

Schéma Relationnel Distribué pour les Arbres

C. Gavoille A. Labourel

Laboratoire Bordelais de Recherche Informatique (LaBRI)
Université Bordeaux 1

Journées Graphes et Algorithmes, 8-9 novembre 2007

Représentation distribuée pour les distances bornées

Problème

Attribuer de l'information à chaque sommet des graphes d'une famille de graphes \mathcal{F} donnée de telle sorte que l'on puisse déterminer localement la distance entre deux sommets à distance au plus k .

L'information locale de chaque sommet est codée sous forme d'une étiquette binaire.

Buts

Minimiser la taille de l'information à stocker et la complexité des fonctions de distance et d'attribution d'information.

(k_1, k_2) -relation

Définition

Deux nœuds u et v sont (k_1, k_2) -reliés si u est à distance k_1 de w , plus petit ancêtre commun à u et v et si v est à distance k_2 de w .

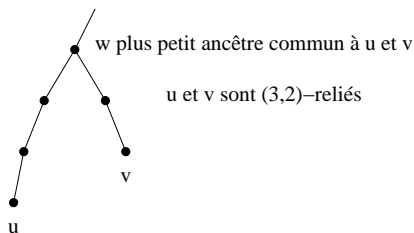


Schéma k -relationnel

Définition

Un schéma k -relationnel est une représentation distribuée permettant de déterminer si deux nœuds sont (k_1, k_2) -reliés en examinant seulement leurs étiquettes

Un schéma 1-relationnel permet de tester si deux sommets sont:

- ▶ identiques: $(0, 0)$ -reliés
- ▶ adjacents: $(1, 0)$ -reliés ou $(0, 1)$ -reliés
- ▶ frères: $(1, 1)$ -reliés

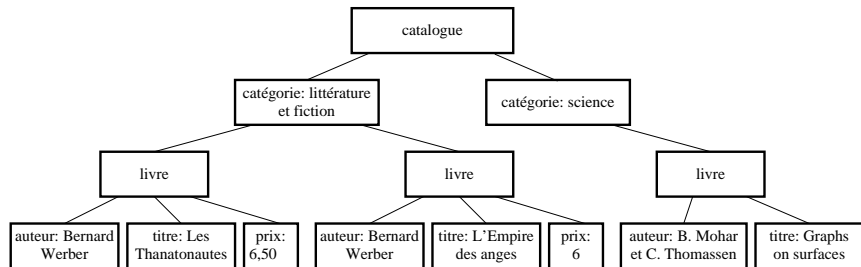
Schéma k -relationnel et de distance

Deux sommets u et v sont à distance d si et seulement s'il existe $j \in \{0, \dots, k\}$ tels que u et v sont $(j, d - j)$ -reliés.

\Rightarrow Un schéma k -relationnel permet donc le calcul des distances.

Format XML

XML est un format de fichier permettant de stocker des bases de données dont la structure globale est un arbre.



Recherche XML structurée et précalcul

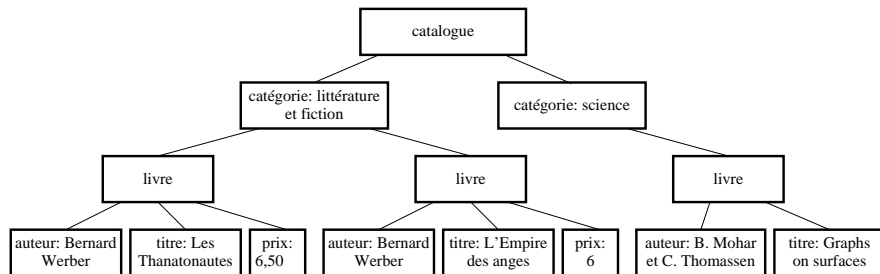
On a besoin d'une structure de données supportant de manière efficace des requêtes structurées comme la recherche d'un livre d'un auteur et d'un prix précis.

Solution

- ▶ Ajouter au document XML une grande table de hashage contenant tous les mots et les items du document.
- ▶ On associe à chaque entrée de la table les étiquettes des nœuds de l'arbre contenant le mot ou l'item.
- ▶ En donnant aux nœuds de l'arbre des étiquettes permettant de tester des k -relations, le calcul des requêtes se résume à rechercher les étiquettes des nœuds correspondant à la recherche et de sélectionner ceux ayant les bonnes relations.

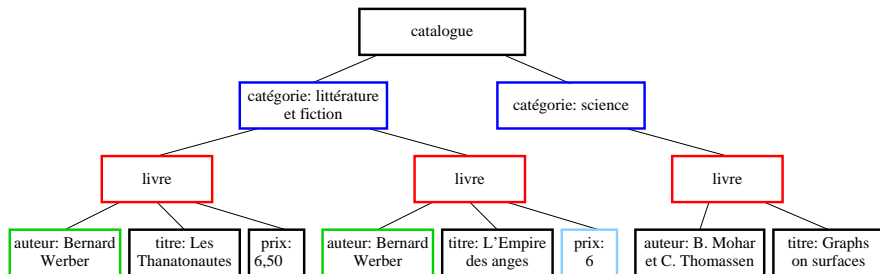
Exemple de requête structurée

Recherche de la catégorie des livres ayant
un prix de 6 et comme auteur Werber



Exemple de requête structurée

Recherche de la catégorie des livres ayant
un prix de 6 et comme auteur Werber



Condition de (k_1, k_2) -relation

$A_u[d]$: ancêtre à distance d de u .

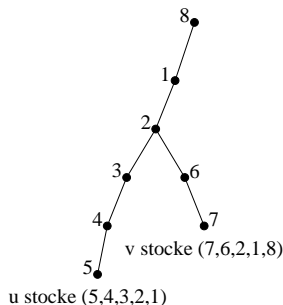
Condition de (k_1, k_2) -relation

u et v sont (k_1, k_2) -reliés $\Leftrightarrow A_u[k_1] = A_v[k_2]$ et $A_u[k_1 - 1] \neq A_v[k_2 - 1]$.

Solution triviale

Schéma k -relationnel trivial

- ▶ attribuer à chaque nœuds de l'arbre un entier distinct servant d'identifiant.
- ▶ stocker dans l'étiquette de chaque nœud u un tableau contenant les identifiants de $A_u[0], A_u[1], \dots, A_u[k]$.



Travaux existants

Théorème (Astrup, Bille, and Rauhe 2003)

- ▶ *Il existe un schéma k -relationnel avec des étiquettes de $\log n + O(k^2 \log(k \log n))$ bits.*
- ▶ *$\log n + \log \log n + \Omega(1)$ bits par nœud sont nécessaire à un schéma 1-relationnel (test frères).*

Notre résultat

Théorème

Il existe un schéma k -relationnel avec des étiquettes de $\log n + O(k \log(k \log(\frac{n}{k})))$ bits.

Terme de second ordre : $k^2 \log(k \log n) \rightarrow k \log(k \log(\frac{n}{k}))$

Pour $k = 1$: $5 \log \log n \rightarrow 2 \log \log n$

Notre schéma

- ▶ Le principe de notre schéma est le même que celui du schéma trivial : stocker dans l'étiquette de chaque nœud u des identifiants de $A_u[0], A_u[1], \dots, A_u[k]$.
- ▶ La difficulté du schéma réside dans le codage de ses identifiants de manière compacte. On a besoin pour cela d'une décomposition spéciale du graphe.

Décomposition k -ascendante

Lemme (Décomposition k -ascendante)

Pour tout k , n'importe quel arbre T de n sommets admet une partition de ses sommets telle que:

- ▶ *les parties de la partition forment les nœuds d'un arbre binaire B de hauteur $\leq \log(n/k)$,*
- ▶ *chaque partie contient au plus $(k + 1) \log n$ sommets de T ,*
- ▶ *pour chaque $u \in V(T)$, les parties correspondant aux sommets $A_u[0], A_u[1], \dots, A_u[k]$ appartiennent à un chemin de la racine à une feuille de B .*

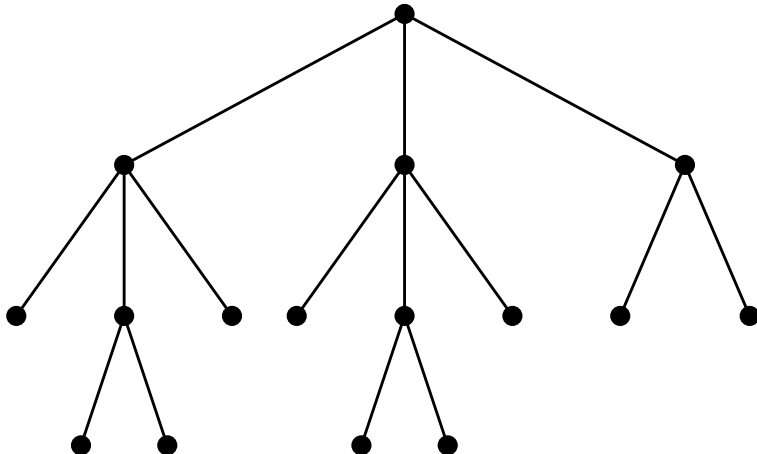
Comment obtenir la décomposition k -ascendante?

Étape 1: Relier chaque sommet de T à ses k plus proche ancêtres.
⇒ on obtient ainsi un graphe T_k de largeur arborescente k .

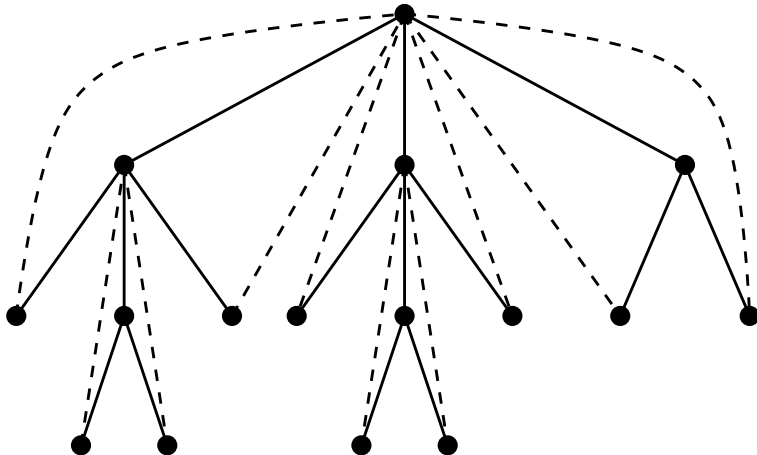
Étape 2: Découper le graphe T_k récursivement en deux parties égales grâce à des séparateurs.

⇒ on obtient ainsi une décomposition en un arbre binaire où les parties sont les séparateurs.

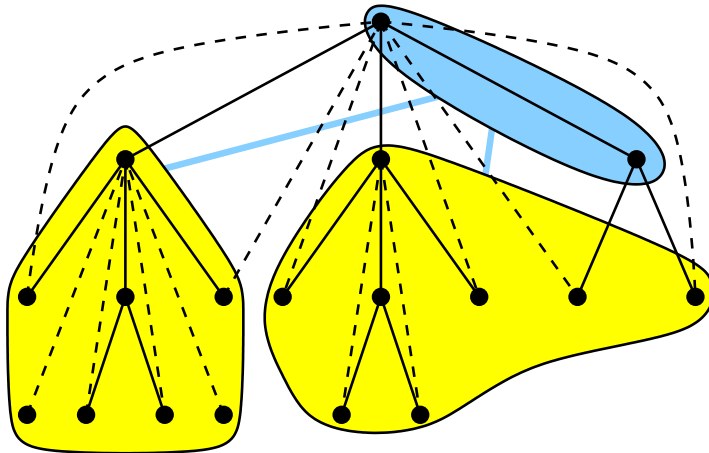
Comment obtenir la décomposition k -ascendante?



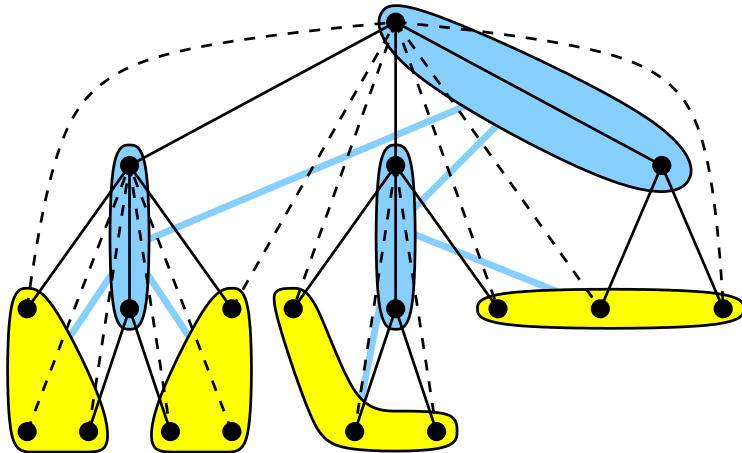
Comment obtenir la décomposition k -ascendante?



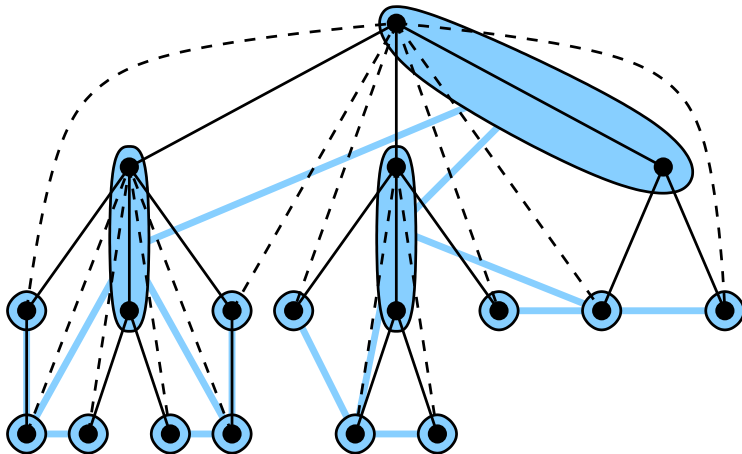
Comment obtenir la décomposition k -ascendante?



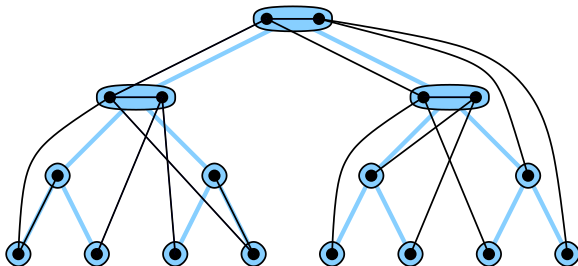
Comment obtenir la décomposition k -ascendante?



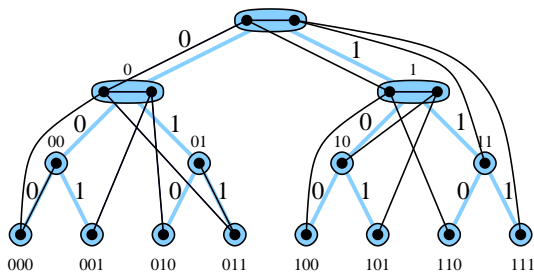
Comment obtenir la décomposition k -ascendante?



De l'utilité de la bidécomposition pour le codage



De l'utilité de la bidécomposition pour le codage



De l'utilité de la bidécomposition pour le codage

Identifiant de u : chemin de la racine de T à la partie contenant u plus la position de u dans sa partie.

Étiquette de u : chemin w_u ($\leq \log n$ bits) contenant toutes les parties contenant les sommets $A_u[0], A_u[1], \dots, A_u[k]$ plus la profondeur et la position de chacun de ces sommets ($O(k \log k \log(n/k))$ bits).

De l'étiquette de u , il est possible d'extraire tous les identifiants des sommets de $A_u[0], A_u[1], \dots, A_u[k]$ car le chemin de la partie d'un sommet $A_u[i]$ est le préfixe de w_u de longueur la profondeur de $A_u[i]$.

Problèmes ouverts

Notre schéma implique que les distances en $o(\log n / \log \log n)$ peuvent être calculées avec $\log n + o(\log n)$ bits par nœud.

On peut se demander s'il est possible de:

- ▶ concevoir un schéma de distance en $\log n + o(\log n)$ bits pour des plus grandes distances, par exemple en $\log n$.
- ▶ concevoir un schéma de distance bornée en $\log n + o(\log n)$ bits pour les graphes de largeur arborescente bornée.

Perspectives

Dans notre modèle, l'ajout d'un sommet dans l'arbre implique un ré-étiquetage complet afin de maintenir la représentation.

Modèle dynamique

Une représentation dynamique qui maintient les étiquettes avec une complexité amortie de $O(\log n)$ messages par modification.

Difficulté : maintenir la décomposition k -ascendante.