932 Fondements des bases de données relationnelles.

* Motivations : gestion de comptes en banque / réservation d'hôtels / ...

I] Bases de données : formalisme.

def 1 : On se donne un ensemble ordonné att (les attributs) et des ensembles dom (les constantes), relname (les noms de relations), var (les variables) tous dénombrables et deux à deux disjoints.
On se donne aussi une fonction $\text{sort} : \text{relname} \to \mathcal{P}_f(\text{att})$ vérifiant :
$\forall U \in \mathcal{P}_f(\text{att}), \ |\text{sort}^{-1}(U)| = +\infty$.

def 2 : un schéma de relation est un élément $R \in \text{relname}$. Son arité est $ar(R) = |\text{sort}(R)|$. Si $U = \text{sort}(R)$, on notera $R[U]$.

def 3 : un schéma de base de données est un ensemble non-vide de schémas de relation. On notera $\mathcal{R} = \{R_1[U_1], \dots, R_n[U_n]\}$.

def 4 : un tuple (libre) sur $R[U]$ est une fonction $u : U \to \text{dom}$ ($u : U \to \text{dom} \cup \text{var}$).
Si $n = ar(R)$, on pourra considérer que $u \in \text{dom}^n$ ($(\text{dom} \cup \text{var})^n$) s'il n'y a pas ambiguïté.

def 5 : une relation I sur $R[U]$ est un ensemble fini de tuples sur $R[U]$.

def 6 : une base de données $\mathcal{I}$ sur $\mathcal{R}$ est une fonction $\mathcal{R} \to \{\text{relations}\}$
$R \mapsto \mathcal{I}(R)$

ex 7 : [Annexe 1] On s'autorise les deux notations :
<Titre : 'Au Poste', Réalisateur : 'Quentin Dupieux', Acteur : 'Marc Fraize'>
<'Au Poste', 'Quentin Dupieux', 'Marc Fraize'>

II] Le langage des requêtes conjonctives.

1) Règles conjonctives.

def 8 : une règle conjonctive sur $\mathcal{R} = \{R_1[U_1], \dots, R_n[U_n]\}$ est une expression de la forme :

$$q : \ \text{res}(u) \leftarrow R_1(u_1), \dots, R_n(u_n)$$

où $u, u_1, \dots, u_n$ sont des tuples libres et $\text{res} \in \text{relname} \setminus \mathcal{R}$.

ex 9 : la requête R_1 : "Quels films Damien Chazelle a-t-il réalisé ?" est exprimée par la règle conjonctive $q_1 : \text{res}(x) \leftarrow \text{Flims}(x, \text{'Damien Chazelle'}, z)$

def 10 : une valuation ν sur $V \in \mathcal{P}_f(\text{var})$ est une fonction $\nu : V \to \text{dom}$, étendue à $\nu : V \cup \text{dom} \to \text{dom}$ par $\forall x \in \text{dom}, \ \nu(x) = x$.

def 11 : le domaine actif d'une base de données $\mathcal{I}$, d'une relation I, d'une règle conjonctive q est l'ensemble $\text{adom}(\mathcal{I})$, $\text{adom}(I)$, $\text{adom}(q)$ des constantes apparaissant dans $\mathcal{I}, I, q$.

def 12 : (sémantique des règles conjonctives) Soit $\mathcal{I}$ une base de données et q une règle conjonctive.

$$q(\mathcal{I}) = \{\nu(u), \ \nu \text{ valuation sur } var(q) \text{ et } \forall i \in [\![1, n]\!], \ \nu(u_i) \in \mathcal{I}(R_i)\}$$

prop 13 : $q(\mathcal{I})$ est une relation et $\text{adom}(q(\mathcal{I})) \subseteq \text{adom}(q) \cup \text{adom}(\mathcal{I})$

prop 14 : les requêtes conjonctives sont monotones et satisfiables. ie : pour toute règle q sur $\mathcal{R}$:
(i) $\forall \mathcal{I}, \mathcal{J}$ sur $\mathcal{R}$, $(\mathcal{I} \subseteq \mathcal{J}) \Rightarrow (q(\mathcal{I}) \subseteq q(\mathcal{J}))$
(ii) $\exists \mathcal{I}$ sur $\mathcal{R}$ / $q(\mathcal{I}) \neq \emptyset$

ex 15 : la requête R_2 : "Quels cinémas de Lille ne diffusent que TAS ?" n'est pas monotone.

2) Requêtes par tableaux.

def 16 : un tableau sur $R[U]$ est un ensemble fini de tuples libres sur $R[U]$. une table est un ensemble non-vide de tableaux. Une requête par tableaux sur $R[U]$ est un couple (T, u) où T est une table et u un tuple libre tq $var(u) \subseteq var(T)$.

ex 17 : la requête R_1 est exprimée par tableaux dans [Annexe 2].

Rq 18 : on définit la sémantique d'une requête par tableaux de manière analogue.

3) Calcul conjonctif (cc).

def 19 : un atome sur $R[U]$ est une expression $R(e_1, \dots, e_n)$ où $e_i \in \text{dom} \cup \text{var}$ et $n = ar(R)$.
L'ensemble $\mathcal{F}^c_{\mathcal{R}}$ des formules du calcul conjonctif sur $\mathcal{R}$ est défini par induction par :
- $\mathcal{F}^c_{\mathcal{R}}$ contient les atomes sur chaque $R[U] \in \mathcal{R}$.
- Si $\varphi, \psi \in \mathcal{F}^c_{\mathcal{R}}$, alors $(\varphi \wedge \psi \in \mathcal{F}^c_{\mathcal{R}})$
- Si $x \in \text{var}$ et $\varphi \in \mathcal{F}^c_{\mathcal{R}}$, alors $(\exists x \varphi \in \mathcal{F}^c_{\mathcal{R}})$

def 20 : une occurrence de la variable x est dite libre dans $\varphi \in \mathcal{F}^c_{\mathcal{R}}$ si :
- φ est atome
- $\varphi = \psi \wedge \xi$ et x est libre dans ψ et dans ξ.
- $\varphi = \exists y \psi$ et $y \neq x$ et x est libre dans ψ.

on note $\text{Free}(\varphi)$ les variables libres dans φ.

def 21 : une requête du calcul conjonctif sur $\mathcal{R}$ est une expression de la forme :

$$q : \ \{\langle e_1, \dots, e_m \rangle : A_1, \dots, A_m \mid \varphi\}$$

où : $\varphi \in \mathcal{F}^c_{\mathcal{R}}$; $\langle e_1 \dots e_m \rangle$ tuple libre dont les variables sont exactement $\text{Free}(\varphi)$.

Rq 22 : on notera aussi $\{e_1, \dots, e_m \mid \varphi\}$ si les attributs sont clairs dans le contexte.

ex 23 : la requête R_1 est exprimée par $\{x \mid \exists z \ \text{Flims}(x, \text{'Damien Chazelle'}, z)\}$

def 24 : Soit $\varphi \in \mathcal{F}^c_{\mathcal{R}}$, ν une valuation sur $\text{Free}(\varphi)$ et $\mathcal{I}$ une base de données sur $\mathcal{R}$.
on dit que $\mathcal{I}$ satisfait φ sous ν (et on note $\mathcal{I} \models \varphi[\nu]$) si :
- $\varphi = R(u)$ et $\nu(u) \in \mathcal{I}(R)$.
- $\varphi = \psi \wedge \xi$ et $(\mathcal{I} \models \psi[\nu_{|\text{Free}(\psi)}]$ et $\mathcal{I} \models \xi[\nu_{|\text{Free}(\xi)}]$.
- $\varphi = \exists x \psi$ et il existe $c \in \text{dom}$ tq $\mathcal{I} \models \psi[\nu \cup \{x \mapsto c\}]$.

def 25 : (sémantique du calcul conjonctif)
Soit $\mathcal{I}$ une base de données sur $\mathcal{R}$ et q une requête du cc avec les notations de def 21.

$$q(\mathcal{I}) = \{\nu(\langle e_1, \dots, e_n \rangle) \mid \mathcal{I} \models \varphi[\nu], \ \nu \text{ valuation sur } \text{Free}(\varphi)\}$$

prop 26 : $q(\mathcal{I})$ est une relation et $\text{adom}(q(\mathcal{I})) \subseteq \text{adom}(q) \cup \text{adom}(\mathcal{I})$.

def 27 : Deux formules $\varphi, \psi \in \mathcal{F}^c_{\mathcal{R}}$ sont sémantiquement équivalentes si $\text{Free}(\varphi) = \text{Free}(\psi)$ et $\mathcal{I} \models \varphi[\nu] \Leftrightarrow \mathcal{I} \models \psi[\nu]$ pour tous $\mathcal{I}, \nu$.

prop 28 : toute formule $\varphi \in \mathcal{F}^c_{\mathcal{R}}$ est sémantiquement équivalente à une formule $\tilde{\varphi} \in \mathcal{F}^c_{\mathcal{R}}$ en forme normale : $\tilde{\varphi} = \exists x_1 \dots \exists x_m \ R_1(x_1) \wedge \dots \wedge R_n(x_m)$

thm 29 : (équivalence des langages de requêtes conjonctives)

Pour tout schéma de base de données $\mathcal{R}$, pour toute q_1 règle conjonctive / requête par tableaux / requête du calcul conjonctif,
il existe q_2 une règle conjonctive / requête par tableaux / requête du calcul conjonctif telle que $q_1(\mathcal{I}) = q_2(\mathcal{I})$.

Rq 30 : on peut ajouter l'égalité en prenant garde aux requêtes :
→ fournissant une réponse infinie : $res(x,y) \leftarrow$ CinéLille$(x, \text{'TAS'}, y) \wedge y = z$
→ insatisfiables : $res(x,y) \leftarrow y = \text{'10h00'} \wedge y = \text{'13h00'}$

Thm 31 : (clôture par composition)
Toute succession de requêtes conjonctives peut s'exprimer à l'aide d'une seule requête.

III] Opérateurs algébriques élémentaires

1) L'algèbre SPC

def 32 : Soient I, J deux relations sur $R[U], S[V]$. On définit les opérateurs suivants.

* sélection
$\sigma_{A=c}(I) = \{t \in I \text{ tq } t(A) = c\}$ où $c \in$ dom
$\sigma_{A=B}(I) = \{t \in I \text{ tq } t(A) = t(B)\}$ et $A, B \in U$.

* projection
$\pi_{A_{j_1} \dots A_{j_k}}(I) = \{\langle A_{j_1} : u_{j_1}, \dots, A_{j_k} : u_{j_k} \rangle, \langle A_1 : u_1, \dots, A_n : u_n \rangle \in I\}$ où $U = \{A_1, \dots, A_n\}$ et $A_{j_1}, \dots A_{j_k}$ deux à deux disjoints.

* produit cartésien
$I \times J = \{\langle I.A_1 : u_1, \dots, I.A_n : u_n, J.B_1 : v_1, \dots, J.B_m : v_m \rangle, \langle A_1 : u_1, \dots, A_n : u_n \rangle \in I, \langle B_1 : v_1, \dots, B_m : v_m \rangle \in J\}$ où $U = \{A_1, \dots, A_n\}$ et $V = \{B_1, \dots, B_m\}$

def 33 : l'algèbre SPC est le plus petit langage contenant les atomes, les constantes $\{\langle a \rangle\}$ $a \in$ dom, et stable par application des symboles de sélection, projection et produit cartésien (avec arités correspondantes). La sémantique est alors définie naturellement.

prop 34 : tout élément de l'algèbre SPC est équivalent sémantiquement à un élément sous forme normale :
$$\pi_{A_{j_1} \dots A_{j_k}}\left(\{\langle a_1 \rangle\} \times \dots \times \{\langle a_n \rangle\} \times \sigma_F(R_1 \times \dots \times R_k)\right)$$
où $k > 0$; $n \geq 0$; $a_1, \dots, a_n \in$ dom ; $\{1 \dots n\} \subseteq \{j_1, \dots, j_k\}$; $R_1 \dots R_k$ relations et F un ensemble de conditions de la forme $A = c$ / $A = B$, $A, B \in$ att et $c \in$ dom.

ex 35 : l'algèbre SPC permet d'exprimer des requêtes insatisfiables :
R_3 : "Parmi les flims réalisés par Damien Chazelle, quels films Alain Chabat a-t-il réalisés ?" ⟶ $\sigma_{\text{Réalisateur = 'Alain Chabat'}}(\sigma_{\text{Réalisateur = 'Damien Chazelle'}}(\text{Flims}))$

2) Stabilité de l'algèbre par extensions.

prop-déf 36 : Soient I, J deux relations sur $R[U], S[V]$. Les opérateurs définis ci-après sont sémantiquement équivalents à des éléments de l'algèbre SPC.

* jointure naturelle
$I \bowtie J = \{t \text{ tuple sur } U \cup V \mid \exists u \in I\ \exists v \in J \text{ tq } t_{|U} = u, t_{|V} = v\}$

* renommage
$f_\delta(I) = \{\langle \delta(A_1) : u_1, \dots, \delta(A_n) : u_n \rangle\ \langle A_1 : u_1, \dots, A_n : u_n \rangle \in I\}$ où $U = \{A_1, \dots, A_n\}$ et $\delta : U \hookrightarrow$ att injective.

* intersection
$I \cap J$ si $R[U] = S[V]$.

3) Lien avec le paradigme logique.

thm 37 : (Codd 1)
Les requêtes conjonctives et les requêtes décidables de l'algèbre SPC sont équivalentes.

ex 38 : la requête R_1 est exprimée par l'élément $\pi_{\text{Titre}}(\sigma_{\text{Réalisateur = 'Damien Chazelle'}}(\text{Flims}))$.

* équi-jointure
$I \bowtie_{A_i = B_k} J = \{\langle I.A_1 : u_1, \dots, I.A_n : u_n, J.B_1 : v_1, \dots, J.B_{k-1} : v_{k-1}, J.B_{k+1} : v_{k+1}, \dots, J.B_m : v_m \rangle, \langle A_1 : u_1, \dots, A_n : u_n \rangle \in I, \langle B_1 : v_1, \dots, B_m : v_m \rangle \in J, u_i = v_k\}$
où $U = \{A_1, \dots, A_n\}$; $V = \{B_1 \dots B_m\}$; $i \in [\![1, n]\!]$; $j \in [\![1, m]\!]$

Rq 39 : on ne peut pas encore exprimer des requêtes du type :
R_4 : "Quels cinémas diffusent TAS ou Réalité à 19h00 ?"
R_5 : "Dans quels films réalisés par Damien Chazelle Emma Stone ne joue-t-elle pas ?"

IV] Le calcul et l'algèbre relationnels.

1) Ajout de la réunion.

def 40 : on complète l'algèbre SPC en lui ajoutant l'opérateur réunion $\cup$ défini par $I \cup J = \{\langle x_1, \dots, x_n \rangle \in I \text{ ou } J\}$ pour I, J relations sur U. On définit ainsi l'algèbre SPCU.

Rq 40 : si I et J sont finis, $I \cup J$ est finie donc la sémantique est naturelle.

def 41 : on étend la def 19 pour des formules $\varphi = \psi \vee \xi$.

ex 42 : La requête R_4 est exprimée par $q = \{\langle x \rangle \mid \text{CinéLille}(x, \text{'TAS'}, \text{'19h00'}) \vee \text{CinéLille}(x, \text{'Réalité'}, \text{'19h00'})\}$

⚠ Certaines requêtes du calcul conjonctif (muni de l'union) ont une sémantique infinie :
$q = \{x, y, z \mid \text{Cinema}(x, y) \vee \text{Cinema}(y, z)\}$

déf 43 : une requête q est dite sûre si $|q(\mathcal{I})| < +\infty$ pour toute base de données $\mathcal{I}$.

3/4

2) Ajout de la négation

def 44 : on complète l'algèbre SPCU en ajoutant la différence ensembliste $-$.

Rq 45 : Si I et J sont finis, $I - J$ est finie donc sa sémantique est naturelle.

ex 46 : la requête R_5 est exprimée par $\pi_{Titre}\left(\sigma_{Réalisateur='Damien Chazelle'}(Flims) - \sigma_{Acteur='Emma Stone'}(Flims)\right)$

def 47 : on étend la définition 41 pour des formules $\varphi = \neg\psi / \varphi = \forall x\psi$. On définit ainsi le calcul relationnel noté $\mathcal{F}^r_{\mathcal{R}}$ (on notera abusivement $q \in \mathcal{F}^r_{\mathcal{R}}$ pour une requête du calcul relationnel).

3) Sémantique du calcul relationnel.

def 48 : (sémantique sous domaine). Soit $q \in \mathcal{F}^r_{\mathcal{R}}$ et $d \subseteq dom$. Pour $\mathcal{I}$ sur $\mathcal{R}$:

$$q_d(\mathcal{I}) = \{\nu(\langle e_1, \dots, e_n \rangle) \mid \mathcal{I} \models_d \varphi[\nu],\ \nu : Free(\varphi) \to d \text{ valuation}\}$$

def 49 : une requête du calcul relationnel est indépendante du domaine si pour toute base de donnée $\mathcal{I}$ sur $\mathcal{R}$ et tout couple $d, d' \subseteq dom$, $q_d(\mathcal{I}) = q_{d'}(\mathcal{I})$.
Dans ce cas, on note $q_{dom}(\mathcal{I})$ sa sémantique.

prop 50 : Si q est indépendante du domaine, $q_{dom}(\mathcal{I}) = q_{adom(q) \cup adom(\mathcal{I})}(\mathcal{I})$ et q est sûre.

(*) ADDENDUM

thm 51 : RELSAT $\in$ RE $-$ R
Entrée : une requête $q \in \mathcal{F}^r_{\mathcal{R}}$
Sortie : oui s'il existe $\mathcal{I}$ sur $\mathcal{R}$ telle que $q(\mathcal{I}) \neq \emptyset$

DEV 1

Coro 52 : les problèmes suivants sont dans coRE $-$ R
(i) RELEQU $(q, q') \to$ oui si q et q' sont sémantiquement équivalentes.
(ii) RELSUB $(q, q') \to$ oui si $\forall \mathcal{I}$ sur $\mathcal{R}$, $q(\mathcal{I}) \subseteq q'(\mathcal{I})$
(iii) RELIND $q \to$ oui si q est indépendante du domaine.

V) Optimisation

1) Stockage.

*Pb : Bases de données de grande taille, sur disque externe (accès coûteux).

def 53 : Un B-arbre de degré minimal t est un arbre T vérifiant :
(i) chaque noeud x possède les attributs suivants :
- $x.n$ le nombre de clefs.
- $x.clef_1, \dots, x.clef_{x.n}$ les $x.n$ clefs stockées dans un ordre croissant (dans une liste)
- $x.feuille$ un booléen indiquant si le noeud est feuille.
- $x.c_1, \dots, x.c_{x.n+1}$ les $(x.n+1)$ pointeurs vers les fils.

(ii) les clefs k_i des arbres fils vérifient : $k_1 \leq x.clef_1 \leq \dots \leq x.clef_{x.n} \leq k_{x.n+1}$
(iii) Toutes les feuilles ont la même profondeur h.
(iv) le nombre de clefs d'un noeud interne est contenu entre $t-1$ et $2t-1$.
le nombre de clefs du noeud racine est contenu entre 1 et $2t-1$.

thm 54 : la hauteur h d'un B-arbre T à n clefs, de degré minimal $t \geq 2$ vérifie $h \leq \log_t\left(\frac{n+1}{2}\right)$

DEV 2

+ Complexité de la Recherche et de l'Insertion dans un B-arbre en nombre d'appels au disque.

Rq 55 : on peut aussi utiliser une table de hachage parfait si on n'a que des recherches à effectuer dans la base.

2) Optimisation locale des requêtes SPC.

Prop 56 : [Annexe 3] Complexité des opérateurs algébriques de base.

Prop 57 : les transformations suivantes préservent l'équivalence sémantique.
- $\sigma_F(\sigma_{F'}(q)) \leftrightarrow \sigma_{F \wedge F'}(q)$
- $\pi_{\vec{j}}(\pi_{\vec{k}}(q)) \leftrightarrow \pi_{\vec{l}}(q)$
- $\sigma_F(\pi_{\vec{l}}(q)) \leftrightarrow \pi_{\vec{l}}(\sigma_{F'}(q))$
- $q_1 \bowtie q_2 \leftrightarrow q_2 \bowtie q_1$
- $\sigma_F(q_1 \bowtie_G q_2) \to \sigma_F(q_1) \bowtie_G q_2$
- $\sigma_F(q_1 \bowtie_G q_2) \to q_1 \bowtie_G \sigma_{F'}(q_2)$
- $\sigma_F(q_1 \bowtie_G q_2) \to q_1 \bowtie_{G'} q_2$
- $\pi_{\vec{l}}(q_1 \bowtie_G q_2) \to \pi_{\vec{l}}(q_1) \bowtie_{G'} q_2$
- $\pi_{\vec{l}}(q_1 \bowtie_G q_2) \to q_1 \bowtie_{G'} \pi_{\vec{k}}(q_2)$

Rq 58 : En pratique, pour évaluer une requête du calcul conjonctif, le SGBD va :
① déterminer plusieurs stratégies d'évaluation à l'aide de ces transformations.
② estimer le coût de chacune d'entre elles avec des heuristiques statistiques.
③ Sélectionner et exécuter la stratégie de coût minimal.

3) Optimisation globale du CC.

def 59 : Soient q_1, q_2 deux requêtes par tableaux. $q_1 \subseteq q_2$ si $\forall \mathcal{I}$ sur $\mathcal{R}$, $q_1(\mathcal{I}) \subseteq q_2(\mathcal{I})$. On notera $q_1 \equiv q_2$ si $q_1 \subseteq q_2$ et $q_2 \subseteq q_1$.

thm 60 : TABEQU [et TABSUB] $\in$ NP-complet :
Entrée : q, q' requêtes par tableaux.
Sortie : oui si $q \equiv q'$ [oui si $q \subseteq q'$]

déf 61 : Soient $q = (T, u)$ et $q' = (T', u')$ deux requêtes par tableaux sur $\mathcal{R}$. Un homomorphisme de q' sur q est une substitution θ tq $\theta(T') \subseteq \theta(T)$ et $\theta(u') = u$.

thm 62 : (Théorème d'homomorphisme)
$q \subseteq q'$ ssi il existe un homomorphisme de q' sur q.

Appli 63 : On minimise une requête par tableaux en minimisant le nombre de lignes. On procède en cherchant des homomorphismes.

ex 64 : [Annexe 4] $q_0 \subseteq q_1 \subseteq q_2 \subseteq q_3$ d'après le théorème d'homomorphisme.

ex 65 : [Annexe 5] deux requêtes équivalentes (invisible par optimisation locale).

Rq 66 : La métho de d'évaluation d'une requête conjonctive q devient :
① Transformer q en requête par tableaux (via le thm 29 d'équivalence)
② Minimiser le nombres de lignes (via le thm 62 d'homomorphisme)
③ Transformer le résultat en SPC (via le thm 37 de Codd).
④ Optimiser localement les requêtes SPC (via les transformations en prop 57).

Référence : Abiteboul _ Foundations of Databases.
~~Delobel - bases de données et systèmes relationnels.~~

- contraintes d'intégrité.
- requêtes récursives.

⊛ADDENDUM :

thm 50bis (codd 2) :
le calcul relationnel indépendant sous domaine est équivalent à l'algèbre relationnelle.

DEV 0

Annexe 1 : exemple de base de données.

Flims

Titre	Réalisateur	Acteur
whiplash	Damien Chazelle	J.K. Simons
lalaland	Damien Chazelle	Ryan Gosling
lalaland	Damien Chazelle	Emma Stone
The Amazing SpiderMan (TAS)	Marc Webb	Emma Stone
L'île aux chiens	Wes Anderson	
Réalité	Quentin Dupieux	A. Chabat
Au Poste	Quentin Dupieux	Marc Fraize
Didier	A. Chabat	A. Chabat

Lieu

Cinema	Adresse
le Métropole	26 rue des Ponts de Comine
UGC	40 rue de Béthune
le Majestic	54 rue de Béthune
L'hybride	18 rue Gosselet

CinéLille

Cinema	Titre	Heure
le Métropole	TAS	10^h00
le Métropole	TAS	13^h00
le Métropole	TAS	16^h00
le Métropole	TAS	19^h00
le Majestic	Réalité	19^h00

Annexe 2 : requête par tableaux.

T :

Titre	Réalisateur	Acteur
x	Damien Chazelle	z

u : $\langle x \rangle$

Annexe 3 : complexité des opérateurs algébriques de base.

Opération	Complexité	Commentaire
$I_1 \times I_2$	$O(n_1 \times n_2)$	à éviter
$I_1 \bowtie I_2$	$O(n_1 \times n_2)$	naïf
	$O(n_1 \log n_2 + n_2 \log n_1 + \lvert I_1 \bowtie I_2 \rvert)$	par tri, puis fusion
$\pi_{j_1 \dots j_k}(I)$	$\underbrace{O(n \times k)}_{\text{recopie}} + \underbrace{O(n \log n)}_{\text{gestion des doublons}}$	on peut garder les doublons → commande DISTINCT (SQL)

Annexe 4 : exemple d'utilisation du théorème d'homomorphisme.

R	A	B
	x	y
	x	y

$q_0 = (T_0, \langle x, y \rangle)$

R	A	B
	x	y_1
	x_1	y_1
	x_1	y
	x	y

$q_1 = (T_1, \langle x, y \rangle)$

R	A	B
	x	y_1
	x_1	y_1
	x_1	y_2
	x_2	y_2
	x_2	y
	x	y

$q_2 = (T_2, \langle x, y \rangle)$

R	A	B
	x	y_1
	x_1	y
	x	y

$q_3 = (T_3, \langle x, y \rangle)$

Annexe 5 : une paire de requêtes par tableaux équivalentes

R	A	B
	x	x
	x	y_1
	y_1	y_2
	$\vdots$	$\vdots$
	y_{n-1}	y_n
	y_n	x
	x	

$q = (T, \langle x \rangle)$

R	A	B
	x	x
	x	

$q' = (T', \langle x \rangle)$