Mathématiques discrètes : Arbre

Licence — Université Lille 1
Pour toutes remarques : Alexandre.Sedoglavic@univ-lille1.fr

Semestre 3 — 2008-09

Un *arbre* est une forme particulière de graphe.

Nous aborderons dans la suite des familles d'arbres aux propriétés de plus en plus contraigantes.

Définition

- *Un arbre libre est un graphe non orienté, non vide, connexe et sans circuit.*
- *Un arbre (sous-entendu enraciné) est un arbre libre dont on distingue un sommet qualifié de racine.*
- *La donnée d'une racine oriente le graphe. Pour tout sommet* s *de ce dernier, il existe un unique chemin de la racine à* x*. Tout sommet distinct de* x *sur ce chemin est un ancêtre de* x *et on utilise la terminologie généalogique (parents, enfants, etc) intuitive pour désigner les relations entre les sommets.*
- *Un sommet sans enfant est une feuille.*
- *La hauteur d'un arbre est la longueur maximale d'un chemin reliant sa racine à une feuille.*
- *Une forêt est un ensemble d'arbre.*

Les arbres sont des graphes bien particuliers comme le montre la proposition suivante :

Proposition

Considérons un graphe non orienté $G := (S, A)$; les conditions suivantes sont équivalentes :

- *G est un arbre libre ;*
- *2 sommets de G quelconques de G sont connectés par un chemin simple unique ;*
- *G est connexe. Il ne l'est plus si on retire une arête quelconque ;*
- *G est sans circuit. Il ne l'est plus si on ajoute une arête quelconque ;*
- *$|A| = |S| - 1$.*

Représentation récursive

Un arbre sur un ensemble fini de sommets est un couple formé d'une racine et d'une partition des sommets restant en un ensemble d'arbre.
Par exemple, on peut avoir :

$$(1, \{(2, \{(5), (6)\}), (3, \{(7), (8), (9)\}), (4)\})$$

Arbre binaire

Definition

Un arbre *binaire* sur un ensemble fini de sommets est soit vide, soit l'union disjointe de sa racine est d'un arbre binaire appelé sous-arbre gauche et d'un arbre binaire appelé sous-arbre droit.

Un arbre binaire A peut être représenté par un triplet (A_g, r, A_d).

L'arbre $(1, \{(2, \{3\})\})$ peut correspondre à plusieurs arbres binaires dont :

$$(\emptyset, 1, (\emptyset, 2, 3)) \quad \text{et} \quad ((\emptyset, 2, 3), 1, \emptyset).$$

Ainsi, un arbre binaire n'est pas simplement un arbre ordonné d'arité 2.

Arbre binaire de recherche

Definition

Un *arbre binaire de recherche* — disons ABR pour faire court — est un arbre binaire dont chaque nœd n possède une clef telle que chaque nœd du sous-arbre gauche a une clef inférieure ou égale à celle de n et que chaque nœd du sous-arbre droit a une clef supérieur ou égale à celle de n.

L'intérêt de cette structure de donnée est que la complexité moyenne de la recherche d'un nœud est proportionnelle à la hauteur moyenne des nœuds de l'arbre.
On peut rééquilibrer un ABR par des *rotations* des sous arbres (cf. TD).

Arbre de syntaxe abstraite

Ce type d'arbre est basé sur la définition récursive suivante :

Definition

Une expression syntaxique est soit

- l'expression vide ;
- un élément d'un ensemble E ;
- un triplet (e_1, o, e_2) ou (e_1, e_2) est un couple d'expression syntaxique et o est un élément d'un ensemble O.

Cette définition peut se lire comme celle d'un arbre binaire que l'on appelle *arbre de syntaxe abstraite*.
Prenons comme exemple les expressions arithmétique :

$$(1, +, (2, \times, 3))$$

Trie

Ce terme de *trie* vient de l'anglais retrieval et on peut le francisé par l'expression *arbre prefix*.

Definition

Un *trie* est un arbre binaire dont chaque nœud est associé à une chaîne par le biais de l'arcs dont il est une feuille.

- La racine est associée à une chaîne vide ;
- À chaque descendant d'un nœud n est associé à une chaîne ayant un prefix commun avec la chaîne associé à n.

Contrairement à un arbre binaire de recherche, aucun nœud dans un trie ne stocke la chaîne à laquelle il est associé. C'est la position du nœud dans l'arbre qui détermine la chaîne correspondante.

Exemple : codage de Huffman

Ce codage a été proposé par David Huffman en 1952.
Considérons une source S (i.e. en ensemble d'éléments à coder) et sa distribution f de fréquence d'apparition.

Definition

La *longueur moyenne* d'un mot de code relativement à la distribution f de la source S est :

$$\sum_{x \in S} f(x)|C(x)|$$

avec $|C(x)|$ la longueur d'un mot codant $x \in S$.
Un codage C de $\mathcal{S}$ est *optimal* au regard de la distribution f s'il est décodable de manière unique et qu'il n'existe aucun autre code décodable de longueur moyenne plus petite que la sienne

Le codage de Huffman permet de construire un codage optimal en utilisant la proposition suivante.

Proposition

Soit une source $\mathcal{S} = \{s_1, \ldots, s_N\}$ munie de la distribution de fréquence f. On suppose les symboles de $\mathcal{S}$ ordonnés suivant f $(f(s_1) \geq \ldots \geq f(s_N))$.
Soit une deuxième souce $\mathcal{S}' = S \setminus \{s_{N-1}, s_N\} \cup \{s_{N-1,N}\}$ munie de la distribution de fréquence donnée par :

$$f'(s) = f(s) \text{ si } s \notin \{s_{N-1}, s_N\} \text{ et } f'(s_{N-1,N}) = f(s_{N-1}) + f(s_N)$$

Alors, si C' est un codage binaire optimal pour $\mathcal{S}'$, le code $\mathcal{C}$:

$$\begin{array}{rcl} C : \mathcal{S} & \rightarrow & \{0,1\}^\star \\ s & \rightarrow & \left\{ \begin{array}{rcl} C(s) & = & C'(s) \text{ si } s \notin \{s_{N-1}, s_N\} \\ C(s_{N-1}) & = & C'(s_{N-1}, s_N).0 \\ C(s_N) & = & C'(s_{N-1}, s_N).1 \end{array} \right. \end{array}$$

est un codage optimal pour la source $\mathcal{S}$.

Exemple

Soit la source

s	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
f(s)	0.4	0.15	0.15	0.1	0.1	0.06	0.02	0.02

La construction du codage de Huffman pour cette source est

symboles	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
fréquence	0.4	0.15	0.15	0.1	0.1	0.06	0.02	0.02
lettres							0	1
symboles	s_1	s_2	s_3	s_4	s_5	s_6	s_{78}	
fréquence	0.4	0.15	0.15	0.1	0.1	0.06	0.04	
lettres						0	1	
symboles	s_1	s_2	s_3	s_4	s_5	s_{678}		
fréquence	0.4	0.15	0.15	0.1	0.1	0.1		
lettres					0	1		

Exemple (suite)

symboles	s_1	s_{5678}	s_2	s_3	s_4
fréquence	0.4	0.2	0.15	0.15	0.1
lettres				0	1
symboles	s_1	s_{34}	s_{5678}	s_2	
fréquence	0.4	0.25	0.2	0.15	
lettres			0	1	
symboles	s_1	s_{25678}	s_{34}		
fréquence	0.4	0.35	0.25		
lettres		0	1		
symboles	$s_{2345678}$	s_1			
fréquence	0.6	0.4			
lettres	0	1			

Exemple (fin)

Pour finir, il suffit de reconstruire les mots du code

$s_{2345678}$ 0	s_1 1						
s_1 1	s_{25678} 00	s_{34} 01					
s_1 1	s_{34} 01	s_{5678} 000	s_2 0001				
s_1 1	s_{5678} 000	s_2 001	s_3 010	s_4 011			
s_1 1	s_2 001	s_3 010	s_4 011	s_5 0000	s_{678} 0001		
s_1 1	s_2 001	s_3 010	s_4 011	s_5 0000	s_6 00010	s_{78} 00011	
s_1 1	s_2 001	s_3 010	s_4 011	s_5 0000	s_6 00010	s_7 000110	s_8 000111