

# Analyse statique de requêtes

## Cours L3 Bases de Données

Pierre Senellart



1<sup>er</sup> avril 2020



# 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  $\mathbf{z}_j$

## 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

## 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
- Correspond essentiellement à des requêtes

**SELECT DISTINCT** ...

**FROM** ...

**WHERE** ...

sans autres opérateurs ou fonctions SQL

# Homomorphisme

## Définition

Un **homomorphisme** d'une CQ  $q$  dans une CQ  $q'$  est une fonction  $\varphi$  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 :

- l'ensemble des constantes utilisées (le **domaine actif**) est  $\{a_z \mid z \in \mathbf{x} \cup \mathbf{y}\}$
- les tuples sont de la forme  $R(a_{z_{i1}}, \dots, a_{z_{ik}})$  pour chaque  $R(z_{i1}, \dots, z_{ik})$  atome de  $q$

## Test d'homomorphisme

### Proposition

Soit deux CQ

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

$$q'(\mathbf{x}') \leftarrow \exists \mathbf{y}' : R'_1(\mathbf{z}'_1) \wedge \cdots \wedge R'_{n'}(\mathbf{z}'_{n'})$$

avec  $\mathbf{x}' = (x'_1, \dots, x'_j)$ . Il existe un homomorphisme de  $q$  dans  $q'$  ssi  $(a_{x'_1}, \dots, a_{x'_j}) \in q(I_{q'})$ .

Démonstration.

## Test d'homomorphisme

### Proposition

Soit deux CQ

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

$$q'(\mathbf{x}') \leftarrow \exists \mathbf{y}' : R'_1(\mathbf{z}'_1) \wedge \cdots \wedge R'_{n'}(\mathbf{z}'_{n'})$$

avec  $\mathbf{x}' = (x'_1, \dots, x'_j)$ . Il existe un homomorphisme de  $q$  dans  $q'$  ssi  $(a_{x'_1}, \dots, a_{x'_j}) \in q(I_{q'})$ .

### Démonstration.

⇐ Supposons  $(a_{x'_1}, \dots, a_{x'_j}) \in q(I_{q'})$ .

Alors il existe  $\psi : \mathbf{x} \cup \mathbf{y} \rightarrow \{a_z \mid z \in \mathbf{x}' \cup \mathbf{y}'\}$  tel que :

- $\psi(\mathbf{x}) = (a_{x'_1}, \dots, a_{x'_j})$
- $\forall 1 \leq i \leq n, R_i(\psi(\mathbf{z}_i)) \in I_{q'}$

Soit  $f : \{a_z \mid z \in \mathbf{x}' \cup \mathbf{y}'\} \rightarrow \mathbf{x}' \cup \mathbf{y}'$  définie par  $f(a_z) = z$ . Alors  $f \circ \psi$  est un homomorphisme de  $q$  dans  $q'$ .

## Test d'homomorphisme

### Proposition

Soit deux CQ

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

$$q'(\mathbf{x}') \leftarrow \exists \mathbf{y}' : R'_1(\mathbf{z}'_1) \wedge \cdots \wedge R'_{n'}(\mathbf{z}'_{n'})$$

avec  $\mathbf{x}' = (x'_1, \dots, x'_j)$ . Il existe un homomorphisme de  $q$  dans  $q'$  ssi  $(a_{x'_1}, \dots, a_{x'_j}) \in q(I_{q'})$ .

### Démonstration.

$\Rightarrow$  Supposons qu'il existe  $\varphi : \mathbf{x} \cup \mathbf{y} \rightarrow \mathbf{x}' \cup \mathbf{y}'$  tel que :

- $\varphi(\mathbf{x}) = (x'_1, \dots, x'_j)$
- $\forall 1 \leq i \leq n, R_i(\varphi(\mathbf{z}_i))$  est l'un des atomes de  $q'$

Soit  $f' : \mathbf{x}' \cup \mathbf{y}' \rightarrow \{a_z \mid z \in \mathbf{x}' \cup \mathbf{y}'\}$  définie par  $f'(z) = a_z$ . Alors  $f' \circ \varphi : \mathbf{x} \cup \mathbf{y} \rightarrow \{a_z \mid z \in \mathbf{x}' \cup \mathbf{y}'\}$  témoigne que  $(a_{x'_1}, \dots, a_{x'_j}) \in q(I_{q'})$ .  $\square$

## Théorème d'homomorphisme

Théorème ([Chandra and Merlin, 1977])

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

$q'(\mathbf{x}') \leftarrow \exists \mathbf{y}' : R'_1(\mathbf{z}'_1) \wedge \cdots \wedge R'_{n'}(\mathbf{z}'_{n'})$

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

## Théorème d'homomorphisme

Théorème ([Chandra and Merlin, 1977])

Soit deux CQ

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

$$q'(\mathbf{x}') \leftarrow \exists \mathbf{y}' : R'_1(\mathbf{z}'_1) \wedge \cdots \wedge R'_{n'}(\mathbf{z}'_{n'})$$

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

Démonstration.

$\Leftarrow$  Soit  $h$  un homomorphisme de  $q'$  dans  $q$ . Soit  $D$  une base de données arbitraire de domaine actif  $A$  et  $t \in q(D)$ .

Soit  $\psi : \mathbf{x} \cup \mathbf{y} \rightarrow A$  la valuation correspondante.

Alors  $\psi \circ h : \mathbf{x}' \cup \mathbf{y}' \rightarrow A$  témoigne que  $t \in q'(D)$ .

## Théorème d'homomorphisme

Théorème ([Chandra and Merlin, 1977])

Soit deux CQ

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

$$q'(\mathbf{x}') \leftarrow \exists \mathbf{y}' : R'_1(\mathbf{z}'_1) \wedge \cdots \wedge R'_{n'}(\mathbf{z}'_{n'})$$

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

Démonstration.

$\Rightarrow$  D'après le test d'homomorphisme, il suffit de montrer que  $(a_{x_1}, \dots, a_{x_j}) \in q'(I_q)$  où  $\mathbf{x} = (x_1, \dots, x_j)$ .

Par construction de  $I_q$ ,  $(a_{x_1}, \dots, a_{x_j}) \in q(I_q)$ .

Comme  $q \sqsubseteq q'$ , on a aussi

$(a_{x_1}, \dots, a_{x_j}) \in q(I_q)$ .



## 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**

## Procédure d'obtention d'une 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.*

## Procédure d'obtention d'une 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.

Soit  $q''$  une requête minimale équivalente à  $q$ .

## Procédure d'obtention d'une 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.**

Soit  $q''$  une requête minimale équivalente à  $q$ .

Comme  $q'' \sqsubseteq q$  et  $q \sqsubseteq q''$ , par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q''$  et  $\psi$  de  $q''$  dans  $q$ .

## Procédure d'obtention d'une 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.**

Soit  $q''$  une requête minimale équivalente à  $q$ .

Comme  $q'' \sqsubseteq q$  et  $q \sqsubseteq q''$ , par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q''$  et  $\psi$  de  $q''$  dans  $q$ .

Alors  $h = \psi \circ \varphi$  est un homomorphisme de  $q$  dans lui-même.

Soit  $q'$  l'ensemble des atomes de  $q$  qui sont images par  $h$  d'un atome de  $q$ .

## Procédure d'obtention d'une 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.**

Soit  $q''$  une requête minimale équivalente à  $q$ .

Comme  $q'' \sqsubseteq q$  et  $q \sqsubseteq q''$ , par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q''$  et  $\psi$  de  $q''$  dans  $q$ . Alors  $h = \psi \circ \varphi$  est un homomorphisme de  $q$  dans lui-même.

Soit  $q'$  l'ensemble des atomes de  $q$  qui sont images par  $h$  d'un atome de  $q$ .

Comme  $\psi$  est un homomorphisme de  $q''$  dans  $q'$ , on a  $|q''| \geq |q'|$  où  $|\cdot|$  dénote le nombre d'atomes d'une requête. Comme  $q''$  est minimale,  $q'$  est également minimale. □

## Unicité de la requête minimale

Proposition ([Chandra and Merlin, 1977])

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

## Unicité de la requête minimale : Preuve

Par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q'$  et  $\psi$  de  $q'$  dans  $q$ .

## Unicité de la requête minimale : Preuve

Par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q'$  et  $\psi$  de  $q'$  dans  $q$ .  $q$  et  $q'$  étant toutes deux minimales, tous les atomes de  $q'$  sont images par  $\varphi$  d'un atome de  $q$ , et réciproquement.

## Unicité de la requête minimale : Preuve

Par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q'$  et  $\psi$  de  $q'$  dans  $q$ .

$q$  et  $q'$  étant toutes deux minimales, tous les atomes de  $q'$  sont images par  $\varphi$  d'un atome de  $q$ , et réciproquement.

$h = \psi \circ \varphi$  est un homomorphisme de  $q$  vers lui-même, et tous les atomes de  $q$  sont images par  $h$  d'un atome de  $q$ ; en particulier, toutes les variables de  $q$  sont dans l'image de  $h$ , et  $h$  est donc bijectif.  $h^{-1}$  étant également un homomorphisme,  $h$  est un isomorphisme de  $q$  dans  $q$ .

## Unicité de la requête minimale : Preuve

Par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q'$  et  $\psi$  de  $q'$  dans  $q$ .

$q$  et  $q'$  étant toutes deux minimales, tous les atomes de  $q'$  sont images par  $\varphi$  d'un atome de  $q$ , et réciproquement.

$h = \psi \circ \varphi$  est un homomorphisme de  $q$  vers lui-même, et tous les atomes de  $q$  sont images par  $h$  d'un atome de  $q$ ; en particulier, toutes les variables de  $q$  sont dans l'image de  $h$ , et  $h$  est donc bijectif.  $h^{-1}$  étant également un homomorphisme,  $h$  est un isomorphisme de  $q$  dans  $q$ .

Comme  $h$  est bijective,  $\varphi$  est injective et  $\psi$  est surjective;  $\varphi$  et  $\psi$  jouant des rôles analogues, on en déduit que les deux fonctions sont bijectives.

## Unicité de la requête minimale : Preuve

Par le théorème d'homomorphisme, il existe deux homomorphismes  $\varphi$  de  $q$  dans  $q'$  et  $\psi$  de  $q'$  dans  $q$ .

$q$  et  $q'$  étant toutes deux minimales, tous les atomes de  $q'$  sont images par  $\varphi$  d'un atome de  $q$ , et réciproquement.

$h = \psi \circ \varphi$  est un homomorphisme de  $q$  vers lui-même, et tous les atomes de  $q$  sont images par  $h$  d'un atome de  $q$ ; en particulier, toutes les variables de  $q$  sont dans l'image de  $h$ , et  $h$  est donc bijectif.  $h^{-1}$  étant également un homomorphisme,  $h$  est un isomorphisme de  $q$  dans  $q$ .

Comme  $h$  est bijective,  $\varphi$  est injective et  $\psi$  est surjective;  $\varphi$  et  $\psi$  jouant des rôles analogues, on en déduit que les deux fonctions sont bijectives.

$\varphi$  est donc un homomorphisme bijectif de  $q$  dans  $q'$ , et  $\varphi^{-1} = h^{-1} \circ \psi$  est un homomorphisme comme composition de deux homomorphismes. Donc  $\varphi$  est un isomorphisme.



## 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

Soient  $q, q'$  deux CQ. Ces problèmes sont **NP-complets** :

- déterminer si  $q \sqsubseteq q'$
- déterminer si  $q \equiv q'$
- déterminer si  $q$  est non minimale

### Démonstration.

- Réduction PTIME de (a) vers (b) ( $q \sqsubseteq q'$  ssi  $q \equiv q \wedge q'$ )
- Réduction PTIME de (b) vers (a) (pour tester si  $q \sqsubseteq q'$  et  $q' \sqsubseteq q$ , si  $q$  et  $q'$  sont connectées et non booléennes, on peut le faire avec un seul test  $q \wedge \bar{q}' \sqsubseteq q' \wedge \bar{q}$  où  $\bar{q}$  et  $\bar{q}'$  sont des copies de  $q, q'$  avec de nouvelles variables ; généralisation au cas non-connecté ou booléen en exercice)
- (c) :  $|q|$  tests de (b)

Il suffit donc de montrer (a) dans NP et (a) et (c) NP-difficiles.

## Inclusion dans NP : Preuve

Soient deux CQ  $q$  et  $q'$ .

Pour déterminer si  $q \sqsubseteq q'$  :

- Proposer de manière non-déterministe une fonction des variables de  $q'$  vers  $q$  (proposition de taille polynomiale en la taille de  $q, q'$ ).
- Vérifier en temps polynomial que c'est un homomorphisme (ce qui implique que  $q \sqsubseteq q'$  par le théorème d'homomorphisme); si oui, accepter.

Cette procédure est un algorithme non-déterministe en temps polynomial.

## Inclusion / non-minimalité NP-difficile : Preuve

On réduit depuis 3-coloriabilité, un problème NP-difficile.

## Inclusion / non-minimalité NP-difficile : Preuve

On réduit depuis 3-coloriabilité, un problème NP-difficile.

Soit un graphe non-orienté connexe  $G = (N, E)$  avec  $N = \{x_1 \dots x_n\}$ . On construit une requête booléenne  $q_G$  de domaine actif  $N \cup \{r, v, b\}$  dont les atomes sont les suivants :

- $E(x, y)$  pour chaque  $\{x, y\} \in E$
- $N_i(x_i)$  pour chaque  $x_i \in N$
- $R(r), V(v), B(b)$
- $E(r, v), E(v, r), E(r, b), E(b, r), E(b, v), E(v, b)$
- $N_i(r), N_i(v), N_i(b)$  pour chaque  $1 \leq i \leq n$

Alors  $q_G \setminus \{N_1(x_1)\} \sqsubseteq q_G$  ssi  $q_G$  est non-minimal ssi  $G$  est 3-coloriable.

Cette réduction est PTIME. □

## Remarque sur la complexité

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

## 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 :  $\Pi_2^P$ -difficile. Décidabilité (et complexité précise le cas échéant) : **problème 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, partie 6.3]. □

# 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.

On a déjà vu des réductions PTIME entre inclusion et équivalence, dans les deux sens. On traite juste l'inclusion.

# 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.

On a déjà vu des réductions PTIME entre inclusion et équivalence, dans les deux sens. On traite juste l'inclusion.

**indécidable.**  $q$  est insatisfiable ssi  $q \sqsubseteq \perp$  où  $\perp$  est une requête ne renvoyant jamais de résultats (p. ex.,  $\exists x, y : R(x, y) \wedge \neg R(x, y)$ ).

# 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.

On a déjà vu des réductions PTIME entre inclusion et équivalence, dans les deux sens. On traite juste l'inclusion.

**indécidable.**  $q$  est insatisfiable ssi  $q \sqsubseteq \perp$  où  $\perp$  est une requête ne renvoyant jamais de résultats (p. ex.,  $\exists x, y : R(x, y) \wedge \neg R(x, y)$ ).

**co-r.é.** On énumère l'ensemble des bases de données possibles (chaque base de données est un objet **fini**). Si on en trouve une telle que  $q(D) \not\subseteq q'(D)$ , c'est une preuve que  $q \not\subseteq q'$ . Donc la non-inclusion est récursivement énumérable. □

# 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.