Analyse statique de requêtes

Cours L3 Bases de Données

Pierre Senellart

19 avril 2017

Plan

Inclusion et équivalence

Requêtes conjonctives

Calcul relationnel

Références

Optimisation de requêtes

- But : Étant donnée une requête q dans un certain langage $\mathcal{Q}$ et une base de données D, trouver une requête équivalente à q sur D et plus rapide à exécuter sur D
- Dans cette séance : $\mathcal{Q}$ est le calcul relationnel (ou un de ses fragments), et on cherche une requête plus rapide sur n'importe quelle base de données (on ne regarde pas D, on parle d'analyse statique)
- Séance à venir : $\mathcal{Q}$ est l'ensemble des plans d'exécutions de requête (une spécialisation de l'algèbre relationnelle où l'on choisit des implémentations pour chacun des opérateurs) et on utilise des statistiques sur D

Optimisation globale

- On considère dans cette séance des techniques d'optimisation globales, considérant la requête dans sa totalité (les techniques qu'on verra sur les plans d'exécutions sont plus locales)
- On a donc besoin de définir formellement :

  Équivalence : $q \equiv q'$ si pour toute base de données D, $q(D) = q'(D)$

  Minimalité : q' est la « meilleure » requête équivalente à q dans $\mathcal{Q}$

Inclusion et équivalence

Définition

Une requête q est incluse dans une requête q' (noté $q \sqsubseteq q'$) si pour toute base de données D, $q(D) \subseteq q'(D)$

Inclusion et équivalence

Définition

Une requête q est incluse dans une requête q' (noté $q \sqsubseteq q'$) si pour toute base de données D, $q(D) \subseteq q'(D)$

Proposition

$q \equiv q'$ si et seulement si $q \sqsubseteq q'$ et $q' \sqsubseteq q$.

Démonstration.

Immédiat. □

Plan

Inclusion et équivalence

Requêtes conjonctives

Calcul relationnel

Références

Cas des requêtes conjonctives

- On considère des requêtes conjonctives (CQ) de la forme :

$$q(\mathbf{x}) \leftarrow \exists \mathbf{y} : R_1(\mathbf{z}_1) \wedge \cdots \wedge R_n(\mathbf{z}_n)$$

  où chaque $\mathbf{z}_i$ est un tuple de variables parmi $\mathbf{x}$ et $\mathbf{y}$ et où chaque x_j apparaît dans au moins un $\mathbf{z}_j$
- Sémantique ensembliste : pour toute base de données D, $q(D)$ est un ensemble fini de tuples

Homomorphisme

Définition

Un homomorphisme d'une CQ q dans une CQ q' est une fonction φ des variables $\mathbf{x}, \mathbf{y}$ de q vers les variables $\mathbf{x}', \mathbf{y}'$ de q' telle que :

- $\varphi(\mathbf{x}) = \mathbf{x}'$
- pour tout atome $R(\mathbf{z}_i)$ de q, il existe un atome $R(\mathbf{z}'_{i'})$ tel que $\varphi(\mathbf{z}_i) = \mathbf{z}'_{i'}$

Définition

Un homomorphisme est un isomorphisme s'il est bijectif et que sa réciproque est un homomorphisme.

Instance associée à une requête

Définition

Pour toute requête conjonctive

$$q(\mathbf{x}) \leftarrow \exists \mathbf{y} : R_1(\mathbf{z}_1) \wedge \cdots \wedge R_n(\mathbf{z}_n)$$

on peut construire l'instance associée à q, notée I_q, dans laquelle le domaine actif est $\{ a_z \mid z \in \mathbf{x} \cup \mathbf{y} \}$ et formée des n tuples $R(a_{z_{i1}, \ldots, z_{ik}})$ pour $R(z_{i1}, \ldots, z_{ik})$ atome de q

Instance associée à une requête

Définition

Pour toute requête conjonctive

$$q(\mathbf{x}) \leftarrow \exists \mathbf{y} : R_1(\mathbf{z}_1) \wedge \cdots \wedge R_n(\mathbf{z}_n)$$

on peut construire l'instance associée à q, notée I_q, dans laquelle le domaine actif est $\{ a_z \mid z \in \mathbf{x} \cup \mathbf{y} \}$ et formée des n tuples $R(a_{z_{i1}, \ldots, z_{ik}})$ pour $R(z_{i1}, \ldots, z_{ik})$ atome de q

Proposition

Pour toutes CQ $q(\mathbf{x})$, $q'(\mathbf{x}')$, il existe un homomorphisme de q dans q' ssi $(a_{x'_1}, \ldots, a_{x'_j}) \in q(I_{q'})$.

Démonstration.

Direct. □

Théorème d'homomorphisme

Théorème ([Chandra and Merlin, 1977])

Pour toutes CQ q, q', $q \sqsubseteq q'$ si et seulement s'il existe un homomorphisme de q' dans q.

Théorème d'homomorphisme

Théorème ([Chandra and Merlin, 1977])

Pour toutes CQ q, q', $q \sqsubseteq q'$ si et seulement s'il existe un homomorphisme de q' dans q.

Démonstration.

- $\Leftarrow$ Immédiat.
- $\Rightarrow$ Au tableau. On utilise la caractérisation des homomorphismes par évaluation de requête.

□

Requête minimale

Définition

Une requête conjonctive est minimale si elle comporte un nombre minimum d'atomes parmi toutes les requêtes conjonctives équivalentes.

Requête minimale

Définition

Une requête conjonctive est minimale si elle comporte un nombre minimum d'atomes parmi toutes les requêtes conjonctives équivalentes.

- Traduction d'une CQ vers une requête de l'algèbre : s'il y a n atomes, on obtient $n - 1$ jointures
- Les jointures sont une des opérations les plus coûteuses de l'algèbre relationnelle
- Trouver une requête minimale revient à faire une optimisation globale

Unicité de la requête minimale

Proposition ([Chandra and Merlin, 1977])

Soit q une CQ. Alors il existe une requête q' obtenue en enlevant des atomes à q qui est minimale.

Démonstration.

Au tableau. Considérer une requête minimale équivalente à q et appliquer le théorème d'homomorphisme. □

Unicité de la requête minimale

Proposition ([Chandra and Merlin, 1977])

Soit q une CQ. Alors il existe une requête q' obtenue en enlevant des atomes à q qui est minimale.

Démonstration.

Au tableau. Considérer une requête minimale équivalente à q et appliquer le théorème d'homomorphisme. □

Proposition ([Chandra and Merlin, 1977])

Soient q, q' deux CQ minimales équivalentes. Alors il existe un isomorphisme de q dans q'.

Démonstration.

Au tableau. On applique le théorème d'homomorphisme. L'image par homomorphisme est une requête minimale équivalente. □

Algorithme de minimisation

On peut donc appliquer la procédure suivante pour minimiser une requête :

> *Pour chaque atome de la requête, tester s'il existe une requête équivalente ne contenant pas cet atome, et donc s'il existe un homomorphisme envoyant cet atome vers un autre atome de la requête. Si oui, le supprimer, et recommencer jusqu'à obtenir une requête minimale.*

Aspects de complexité

Proposition

Les problèmes suivants sont NP-complets :

- *étant données deux CQ q, q', déterminer si $q \sqsubseteq q'$*
- *étant données deux CQ q, q', déterminer si $q \equiv q'$*
- *étant donnée une CQ q, déterminer si q est non minimale*

Démonstration.

Au tableau. La NP-difficulté est par réduction depuis 3-coloriabilité, comme pour la complexité combinée de l'évaluation de requêtes. L'appartenance à NP se montre directement. □

Aspects de complexité

Proposition

Les problèmes suivants sont NP-complets :

- *étant données deux CQ q, q', déterminer si $q \sqsubseteq q'$*
- *étant données deux CQ q, q', déterminer si $q \equiv q'$*
- *étant donnée une CQ q, déterminer si q est non minimale*

Démonstration.

Au tableau. La NP-difficulté est par réduction depuis 3-coloriabilité, comme pour la complexité combinée de l'évaluation de requêtes. L'appartenance à NP se montre directement. □

NP-difficile... en les requêtes. Les requêtes peuvent être suffisamment petites pour qu'un algorithme exponentiel ne soit

Sémantique multi-ensembliste [Chaudhuri and Vardi, 1993]

- En pratique, les SGBD implémentent une sémantique multi-ensembliste
- Deux requêtes en sémantique multi-ensembliste sont équivalentes ssi elles sont isomorphes (intuitivement, parce que deux requêtes similaires mais non isomorphes peuvent produire un nombre différents de résultats)
- Inclusion de requêtes : Π_2^{P}-difficile. Décidabilité (et complexité précise le cas échéant) : ouvert !

En pratique dans les SGBD

- L'algorithme de minimisation n'est pas implémenté, pour de nombreuses (plus ou moins bonnes) raisons :
  - La plupart des requêtes ont une sémantique multi-ensembliste
  - Algorithme exponentiel
  - L'algorithme de minimisation ne marche que pour les CQ (donc pas de négation, d'agrégation, d'union...)
  - Les SGBD sont très conservateurs dans leur approche de l'optimisation de requêtes, ne veulent pas causer de régression
- À la place, les SGBD utilisent des optimisations locales des plans d'exécution basées sur des statistiques (cf. cours ultérieur)
- Exemple assez frappant du fossé entre théorie des BD et BD systèmes

Plan

Inclusion et équivalence

Requêtes conjonctives

Calcul relationnel

Références

Satisfiabilité du calcul relationnel

Définition

Une requête booléenne q du calcul relationnel est satisfiable s'il existe une base de données (finie) D telle que $D \models q$.

Satisfiabilité du calcul relationnel

Définition

Une requête booléenne q du calcul relationnel est satisfiable s'il existe une base de données (finie) D telle que $D \models q$.

Théorème ([Trakhtenbrot, 1963])

La satisfiabilité du calcul relationnel (dans le cadre fini) est indécidable.

Démonstration.

Admis. Réduction possible depuis le problème de correspondance de POST, technique, voir [Abiteboul et al., 1995]. □

Inclusion et équivalence de requêtes du calcul

Théorème

L'inclusion et l'équivalence de requêtes du calcul relationnel sont indécidables et co-récursivement énumérables.

Inclusion et équivalence de requêtes du calcul

Théorème

L'inclusion et l'équivalence de requêtes du calcul relationnel sont indécidables et co-récursivement énumérables.

Démonstration.

L'indécidabilité est par réduction directe depuis l'indécidabilité de la satisfiabilité.

La co-récursive énumérabilité se montre directement, en énumérant les contre-exemples possibles. □

Plan

Inclusion et équivalence

Requêtes conjonctives

Calcul relationnel

Références

Références

- L'article historique sur le théorème d'homomorphisme [Chandra and Merlin, 1977]
- Base pour travailler sur un des problèmes ouverts de théorie des BD les plus majeurs [Chaudhuri and Vardi, 1993]
- Chapitre 6 de [Abiteboul et al., 1995]

Bibliographie I

Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. ISBN 0-201-53771-0. URL `http://www-cse.ucsd.edu/users/vianu/book.html`.

Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, pages 77–90, 1977. doi : 10.1145/800105.803397. URL `http://doi.acm.org/10.1145/800105.803397`.

Surajit Chaudhuri and Moshe Y. Vardi. Optimization of *Real* conjunctive queries. In *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 25-28, 1993, Washington, DC, USA*, pages 59–70, 1993. doi : 10.1145/153850.153856. URL `http://doi.acm.org/10.1145/153850.153856`.

Bibliographie II

Boris A. Trakhtenbrot. Impossibility of an algorithm for the decision problem in finite classes. *American Mathematical Society Translations Series 2*, 23 :1–5, 1963.