

Bases de données

L2 sciences et technologies, mention informatique

algèbres conjonctives

ou : comment extraire de l'information de ceci (avec un langage algébrique)?

films	titre	réalisateur	année		réalisateurs	nom	nationalité
	starwars	lucas	1977			lucas	américaine
	nikita	besson	1990			lynch	américaine
	locataires	ki-duk	2005			besson	française
	dune	lynch	1984			ki-duk	coréenne

patrick.marcel@univ-tours.fr

<http://celene.univ-tours.fr/course/view.php?id=3131>

pour les exemples...

soit l'instance de base de données suivante :

films	titre	réalisateur	année
	starwars1	lucas	1977
	nikita	besson	1990
	les locataires	ki-duk	2005
	dune	lynch	1984
	starwars4	lucas	1999
	starwars5	lucas	2002
	l'île	ki-duk	2000
	angela	besson	2005
	eraserhead	lynch	1976
	...		

realisateurs	nom	nationalité
	lucas	américaine
	lynch	américaine
	besson	française
	ki-duk	coréenne
	...	
acteurs	nom	films
	ford	starwars1
	ford	indiana jones
	willis	5eme élément
	...	

langage algébrique

définition d'opérateurs unaires et binaires sur les instances de relations

nous retrouvons les deux approches :

- ▶ approche non nommée

algèbre SPC

- ▶ approche nommée

algèbre SPJR

algèbre SPC

algèbre SPC

les noms d'attributs ne sont pas utilisés dans la définition des opérateurs

S :	Sélection	σ
P :	Projection	π
C :	produit Cartésien	\times

exemple

comment construire les tuples résultats de la requête
“lister les films réalisés par des américains”

4 étapes :

1. sélection des tuples de *réalisateur*s correspondant à des réalisateurs américains
2. combinaison par produit cartésien de ces tuples avec ceux de *films*
3. restriction aux tuples où les positions correspondant aux réalisateurs ont les mêmes valeurs
4. récupération des titres des films

étape 1

$$I_1 := \sigma_{2=\text{"américaine"}}(\text{réalisateurs})$$

$$I_1 = \{(\text{lucas}, \text{américaine}), (\text{lynch}, \text{américaine}), \dots\}$$

étape 2

$$I_2 := I_1 \times \text{films}$$

$$I_2 = \{(\text{lucas}, \text{américaine}, \text{starwars1}, \text{lucas}, 1977), \\ (\text{lucas}, \text{américaine}, \text{dune}, \text{lynch}, 1984), \\ (\text{lucas}, \text{américaine}, \text{nikita}, \text{besson}, 1990), \\ (\text{lynch}, \text{américaine}, \text{dune}, \text{lynch}, 1984), \\ (\text{lynch}, \text{américaine}, \text{starwars1}, \text{lucas}, 1977), \dots \}$$

étape 3

$$I_3 := \sigma_{1=4}(I_2)$$

$I_3 = \{(\text{lucas}, \text{américaine}, \text{starwars1}, \text{lucas}, 1977),$
 $(\text{lynch}, \text{américaine}, \text{dune}, \text{lynch}, 1984),$
 $(\text{lucas}, \text{américaine}, \text{starwars4}, \text{lucas}, 1999),$
 $(\text{lucas}, \text{américaine}, \text{starwars5}, \text{lucas}, 2002),$
 $(\text{lucas}, \text{américaine}, \text{starwars6}, \text{lucas}, 2005)$
 $(\text{lynch}, \text{américaine}, \text{eraserhead}, \text{lynch}, 1976) \dots \}$

étape 4

$$I_4 := \pi_3(I_3)$$

$$I_4 = \{(\text{starwars1}), (\text{dune}), (\text{starwars4}), (\text{starwars5}), (\text{starwars6}) \\ (\text{eraserhead}) \dots\}$$

étape 4

$$I_4 := \pi_3(I_3)$$

$$I_4 = \{(\text{starwars1}), (\text{dune}), (\text{starwars4}), (\text{starwars5}), (\text{starwars6}) \\ (\text{eraserhead}) \dots\}$$

$$\text{soit } I_4 = \pi_3(\sigma_{1=4}(\sigma_{2=\text{"américaine"}}(\text{réalisateurs}) \times \text{films}))$$

sélection

soient $j, k \in \mathbb{N}$ et $a \in \mathbf{dom}$, I une instance de relation, tels que $\max(j, k) \leq \text{arité}(I)$

$$\sigma_{j=a}(I) = \{t \in I \mid t(j) = a\}$$

$$\sigma_{j=k}(I) = \{t \in I \mid t(j) = t(k)\}$$

sélection généralisée

on utilise plutôt la forme généralisée de la sélection :

$$\sigma_{\varphi}$$

où φ est une formule conjonctive de sélection :

- ▶ $\varphi = \gamma_1 \wedge \dots \wedge \gamma_n$
- ▶ et les γ_i sont de la forme $j = a$ ou $j = k$

σ_{φ} est équivalent à $\sigma_{\gamma_1}(\dots(\sigma_{\gamma_n}(I)))$

projection

soient $j_1, \dots, j_n \in \mathbb{N}$ et I une instance de relation, tels que
 $\max(j_1, \dots, j_n) \leq \text{arité}(I)$

$$\pi_{j_1, \dots, j_n}(I) = \{(t(j_1), \dots, t(j_n)) \mid t \in I\}$$

produit cartésien

soient I et J deux instances de relations telles que $\text{arité}(I) = n$ et $\text{arité}(J) = m$

$$I \times J = \{(t(1), \dots, t(n), s(1), \dots, s(m)) \mid t \in I, s \in J\}$$

produit cartésien

soient I et J deux instances de relations telles que $\text{arité}(I) = n$ et $\text{arité}(J) = m$

$$I \times J = \{(t(1), \dots, t(n), s(1), \dots, s(m)) \mid t \in I, s \in J\}$$

opération associative, non commutative, ayant la relation $\{()\}$ comme élément neutre

syntaxe des requêtes SPC

pour une base de données de schéma D , une requête SPC q est :

syntaxe des requêtes SPC

pour une base de données de schéma D , une requête SPC q est :

- ▶ R , si $R \in D$, $\text{arité}(q) = \text{arité}(R)$

syntaxe des requêtes SPC

pour une base de données de schéma D , une requête SPC q est :

- ▶ R , si $R \in D$, $\text{arité}(q) = \text{arité}(R)$
- ▶ $\{(a)\}$ si $a \in \mathbf{dom}$, $\text{arité}(q) = 1$

syntaxe des requêtes SPC

pour une base de données de schéma D , une requête SPC q est :

- ▶ R , si $R \in D$, $\text{arité}(q) = \text{arité}(R)$
- ▶ $\{(a)\}$ si $a \in \mathbf{dom}$, $\text{arité}(q) = 1$
- ▶ $\sigma_\varphi(q')$ si q' est une requête, $\text{arité}(q) = \text{arité}(q')$

syntaxe des requêtes SPC

pour une base de données de schéma D , une requête SPC q est :

- ▶ R , si $R \in D$, $\text{arité}(q) = \text{arité}(R)$
- ▶ $\{(a)\}$ si $a \in \mathbf{dom}$, $\text{arité}(q) = 1$
- ▶ $\sigma_{\varphi}(q')$ si q' est une requête, $\text{arité}(q) = \text{arité}(q')$
- ▶ $\pi_{j_1, \dots, j_n}(q')$ si q' est une requête, $\text{arité}(q) = n$

syntaxe des requêtes SPC

pour une base de données de schéma D , une requête SPC q est :

- ▶ R , si $R \in D$, $\text{arité}(q) = \text{arité}(R)$
- ▶ $\{(a)\}$ si $a \in \mathbf{dom}$, $\text{arité}(q) = 1$
- ▶ $\sigma_{\varphi}(q')$ si q' est une requête, $\text{arité}(q) = \text{arité}(q')$
- ▶ $\pi_{j_1, \dots, j_n}(q')$ si q' est une requête, $\text{arité}(q) = n$
- ▶ $q_1 \times q_2$ si q_1 et q_2 sont des requêtes, $\text{arité}(q) = \text{arité}(q_1) + \text{arité}(q_2)$

syntaxe des requêtes SPC

pour une base de données de schéma D , une requête SPC q est :

- ▶ R , si $R \in D$, $\text{arité}(q) = \text{arité}(R)$
- ▶ $\{(a)\}$ si $a \in \mathbf{dom}$, $\text{arité}(q) = 1$
- ▶ $\sigma_{\varphi}(q')$ si q' est une requête, $\text{arité}(q) = \text{arité}(q')$
- ▶ $\pi_{j_1, \dots, j_n}(q')$ si q' est une requête, $\text{arité}(q) = n$
- ▶ $q_1 \times q_2$ si q_1 et q_2 sont des requêtes, $\text{arité}(q) = \text{arité}(q_1) + \text{arité}(q_2)$

pour une instance I et une requête q , on note $q(I)$ l'image de I par q

exemples

en quelle année est sorti “nikita” ?

$$\pi_3(\sigma_{1=\text{''nikita''}}(films))$$

exemples

en quelle année est sorti “nikita” ?

$$\pi_3(\sigma_{1=\text{''nikita''}}(\textit{films}))$$

quelle est la nationalité du réalisateur de “locataires” ?

$$\pi_5(\sigma_{2=4}(\sigma_{1=\text{''locataires''}}(\textit{films}) \times \textit{réalisateurs}))$$

remarques

certaines requêtes SPC ne sont pas satisfiables...

remarques

certaines requêtes SPC ne sont pas satisfiables...

$\sigma_{1=a}(\sigma_{1=b}(I))$ avec $\text{arité}(I) \geq 1$ et $a \neq b$

intersection

considérons l'opération d'intersection de 2 instances : \cap

soient I et J deux instances de relations de même arité

$$I \cap J = \{t \mid t \in I \text{ et } t \in J\}$$

\cap peut-être simulée par les opérateurs σ, π, \times

equi-jointure

pour deux instances I et J , l'équi-jointure \bowtie_{φ} est définie comme suit :

soit une formule $\varphi = (j_1 = k_1) \wedge \dots \wedge (j_n = k_n)$ telle que $j_i \in [1, \text{arité}(I)]$ et $k_i \in [1, \text{arité}(J)]$

$$I \bowtie_{\varphi} J = \sigma_{\varphi'}(I \times J)$$

avec $\varphi' = (j_1 = k_1 + \text{arité}(I)) \wedge \dots \wedge (j_n = k_n + \text{arité}(I))$

exemple

la requête “lister les films réalisés par des américains”

peut s'écrire :

$$\pi_3(\sigma_{2=\text{''américaine''}}(\text{réalisateurs} \bowtie_{1=2} \text{films}))$$

forme normale

toute requête SPC peut être mise sous la forme :

$$\pi_{j_1, \dots, j_n}(\{(a_1)\} \times \dots \times \{(a_m)\} \times \sigma_{\varphi}(R_1 \times \dots \times R_k))$$

où

- ▶ $a_1, \dots, a_m \in \mathbf{dom}$,
- ▶ les R_i sont des noms de relation,
- ▶ φ est une formule conjonctive de sélection

règles de réécriture

les règles suivantes

- ▶ illustrent les propriétés des opérateurs
- ▶ peuvent être appliquées à toute requête SPC
- ▶ permettent d'obtenir une requête équivalente

fusion

merge select

$$\sigma_F(\sigma_{F'}(q)) \rightarrow \sigma_{F \wedge F'}(q)$$

fusion

merge select

$$\sigma_F(\sigma_{F'}(q)) \rightarrow \sigma_{F \wedge F'}(q)$$

merge project

$$\pi_{\vec{j}}(\pi_{\vec{k}}(q)) \rightarrow \pi_{\vec{l}}(q)$$

avec $l_i = k_{j_i}$ pour tout l_i dans \vec{l}

produit cartésien

associativité

$$((q_1 \times \dots \times q_n) \times q) \rightarrow (q_1 \times \dots \times q_n \times q)$$

produit cartésien

associativité

$$((q_1 \times \dots \times q_n) \times q) \rightarrow (q_1 \times \dots \times q_n \times q)$$

commutativité

$$(q \times q') \rightarrow \pi_{\vec{j}\vec{j'}}(q' \times q)$$

avec $\vec{j} = \text{arité}(q') + 1, \dots, \text{arité}(q') + \text{arité}(q)$ et $\vec{j'} = 1, \dots, \text{arité}(q')$

sélection

push select through project

$$\sigma_F(\pi_{\vec{j}}(q)) \rightarrow \pi_{\vec{j}}(\sigma_{F'}(q))$$

où F' est obtenu en remplaçant dans F les coordonnées i par j_i

sélection

push select through project

$$\sigma_F(\pi_{\vec{j}}(q)) \rightarrow \pi_{\vec{j}}(\sigma_{F'}(q))$$

où F' est obtenu en remplaçant dans F les coordonnées i par j_i

push select through singleton

$$\sigma_{1=j}(\{(a)\} \times q) \rightarrow \{(a)\} \times \sigma_{j-1=a}(q)$$

produit cartésien et sélection

push cross through select

$$(\sigma_F(q) \times q') \rightarrow \sigma_F(q \times q')$$

produit cartésien et sélection

push cross through select

$$(\sigma_F(q) \times q') \rightarrow \sigma_F(q \times q')$$

$$(q \times \sigma_F(q')) \rightarrow \sigma_{F'}(q \times q')$$

où F' est obtenu en remplaçant dans F les coordonnées i par $i + \text{arité}(q)$

produit cartésien et projection

push cross through project

$$(\pi_{\vec{j}}(q) \times q') \rightarrow \pi_{\vec{j}\vec{j}'}(q \times q')$$

où $\vec{j}' = \text{arité}(q) + 1, \dots, \text{arité}(q) + \text{arité}(q')$

produit cartésien et projection

push cross through project

$$(\pi_{\vec{j}}(q) \times q') \rightarrow \pi_{\vec{j}\vec{j}'}(q \times q')$$

où $\vec{j}' = \text{arité}(q) + 1, \dots, \text{arité}(q) + \text{arité}(q')$

$$(q \times \pi_{\vec{j}}(q')) \rightarrow \pi_{1, \dots, \text{arité}(q), \vec{j}'}(q \times q')$$

où \vec{j}' est obtenu en remplaçant dans \vec{j} les coordonnées i par $i + \text{arité}(q)$

exemple

la requête

$$q = \pi_3(\sigma_{1=4}(\sigma_{2=\text{"américaine"}}(\text{réalisateurs}) \times \text{films}))$$

peut être réécrite en

$$q = \pi_2(\sigma_{2=4 \wedge 5=\text{"américaine"}}(\text{films} \times \text{réalisateurs}))$$

algèbre SPJR

algèbre SPJR

les noms d'attributs sont utilisés dans la définition des opérateurs

S :	Sélection	σ
P :	Projection	π
J :	Jointure naturelle	\bowtie
R :	Renommage	ρ

exemple

la requête “lister les films réalisés par des américains”

peut s'écrire

$$\pi_{\text{titre}}(\sigma_{\text{nationalité} = \text{'américaine'}}(\text{réalisateurs}) \bowtie \rho_{\text{réalisateur} \rightarrow \text{nom}}(\text{films}))$$

sélection

soit $c \in \mathbf{dom}$, I une instance avec $A, B \in \text{sorte}(I)$

$$\begin{aligned}\sigma_{A=c}(I) &= \{t \in I \mid t(A) = c\} \\ \sigma_{A=B}(I) &= \{t \in I \mid t(A) = t(B)\}\end{aligned}$$

sélection

soit $c \in \mathbf{dom}$, I une instance avec $A, B \in \text{sorte}(I)$

$$\begin{aligned}\sigma_{A=c}(I) &= \{t \in I \mid t(A) = c\} \\ \sigma_{A=B}(I) &= \{t \in I \mid t(A) = t(B)\}\end{aligned}$$

cette sélection admet une forme généralisée σ_φ où φ est une formule conjonctive de sélection

projection

soient une instance I et $A_1, \dots, A_n \in \text{sorte}(I)$

$$\pi_{A_1, \dots, A_n}(I) = \{(A_1 : t(A_1), \dots, A_n : t(A_n)) \mid t \in I\}$$

plus simplement : $\pi_{A_1, \dots, A_n}(I) = \{t \mid \{A_1, \dots, A_n\} \subseteq t \mid t \in I\}$

jointure naturelle

soient I et J deux instances

$$I \bowtie J = \{t \text{ sur } \text{sorte}(I) \cup \text{sorte}(J) \mid \\ \exists v \in I \text{ et } w \in J, t|_{\text{sorte}(I)} = v \\ \text{et } t|_{\text{sorte}(J)} = w\}$$

jointure naturelle

soient I et J deux instances

$$I \bowtie J = \{t \text{ sur } \text{sorte}(I) \cup \text{sorte}(J) \mid \\ \exists v \in I \text{ et } w \in J, t|_{\text{sorte}(I)} = v \\ \text{et } t|_{\text{sorte}(J)} = w\}$$

opération associative, commutative ayant la relation $\{()\}$ comme élément neutre

exemple de jointure naturelle

soient les instances de relations :

<i>I</i>	A	B
	1	2
	4	2

<i>J</i>	B	C
	2	3
	2	5

<i>K</i>	A	B
	1	2
	2	3

<i>L</i>	A	D
	1	2
	3	4

exemple de jointure naturelle

soient les instances de relations :

$$I \begin{array}{c|cc} A & B \\ \hline 1 & 2 \\ 4 & 2 \end{array}$$

$$J \begin{array}{c|cc} B & C \\ \hline 2 & 3 \\ 2 & 5 \end{array}$$

$$K \begin{array}{c|cc} A & B \\ \hline 1 & 2 \\ 2 & 3 \end{array}$$

$$L \begin{array}{c|cc} A & D \\ \hline 1 & 2 \\ 3 & 4 \end{array}$$

$$I \bowtie J \begin{array}{c|ccc} A & B & C \\ \hline 1 & 2 & 3 \\ 1 & 2 & 5 \\ 4 & 2 & 3 \\ 4 & 2 & 5 \end{array}$$

$$I \bowtie K \begin{array}{c|cc} A & B \\ \hline 1 & 2 \end{array}$$

$$J \bowtie L \begin{array}{c|cccc} A & B & C & D \\ \hline 1 & 2 & 3 & 2 \\ 3 & 2 & 3 & 4 \\ 1 & 2 & 5 & 2 \\ 3 & 2 & 5 & 4 \end{array}$$

fonction de renommage

U un ensemble d'attributs

un renommage des attributs de U est une fonction f

- ▶ de U dans **att**
- ▶ s'écrit $A_1, \dots, A_n \rightarrow B_1, \dots, B_n$
- ▶ $f(A_i) = B_i$ pour $i \in [1, n]$

opération de renommage

soit I une instance, f une fonction de renommage de $sorte(I)$ dans **att**

$$\rho_f(I) = \{t \text{ sur } f[sorte(I)] \mid \text{pour } u \in I, \\ t(f(A)) = u(A) \text{ pour tout } A \in sorte(I)\}$$

syntaxe et forme normale

la syntaxe est définie de manière analogue à celle des requêtes SPC

syntaxe et forme normale

la syntaxe est définie de manière analogue à celle des requêtes SPC

toute requête SPJR peut être mise sous la forme :

$$\pi_{B_1, \dots, B_n}(\{(A_1 : a_1)\} \bowtie \dots \bowtie \{(A_m : a_m)\} \bowtie \sigma_{\varphi}(\rho_{f_1}(R_1) \bowtie \dots \bowtie \rho_{f_k}(R_k)))$$

forme normale

dans cette forme normale :

$$\pi_{B_1, \dots, B_n}(\{(A_1 : a_1)\} \bowtie \dots \bowtie \{(A_m : a_m)\} \bowtie \sigma_{\varphi}(\rho_{f_1}(R_1) \bowtie \dots \bowtie \rho_{f_k}(R_k)))$$

- ▶ $a_1, \dots, a_m \in \mathbf{dom}$,
- ▶ les R_i sont des noms de relation,
- ▶ φ est une formule conjonctive de sélection,
- ▶ les A_i sont distincts et apparaissent dans les B_j
- ▶ les f_j sont des renommages sur $sorte(R_j)$
- ▶ les A_i n'apparaissent pas parmi les $\rho_{f_j}(R_j)$
- ▶ les sortes des $\rho_{f_j}(R_j)$ sont disjointes deux à deux

équivalences

théorème :

l'algèbre SPC et l'algèbre SPJR sont équivalentes

q est une requête SPC ssi q est une requête SPJR

équivalences

il faut démontrer que :

1. toute requête SPC peut s'écrire comme une requête SPJR
2. toute requête SPJR peut s'écrire comme une requête SPC

ébauche de démonstration pour 1

toute requête SPC q peut être écrite sous forme normale

$$q = \pi_{j_1, \dots, j_n}(\{(a_1)\} \times \dots \times \{(a_m)\} \times \sigma_{\varphi}(R_1 \times \dots \times R_k))$$

écrivons une requête SPJR q' qui est équivalente à q

comme toute requête SPJR, q' peut être écrite sous forme normale

$$q' = \pi_{B_1, \dots, B_n}(\{(A_1 : a_1)\} \bowtie \dots \bowtie \{(A_m : a_m)\} \bowtie \sigma_{\varphi'}(\rho_{f_1}(R_1) \bowtie \dots \bowtie \rho_{f_k}(R_k)))$$

démonstration

quel est le problème?

on a

$(R_1 \times \dots \times R_k)$ à exprimer par $\rho_{f_1}(R_1) \bowtie \dots \bowtie \rho_{f_k}(R_k)$

et $\sigma_{\varphi}(R_1 \times \dots \times R_k)$ à exprimer par $\sigma_{\varphi'}(\rho_{f_1}(R_1) \bowtie \dots \bowtie \rho_{f_k}(R_k))$

avec φ une formule sur les positions de $(R_1 \times \dots \times R_k)$

démonstration

pour $t \in [0, k]$, on définit $\beta(t) = m + \sum_{s=1}^t \text{arité}(R_s)$

soient $A_{m+1}, \dots, A_{\beta(k)}$ de nouveaux attributs

pour $t \in [1, k]$, on définit les fonctions de renommage f_t telle que pour R_t , $f_t(B_i) = A_{\beta(t-1)+i}$ si B_i est le i ème attribut de R_t

on traduit $(R_1 \times \dots \times R_k)$ par $\rho_{f_1}(R_1) \bowtie \dots \bowtie \rho_{f_k}(R_k)$

vérification sur un exemple (1)

soit $R_1[A,B,C]$ et $R_2[C,D]$ et A_1, \dots, A_5 des nouveaux attributs

soit la requête $q = \sigma_{3=4}(R_1 \times R_2)$

R_1 est d'arité 3, R_2 d'arité 2 donc $\beta(0) = 0, \beta(1) = 3, \beta(2) = 5$

$f_1(A) = A_1, f_1(B) = A_2, f_1(C) = A_3, f_2(C) = A_4, f_2(D) = A_5$

on traduit $(R_1 \times R_2)$ par $\rho_{f_1}(R_1) \bowtie \rho_{f_2}(R_2)$

démonstration

soit la fonction γ de $[1, \beta(k)] \cup \mathbf{dom}$ vers $\{A_{m+1}, \dots, A_{\beta(k)}\} \cup \mathbf{dom}$

définie par

- ▶ $\gamma(j) = A_{m+j}$ si $j \in [1, \beta(k)]$
- ▶ $\gamma(a) = a$ si $a \in \mathbf{dom}$
- ▶ $\gamma(\alpha_1 = \alpha_2) = (\gamma(\alpha_1) = \gamma(\alpha_2))$
- ▶ $\gamma(\alpha_1 \wedge \alpha_2) = \gamma(\alpha_1) \wedge \gamma(\alpha_2)$

alors on remplace σ_φ par $\sigma_{\gamma(\varphi)}$

vérification sur un exemple (2)

pour $q = \sigma_{3=4}(R_1 \times R_2)$,

on a traduit $(R_1 \times R_2)$ par $\rho_{f_1}(R_1) \bowtie \rho_{f_2}(R_2)$

on définit γ telle que

$$\gamma(1) = A_1, \gamma(2) = A_2, \gamma(3) = A_3, \gamma(4) = A_4, \gamma(5) = A_5$$

on traduit $\sigma_{3=4}$ par $\sigma_{A_3=A_4}$

démonstration

il ne reste plus que la projection... triviale !

Quod Erat Demonstrandum

équivalences

les langages suivants, dans le cas des requêtes conjonctives satisfiables, sont équivalents:

1. l'algèbre SPC
2. l'algèbre SPJR
3. langage des règles conjonctives
4. le calcul conjonctif

ébauche de démonstration

on savait déjà que le calcul et le langage de règles l'étaient

ébauche de démonstration

on savait déjà que le calcul et le langage de règles l'étaient

montrons que toute règle

$$\text{résultat}(\vec{x}) \leftarrow R_1(\vec{x}_1), \dots, R_k(\vec{x}_k)$$

peut s'écrire

$$\pi_{j_1, \dots, j_n}(\{(a_1)\} \times \dots \times \{(a_m)\} \times \sigma_\varphi(R_1 \times \dots \times R_k))$$

ébauche de démonstration

il suffit de

1. faire le produit cartésien de $R_1 \times \dots \times R_k$
2. faire la sélection reflétant
 - ▶ les constantes apparaissant parmi $\vec{x}_1, \dots, \vec{x}_k$
 - ▶ les variables répétées dans $\vec{x}_1, \dots, \vec{x}_k$
3. faire un produit cartésien avec les constantes apparaissant dans \vec{x}
4. faire la projection sur les coordonnées correspondant à \vec{x}

vérification sur un exemple

$$\text{résultat}(x,y,1) \leftarrow R_1(x,y), R_2(y,1,z)$$

1. le produit cartésien : $R_1 \times R_2$
2. la sélection : $\sigma_{2=3 \wedge 4="1"}(R_1 \times R_2)$
3. le produit cartésien avec les constantes :
 $\{"1"\} \times \sigma_{2=3 \wedge 4="1"}(R_1 \times R_2)$
4. la projection : $\pi_{2,3,1}(\{"1"\} \times \sigma_{2=3 \wedge 4="1"}(R_1 \times R_2))$