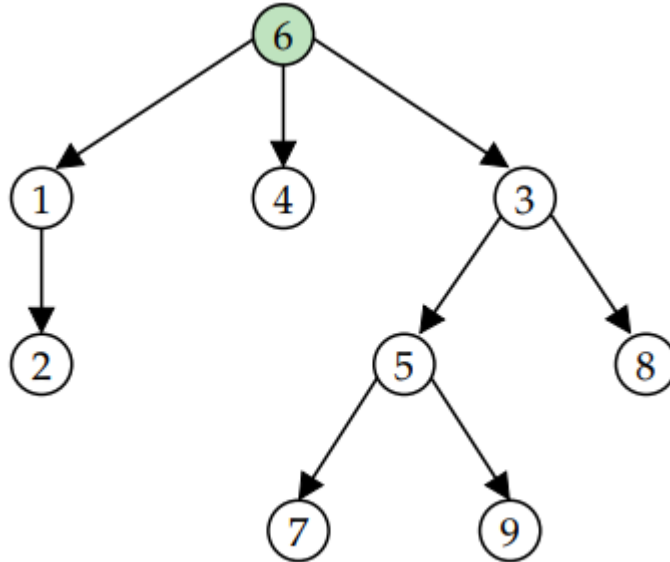

Introduction à l'informatique : Arbres

Table des matières

Définitions basiques.....	3
Arbres binaires.....	3
Arbres binaires de recherche	5
Autre application : arbres de décision	6

Définitions basiques

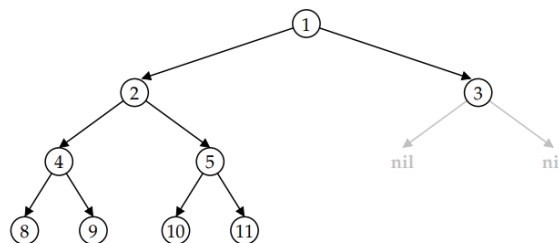
- Un arbre est un graphe non orienté, connexe et sans cycle.
- Un arbre enraciné est un graphe orienté, connexe et sans cycle tel que :
 - un unique sommet est de degré entrant nul, la racine
 - tous les autres sommets ont pour degré entrant 1



- 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 valeur (e.g, un entier), un fils gauche et un fils droit, tel que les fils sont des (sous-)arbres binaires.
- $\{4\} \Leftrightarrow \{4\} - \{\text{nil}, \text{nil}\}$



- Arbre complet : où toutes les feuilles ont la même profondeur.
- Types de parcours :
parcours préfixe

Procédure parcours_préfixe(**d** n : **nœud**)

Début

Si ($n \neq \text{nil}$) **alors**

 écrire(valeur[n]);

 parcours_préfixe(fils_gauche[n]);

 parcours_préfixe(fils_droit[n]);

Fin si

Fin

parcours infixe

Procédure parcours_infixe(**d** n : **nœud**)

Début

Si ($n \neq \text{nil}$) **alors**

 parcours_infixe(fils_gauche[n]);

 écrire(valeur[n]);

 parcours_infixe(fils_droit[n]);

Fin si

Fin

parcours post fixe

Procédure parcours_postfixe(**d** n : **nœud**)

Début

Si ($n \neq \text{nil}$) **alors**

 parcours_postfixe(fils_gauche[n]);

 parcours_postfixe(fils_droit[n]);

 écrire(valeur[n]);

Fin si

Fin

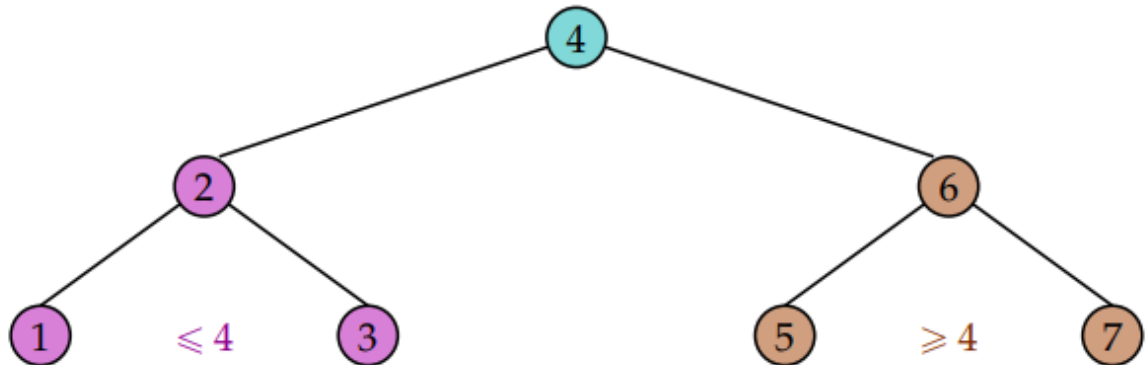
-Une expression arithmétique (totalement parenthèse) est :

- 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. supérieures) 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 :

```

fonction rechercher_abr(nœud, x)
  si nœud = nil alors
    retourner Faux
  sinon si x = valeur[nœud] alors
    retourner Vrai
  sinon si x < valeur[nœud] alors
    retourner rechercher_abr(enfant_gauche[nœud], x)
  sinon
    retourner rechercher_abr(enfant_droit[nœud], x)
  
```

- Complexité d'un ABR équilibré : $\log_2(n)$; pire cas : $O(n)$

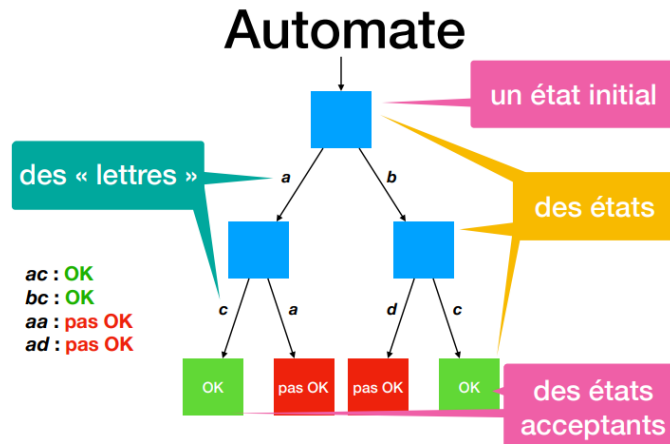
Autres utilisations d'arbres (ex : annuaire pour mettre à jour de manière plus efficace qu'avec un tableau) ;

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.

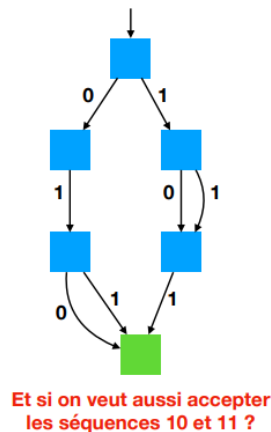
Nœud d'automate : états ; des lettres/symboles pour les decision, un état initial, des états acceptant/ !acceptant (true/false)



L'automate accepte l'ensemble {ac, aa, bd, bc}.

Un automate pour accepter les entiers premiers codés sur 3 bits

Entier	Codage	Premier / pas premier
0	000	pas premier
1	001	pas premier
2	010	premier
3	011	premier
4	100	pas premier
5	101	premier
6	110	pas premier
7	111	premier

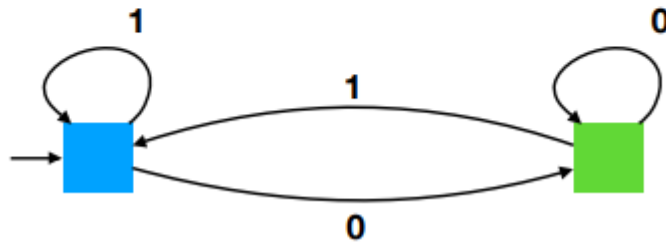


-> on peut avoir un état acceptant, mais continuer après.

Comment accepter un ensemble infini ? arbre infini : problématique.

-> utilisation de boucle a la place !

**Un code binaire représente un entier pair
si et seulement si
il termine par un 0 !**



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

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



- Café à 30 centimes €
- Thé à 50 centimes €
- La machine n'accepte que les pièces de 10 et 20 centimes €
- On peut insérer 50 centimes € maximum dans la machine
- On peut annuler à tout moment (et récupérer la monnaie)

