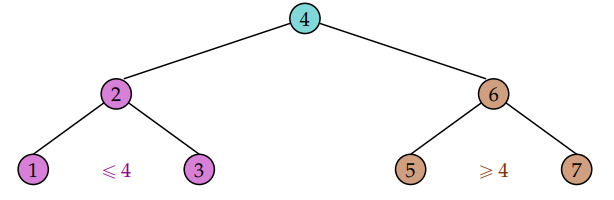
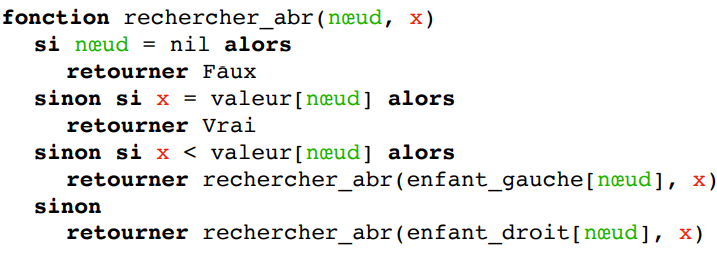
* Un arbre est un graphe non orienté, connexe et sans cycle.
* Un arbre enraciné est un graphe orienté, connexe et sans cycle tel que :  
  - un unique sommet est de degré entrant nul, la racine  
  - tous les autres sommets ont pour degré entrant 1  
  
* Feuille : ensemble des nœuds ayant un degré sortant nul.
* Hauteur : sa profondeur maximale
* Profondeur : un nœud est de profondeur p s’il est à une distance p-1 de la racine de l’arbre (nœud racine de profondeur 1)
* On peut représenter un arbre par la matrice d’adjacence.

# Arbres binaires

* Un arbre binaire est un arbre enraciné tel que le degré sortant de chaque nœud est au plus 2.
* Un arbre binaire est :  
  - soit l’arbre vide, noté nil  
  - soit un nœud ayant une valeug (e.g, un entier), un fils gauche et un fils droit, tel que les fils sont des (sous-)arbres binaires.  
  {4} ⬄{4}-{nil, nil}  
   
* Arbre complet : où toutes les feuilles ont la même profondeur.
* Types de parcours :   
  parcours préfixe  
    
    
   parcours infixe   
    
    
  parcours post fixe  
  

-Une expression arithmétique (totalement parenthèse) est :  
- un nb entier n  
- une somme/soustraction/produit/division parenthèse  
- parcours post fixe -> notation polonaise inversée :  
(7 8 + 4 \* 3 / 1 -) = ((((7+8)\*4)/3)-1)

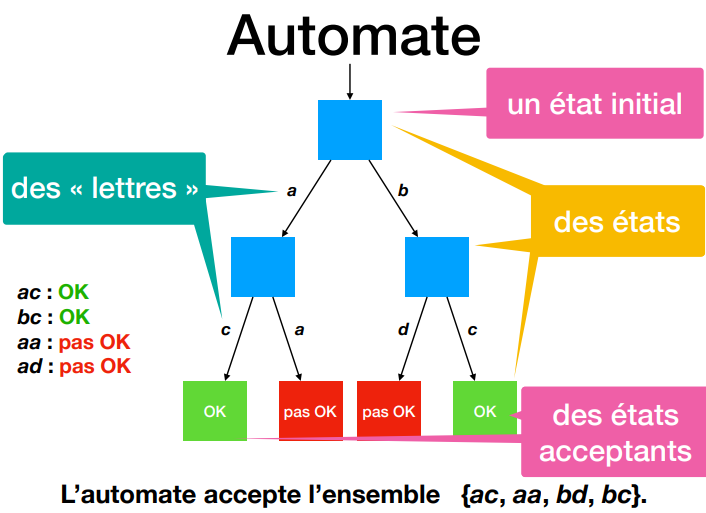
# Arbres binaires de recherche

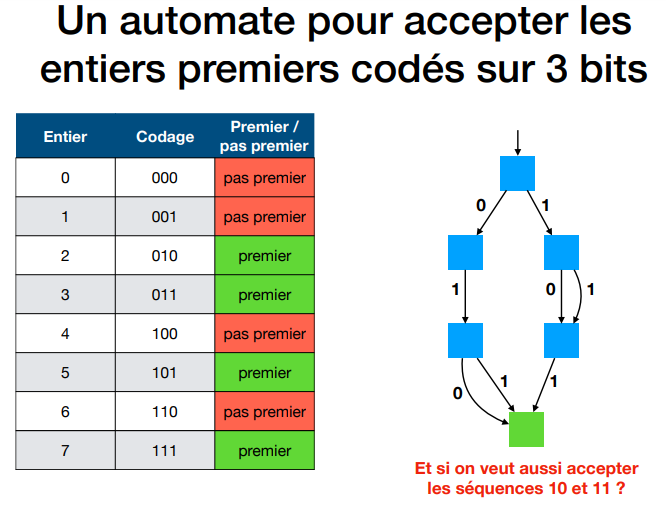
* Un arbre binaire de recherche (ABR) est un arbre binaire tel que, pour tout nœud n ayant pour valeur v\_n, l’ensemble des nœuds de son fils gauche (resp. droit) admettent des valeurs inférieures (resp. supérieurs) à v\_n selon une relation d’ordre totale donnée.   
  
* Recherche de type parcours préfixe.
* Soit ABR composé de n nœuds  
  - O(h) dans le cas général avec a SLIDE 25
* Insertion dans un ABR : parcours préfixe.
* Voir arbre AVL
* Recherche en ABR :  
  
* Complexité d’un ABR équilibré : log\_2(n) ; pire cas : O(n)

Autres utilisations d’arbres (ex : annuaire pour mettre à jour de manière plus efficace qu’avec un tableau) ;

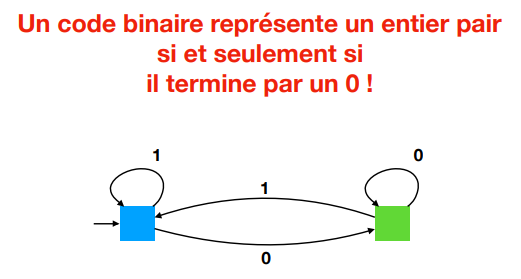
# Autre application : arbres de décision

(question ? oui | non) ; (question ? reponse A -> reponse | reponse B -> question ? question ? oui -> reponse | non -> reponse)… (ex : detection de spam avec arbre de décision -> reconnaissance de motifs) -> abstraction en automate (séquence -> automate -> sortie)  
Un automate décrit un ensemble de séquences valides, celles pour laquelle il décide true.

Nœud d’automate : états ; des lettres/symboles pour les decision, un état ititial, des états acceptant/ !acceptant (true/false)  




-> on peut avoir un état acceptant, mais continuer après.

Comment accepter un ensemble infini ? abre infini : problématique.  
-> utilisation de boucle a la place !  


Un code binaire représente un entier pair ssi il termine par un 0.

Un transducteur : automate qui répond oui/non et peut écrire.   
