

ALAIN BILLIONNET

**Implémentation optimale d'une relation
dans une base de données relationnelle par
un système de projecteurs**

*Revue française d'automatique, d'informatique et de recherche
opérationnelle. Recherche opérationnelle*, tome 16, n° 4 (1982),
p. 349-363.

http://www.numdam.org/item?id=RO_1982__16_4_349_0

© AFCET, 1982, tous droits réservés.

L'accès aux archives de la revue « Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/legal.php>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

IMPLÉMENTATION OPTIMALE D'UNE RELATION DANS UNE BASE DE DONNÉES RELATIONNELLE PAR UN SYSTÈME DE PROJECTEURS (*)

par Alain BILLIONNET (¹)

Résumé. — Dans l'implémentation logique d'une base de données relationnelle des chemins permettent d'accéder, à partir des n-uplets d'une relation, aux valeurs de certains de leurs attributs, stockées dans d'autres relations, ce qui autorise une structure non redondante des données. De tels chemins d'accès sont appelés projecteurs. Pour une relation, cet article propose un algorithme visant à construire un ensemble de projecteurs qui assure la non redondance des données et qui est optimal vis-à-vis des deux critères : encombrement en mémoire et temps d'accès. Cet algorithme est fondé, d'une part, sur le principe de la programmation dynamique et, d'autre part, sur la résolution de problèmes de partitionnement.

Mots clés : Base de données; espace mémoire; temps d'accès; graphe; optimisation.

Abstract. — The access paths in a logical implementation of a relational data base allow to access from n-uples of relations to values of some of their attributes stored in other relations in order to obtain a non-redundant structure of data. Such access paths are called projectors. This paper gives for a relation an algorithm to construct a set of projectors which ensures the non-redundancy and is optimal towards space in memory and access times. This algorithm uses the dynamic programming principle and an algorithm for the set partitioning problem.

Keywords: Data base; storage requirement; access time; graph; optimization.

1. INTRODUCTION

La conception des bases de données est, depuis plusieurs années, un sujet du plus grand intérêt pour les organisations. Elle a conduit à développer différents modèles utilisés dans l'ordre suivant :

- le modèle conceptuel, de type relationnel [6], qui traduit par des données, de manière synthétique, la représentation du réel;
- le modèle logique ou modèle d'accès [5, 7, 11], qui consiste à choisir les structures d'accès logiques en tenant compte de tous les usages que les utilisateurs veulent faire des données;

(*) Reçu juillet 1981.

(¹) Institut d'Informatique d'Entreprise, C.N.A.M., 292, rue Saint-Martin, 75141 Paris Cedex 03.

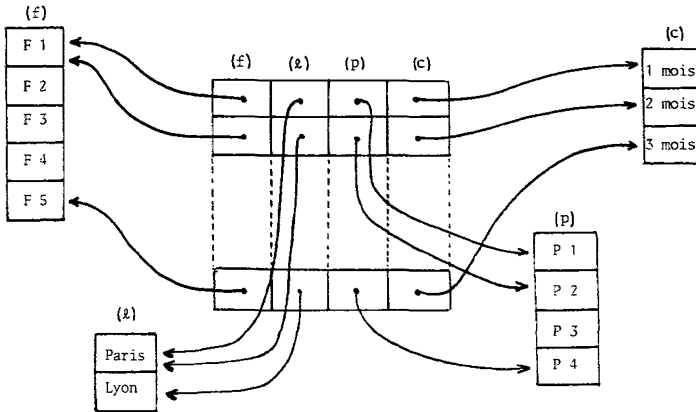


Figure 1

– le modèle physique qui définit la réalisation technique de la base en tenant compte des performances.

Le but de cet article est d'étudier la façon d'accéder, à partir d'un n -uplet d'une relation, à la valeur des attributs de ce n -uplet par l'intermédiaire de projecteurs. Un projecteur d'une relation $\mathcal{R}(B, C)$ vers une de ses projections $\mathcal{R}[B]$ est une fonction d'accès qui associe à chaque n -uplet $B^i C^i$ de BC le n -uplet B^i de $\mathcal{R}[B]$. Ce type de pointeurs permet, d'une part, une implantation non redondante des données (il est toujours possible de stocker une seule occurrence de chaque valeur d'attribut à l'aide de pointeurs comme sur la figure 1) ce qui peut faciliter leur mise à jour et, d'autre part, une économie

TABLEAU

Fournisseur (f)	Localité (l)	Produit (p)	Conditions de paiement (c)
F 1	Paris	P 1	1 mois
F 1	Paris	P 2	2 mois
F 1	Paris	P 3	1 mois
F 1	Paris	P 4	3 mois
F 2	Lyon	P 1	1 mois
F 2	Lyon	P 2	2 mois
F 2	Lyon	P 3	1 mois
F 2	Lyon	P 4	1 mois
F 3	Lyon	P 2	2 mois
F 3	Lyon	P 3	2 mois
F 4	Paris	P 1	1 mois
F 4	Paris	P 2	2 mois
F 5	Lyon	P 3	1 mois
F 5	Lyon	P 4	3 mois

d'espace pour la mémorisation de la relation si l'espace occupé par un pointeur est petit devant l'espace occupé par une donnée.

Exemple : Considérons la relation \mathcal{R} (fournisseur, localité, produit, conditions de paiement) décrite dans le tableau.

On peut implanter ces données de la manière non redondante [8], présentée sur la figure 1, que l'on schématisera par le graphe de projection de la figure 2.

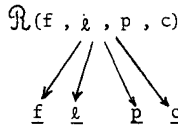


Figure 2

Évaluons, dans le cas général, l'économie d'espace obtenue grâce à cette implantation. Notons m l'espace occupé par un pointeur et M l'espace occupé par une donnée. Calculons, pour une relation $\mathcal{R}(A) = \mathcal{R}(a_1, \dots, a_p)$, le rapport ρ_A entre l'espace total occupé par la solution non redondante triviale et celui qu'occupe la relation stockée sans l'intermédiaire de projecteurs :

$$\rho_A = \frac{[A] \cdot p \cdot m + \sum_{i=1}^p [a_i] \cdot M}{[A] \cdot p \cdot M} = \frac{m}{M} + \frac{\sum_{i=1}^p [a_i]}{[A] \cdot p},$$

où $[A]$ désigne le nombre de n -uplets de la relation $\mathcal{R}[A]$ et $[a_i]$ le nombre de valeurs de l'attribut a_i . De façon générale, nous noterons $[E]$ le nombre de n -uplets de la relation $\mathcal{R}[E]$, pour tout ensemble d'attributs E inclus dans A .

Remarquons que si l'espace occupé par une donnée est très important par rapport à celui occupé par un pointeur, alors ρ_A tend vers :

$$\rho_A \simeq \frac{\sum_{i=1}^p [a_i]}{[A] \cdot p}.$$

Le problème que nous considérons ici est la détermination, pour une relation $\mathcal{R}(A)$, d'un système de projecteurs qui assure la *non redondance* des données et soit « optimal » vis-à-vis des deux critères : *nombre total de pointeurs* et *temps d'accès*. Considérons de nouveau l'exemple précédent. La figure 3 représente une amélioration de la solution triviale de la figure 1, en ce qui concerne le nombre de pointeurs; en revanche, le temps d'accès aux valeurs de l'attribut l est augmenté. La figure 4 représente le graphe de projection correspondant.

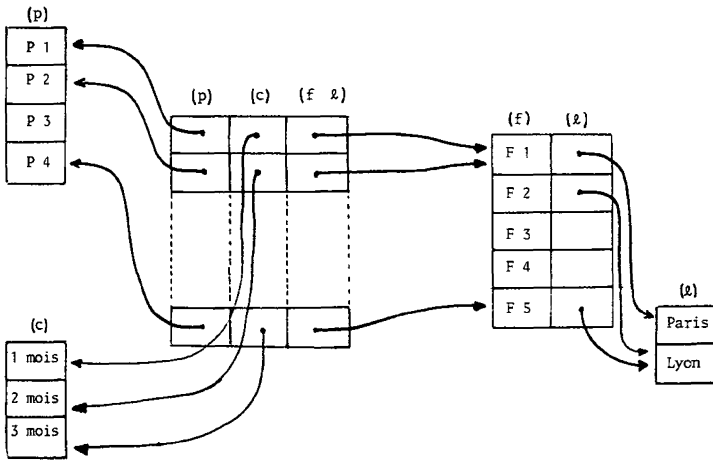


Figure 3

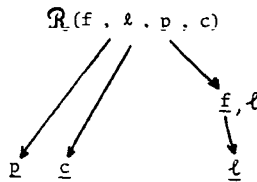


Figure 4. — Le fait que l'attribut f soit souligné au sommet f, l signifie que la valeur de cet attribut est stockée en ce sommet.

Dans certains cas, l'économie en nombre de pointeurs réalisée par un système de projecteurs à plusieurs niveaux peut être très appréciable. En effet, évaluons l'économie réalisée dans l'exemple (très favorable) suivant :

Soit $\mathcal{R}(A) = \mathcal{R}(a_1, \dots, a_p)$ une relation particulière telle que :

- $p = 2^k$ ($k \geq 1$);
- $[A] = 2^p$;
- il existe deux sous-ensembles de A , A_1 et A_2 , ayant le même nombre d'attributs, formant une partition de A et tels que :

$$[A_1] = [A_2] = 2^{(2^k - 1)};$$

- il existe deux sous-ensembles de A_1 , A_3 et A_4 , ayant le même nombre d'attributs, formant une partition de A_1 et tels que :

$$[A_3] = [A_4] = 2^{(2^k - 2)};$$

— il existe deux sous-ensembles de A_2 , A_5 et A_6 , ayant le même nombre d'attributs, formant une partition de A_2 et tels que :

— $[A_5]=[A_6]=2^{(2^{k-2})}$, etc.

(Remarquons qu'il est possible de construire une telle relation.)

Stockons cette relation à l'aide du système de projecteurs représenté par le graphe de projection (arborescence dichotomique) de la figure 5.

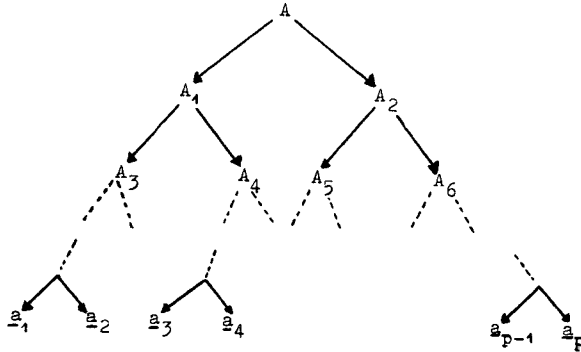


Figure 5

Ce système de projecteurs utilise un nombre de pointeurs :

$$N_1 = 2^1 2^{(2^k)} + 2^2 2^{(2^{k-1})} + 2^3 2^{(2^{k-2})} + \dots + 2^k 2^{(2^1)};$$

La représentation par un système de projecteurs à 1 niveau utilise $N_2 = 2^k 2^{(2^k)}$ pointeurs. Remarquons que $\lim_{k \rightarrow \infty} N_1/N_2 = 0$. En effet, chaque terme de N_1 étant inférieur ou égal à $2^1 2^{(2^k)}$, on a :

$$\frac{N_1}{N_2} \leq \frac{k 2 \cdot 2^{(2^k)}}{2^k 2^{(2^k)}} = \frac{2 \cdot k}{2^k} \quad \text{et} \quad \lim_{k \rightarrow \infty} \frac{k}{2^k} = 0.$$

Pour $k = 3$, le rapport N_1/N_2 est déjà égal à 0,3 environ.

2. DÉFINITION D'UN GRAPHE DE PROJECTION

Considérons la relation $\mathcal{R}(A) = \mathcal{R}(a_1, \dots, a_p)$.

Étant donné un sous-ensemble x de A , nous noterons $K(x)$ l'ensemble des attributs simples clés de la relation $\mathcal{R}[x]$ (on dit que l'ensemble d'attributs

$A_i \subseteq x$ est clé de $\mathcal{R}[x]$ — ou simplement de x — si et seulement si à toute valeur de A_i est associée une et une seule valeur de x .

Un graphe de projection $H=(X, U)$, associé à $\mathcal{R}(A)$, est alors défini de la façon suivante :

- (a) $X \subseteq \mathcal{P}(A)$;
- (b) $A \in X$;
- (c) $\forall i \in \{1, \dots, p\}$, un sommet dont a_i est clé appartient à X ;
- (d) $\forall (x_i, x_j) \in U$, une clé de x_j est incluse dans x_i ;
- (e) $\forall x \in X$ et $\forall a_i \in x$, il existe un chemin, de x au sommet où est stockée la valeur de l'attribut a_i , dont tous les arcs (x_i, x_j) vérifient : $a_i \in x_i \cap x_j$.

Exemple : La figure 6 représente un graphe de projection possible pour la relation $\mathcal{R}(a_1, a_2, a_3, a_4, a_5; a_1, a_5)$.

$\mathcal{R}(A; B)$ signifie que B est une clé de $\mathcal{R}(A)$.

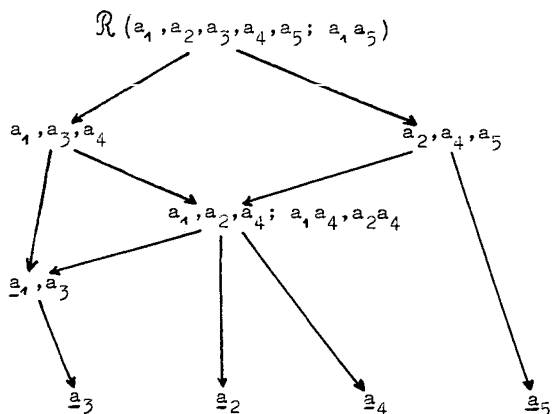


Figure 6

REMARQUES : à chaque sommet de H est associé un ensemble d'attributs appartenant à $\mathcal{P}(A)$. Ce sommet représente la projection de $\mathcal{R}(A)$ sur cet ensemble d'attributs. Nous ne distinguerons pas, dans cet article, un sommet de H , l'ensemble d'attributs associé et la projection correspondante;

- l'arc (x_i, x_j) représente un projecteur qui permet d'accéder aux valeurs de chaque attribut de $x_i \cap x_j$ par l'intermédiaire du sommet x_j ;
- étant donné un projecteur (x_i, x_j) , la condition d précédente assure que le n -uplet de x_j associé à un n -uplet de x_i est unique;
- la valeur de chaque attribut a_i est stockée en un des sommets dont a_i est clé (éventuellement le sommet a_i) ce qui permet de garantir la non redondance des données.

3. LA FONCTION ÉCONOMIQUE

Étant donné un graphe de projection $H = (X, U)$ associé à $\mathcal{R}(A)$, on désire prendre en compte dans l'évaluation de son coût, d'une part, la place en mémoire nécessaire à la réalisation du système de projecteurs et, d'autre part, les temps d'accès aux valeurs des différents attributs.

L'espace de la mémoire utilisé pour implanter le projecteur (x_i, y_j) est proportionnel au nombre de pointeurs nécessaires à cette implantation, qui est lui-même égal au nombre de n -uplets associés à x_i (noté $[x_i]$).

Le temps d'accès aux valeurs de l'attribut a_i est proportionnel à la longueur l_i du plus court chemin de A au sommet où sont stockées les valeurs de a_i . On suppose connue, pour chaque attribut a_i , sa fréquence d'accès f_i .

Nous prendrons donc, comme coût d'un graphe de projection $H = (X, U)$:

$$c(H) = \alpha \sum_{(x_i, x_j) \in U} [x_i] + (1 - \alpha) \sum_{i=1}^p f_i l_i,$$

où α est un coefficient compris entre 0 et 1 visant à pondérer les deux critères : encombrement en mémoire et temps d'accès.

4. PROPRIÉTÉS D'UN GRAPHE DE PROJECTION OPTIMAL

PROPRIÉTÉ 1 : *Tout graphe de projection optimal, H^* , associé à $\mathcal{R}(A)$ est sans circuit.*

Démonstration : Soit x et y deux sommets consécutifs d'un circuit de H^* .

D'après la définition des arcs d'un graphe de projection, x et y sont fonctionnellement équivalents (c'est-à-dire que x et y sont clés de $x \cup y$). Construisons un nouveau graphe de projection H' de la façon suivante :

(a) créons le sommet $z = x \cup y$;

(b) ajoutons un arc de chaque prédécesseur de x , différent de y , vers z et de chaque prédécesseur de y , différent de x , vers z ;

(c) ajoutons un arc de z vers chaque successeur de y , différent de x ;

(d) supprimons les sommets x et y ainsi que leurs arcs incidents.

Le coût de H' est strictement inférieur à celui de H^* puisque $[x \cup y] = [x] = [y]$ d'où la contradiction. ■

PROPRIÉTÉ 2 : *Il existe un graphe de projection optimal, H^* , associé à $\mathcal{R}(A)$ tel que $\forall (x, y) \in H^*; x \cap y = y$.*

Démonstration : Soit (x, y) un arc de H^* tel que : $x \cap y \neq y$. Remplaçons, dans H^* , x par $x \cup y$. Puisque, par définition, une clé de y est incluse dans x , un ensemble d'attributs clé de x est aussi clé de $x \cup y$. Le graphe obtenu est donc toujours un graphe de projection et puisque $[x \cup y] = [x]$, son coût est égal à celui de H^* (on remarquera que la transformation effectuée sur les sommets ne modifie pas les chemins d'accès). ■

PROPRIÉTÉ 3 : *Il existe un graphe de projection optimal associé à $\mathcal{R}(A)$ qui est une arborescence.*

Démonstration : Soit H^* un graphe de projection optimal tel que :

- (a) tout arc (x, y) de H^* vérifie : $x \cap y = y$;
- (b) H^* n'est pas une arborescence.

Il existe alors un sommet x de H^* de demi-degré intérieur supérieur à 1 et dont tous les ascendants, sauf A , ont un demi-degré intérieur égal à 1 (fig. 7).

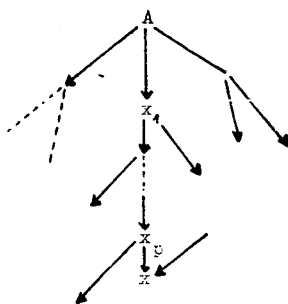


Figure 7

Modifions H^* de la façon suivante : pour tous les chemins $\mu(A, x_1, \dots, x_p, x)$ sauf le plus court, remplaçons les sommets x_1, \dots, x_p par x'_1, \dots, x'_p tels que :

$$\begin{aligned}
 x'_p &= x_p - x \cup y, & y &\in \Gamma_{H^*}(x_p), & y &\neq x, \\
 x'_{p-1} &= x_{p-1} - (x_p - x'_p) \cup y, & y &\in \Gamma_{H^*}(x_{p-1}), & y &\neq x_p, \\
 \hline
 x'_1 &= x_1 - (x_2 - x'_2) \cup y, & y &\in \Gamma_{H^*}(x_1), & y &\neq x_2.
 \end{aligned}$$

Puis supprimons l'arc (x_p, x) . ($\Gamma_{H^*}(x)$ désigne l'ensemble des successeurs de x dans le graphe H^*).

Ces modifications opérées sur H^* conduisent à un nouveau graphe de projection dont le coût est inférieur à celui de H^* . En effet, pour tout $i \in \{1, \dots, p\}$, $[x'_i] \subseteq [x_i]$; d'autre part, certains arcs de H^* sont supprimés.

Tout graphe de projection admet une racine A . Nous venons de montrer qu'il existe un graphe de projection optimal dont tous les sommets, sauf A , sont de demi-degré intérieur égal à 1. Il existe donc un graphe de projection optimal qui est une arborescence. ■

Dans la suite de l'article, nous nous intéressons seulement aux graphes de projection qui sont des arborescences et qui vérifient en outre : pour tout arc $(x, y) : y \subset x$. Nous les appellerons *arborescences de projection*.

PROPRIÉTÉ 4 : *Il existe une arborescence de projection optimale H^* telle que : pour tout x appartenant à H^* les valeurs des attributs de $K(x)$ sont stockées au sommet x .*

Démonstration : Puisque H^* est une arborescence et du fait de la non redondance des données, tous les sommets de H^* dont un attribut a est clé sont sur un même chemin μ de H^* . Les valeurs de a sont stockées au sommet extrémité terminale de μ .

Soit x l'extrémité initiale de ce chemin; modifions H^* en stockant les valeurs de a au sommet x et en retirant l'attribut a de tous les descendants de x . On obtient un graphe de projection de coût inférieur ou égal à celui de H^* . En effectuant cette modification pour tous les attributs de A on obtient une arborescence de projection qui vérifie la propriété. ■

PROPRIÉTÉ 5 : *Il existe une arborescence de projection optimale H^* associée à $\mathcal{R}(A)$ telle que : $\forall x (\neq a_1, \dots, a_p) \in H^*, \Gamma_{H^*}(x)$ est une partition de $x - K(x)$.*

Démonstration : Soit H^* une arborescence de projection optimale telle que : $\forall x \in H^*$, les valeurs des attributs de $K(x)$ sont stockées en x . D'après la définition d'une arborescence de projection :

$$x - K(x) = \bigcup_{y \in \Gamma_{H^*}(x)} y.$$

Supposons que y_1 et y_2 soient deux successeurs de x tels que :

$$[y_1 - K(y_1)] \cap [y_2 - K(y_2)] \neq \emptyset.$$

D'après la définition d'une arborescence de projection, il existe un chemin de y_1 vers chaque sommet stockant les valeurs d'un attribut de $y_1 - K(y_1)$ et de y_2 vers chaque sommet stockant les valeurs d'un attribut de $y_2 - K(y_2)$. Le sommet de H^* stockant les valeurs d'un attribut de :

$$[y_1 - K(y_1)] \cap [y_2 - K(y_2)]$$

est donc de demi-degré intérieur supérieur à 1. Cela contredit l'hypothèse : H^* est une arborescence.

$\{y | y \in \Gamma_{H^*}(x)\}$ est donc bien une partition de $x - K(x)$, puisque $K(y_1) \cap K(y_2) = \emptyset$ d'après l'hypothèse de non redondance des données. ■

Les propriétés précédentes nous permettent de limiter la recherche d'un graphe de projection optimal à un graphe $H = (X, U)$ tel que :

- (a) $X \subseteq \mathcal{P}(A)$;
- (b) H est une arborescence de racine A ;
- (c) $\forall x \in X, \Gamma_H(x)$ est une partition de $x - K(x)$.

Nous pouvons maintenant énoncer le théorème 1 qui définit un graphe porteur d'au moins une arborescence de projection optimale.

5. GRAPHE PORTEUR D'AU MOINS UNE ARBORESCENCE DE PROJECTION OPTIMALE

THÉORÈME 1 : Soit $G = (S, V)$ tel que :

$$S = \mathcal{P}(A) - \emptyset;$$

$$\forall x, y \in S, (x, y) \in V \Leftrightarrow y \subset x - K(x),$$

alors, G admet un sous-graphe partiel qui est une arborescence de projection optimale associée à $\mathcal{R}(A)$.

La démonstration est immédiate d'après les propriétés précédentes. ■

Nous pouvons redéfinir le coût d'une arborescence de projection $H = (X, U)$ en fonction de chaque arc de ce graphe. En effet :

$$C(H) = \alpha \sum_{(x, y) \in U} [x] + (1 - \alpha) \sum_{i=1}^p f_i l_i,$$

où l_i est la longueur de l'unique chemin de A au sommet où est stockée la valeur de a_i , mais :

$$\sum_{i=1}^p f_i l_i = \sum_{(x, y) \in U} \sum_{i/a_i \in y} f_i$$

donc :

$$C(H) = \alpha \sum_{(x, y) \in U} [x] + (1 - \alpha) \sum_{(x, y) \in U} \sum_{i/a_i \in y} f_i$$

soit :

$$C(H) = \sum_{(x, y) \in U} [\alpha[x] + (1 - \alpha) \sum_{i/a_i \in y} f_i],$$

ou encore, en posant :

$$c(x, y) = \alpha[x] + (1 - \alpha) \sum_{i | a_i \in y} f_i,$$

$$C(H) = \sum_{(x, y) \in U} c(x, y).$$

6. RÉDUCTION DU GRAPHE PORTEUR D'AU MOINS UNE ARBORESCENCE DE PROJECTION OPTIMALE

RÉDUCTION 1 : Soit $G = (S, V)$ le graphe porteur d'une arborescence de projection optimale associée à $\mathcal{R}(A)$ et x un sommet de S tel que : $K(x) = \emptyset$.

Si $|x| \cdot [x] \geq (|x| - 1) \cdot [y]$ pour un y de S tel que $x \subset y$, alors il existe une arborescence de projection optimale qui ne comporte pas l'arc (y, x) .

Démonstration : Soit H^* une arborescence de projection optimale comportant l'arc (y, x) tel que $|x| \cdot [x] \geq (|x| - 1) \cdot [y]$.

Une nouvelle arborescence de projection est obtenue en effectuant les modifications suivantes sur H^* :

- (a) ajouter un arc de y vers chaque successeur de x ;
- (b) supprimer x et ses arcs incidents.

Soit k le nombre de successeurs de x . D'après la propriété 5, $k \leq |x|$.

D'autre part, $x \subset y \Rightarrow [x] \leq [y]$. La seconde hypothèse entraîne :

$$|x| \cdot [x] \geq (|x| - 1) \cdot [y] \Rightarrow (|x| - k) \cdot [x] + k \cdot [x] \geq (|x| - k) \cdot [y] + (k - 1) \cdot [y],$$

mais :

$$0 \leq (|x| - k) \cdot [x] \leq (|x| - k) \cdot [y].$$

Donc : $k \cdot [x] \geq (k - 1) \cdot [y]$ ce qui implique $k \cdot [x] + [y] \geq k \cdot [y]$.

D'après cette dernière inégalité le nombre de pointeurs de la nouvelle arborescence de projection obtenue est inférieur à celui de H^* . D'autre part, le temps d'accès à la valeur de chaque attribut ne change pas ou diminue. ■

Conséquence : Soit $G = (S, V)$ le graphe porteur d'au moins une arborescence de projection optimale de $\mathcal{R}(A)$ et $x \in S$ un sommet (différent de A) tel que $K(x) = \emptyset$. Si tous les prédécesseurs y de x vérifient : $|x| \cdot [x] \geq (|x| - 1) \cdot [y]$ alors il existe une arborescence de projection optimale qui ne contient pas x .

La démonstration est immédiate d'après la réduction 1. ■

RÉDUCTION 2 : Soit $G=(S, V)$ le graphe porteur d'au moins une arborescence de projection optimale de $\mathcal{R}(A)$ et $x \in S$ un sommet (différent de A) tel que $K(x) = \emptyset$. Si $|x| \cdot [x] \geq (|x| - 1) \cdot [A]$ alors il existe une arborescence de projection optimale qui ne contient pas x .

Démonstration : Soit y un prédécesseur de x :

$$y \subset A \Rightarrow [y] \leq [A]; |x| \cdot [x] \geq (|x| - 1) \cdot [A] \Rightarrow |x| \cdot [x] \geq (|x| - 1) \cdot [y].$$

Donc, pour tout sommet y , prédécesseur du sommet x , $|x| \cdot [x] \geq (|x| - 1) [y]$. Cette dernière inégalité permet de conclure, d'après la conséquence de la réduction 1, qu'il existe une arborescence de projection optimale qui ne contient pas x . ■

Conséquence : Soit $G=(S, V)$ le graphe porteur d'au moins une arborescence de projection optimale de $\mathcal{R}(A)$ et $x \in S$ un sommet (différent de A) tel que $K(x) = \emptyset$. Si une clé de A est incluse dans x alors il existe une arborescence de projection optimale qui ne contient pas x .

Démonstration : Si x comporte une clé de A alors $[x] = [A]$. Donc : $|x| [x] > (|x| - 1) [A]$ et d'après la réduction 2, il existe une arborescence de projection optimale qui ne contient pas x . ■

Exemple : Considérons de nouveau la relation $\mathcal{R}(f, l, p, c)$ dont tout ensemble d'attributs contenant f et p est clé. Le graphe $G=(S, V)$ porteur d'au moins une arborescence de projection optimale est donné par la figure 8. (Les arcs de transitivité ne sont pas tracés et les sommets comportant une clé de \mathcal{R} ont été supprimés en vertu de la conséquence de la réduction 2.)

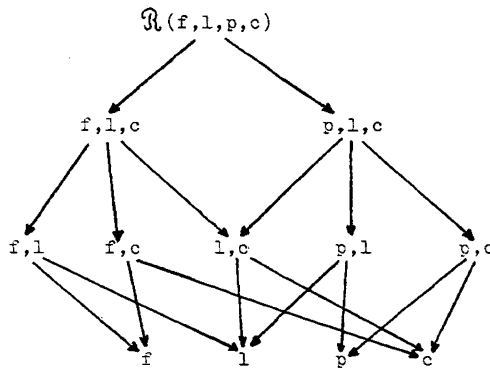


Figure 8

La figure 9 donne le graphe porteur d'au moins une arborescence de projection optimale, obtenu après application des réductions.

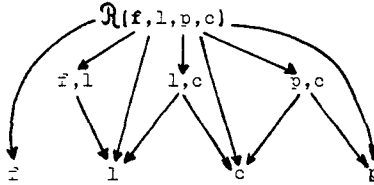


Figure 9

7. UN ALGORITHME POUR CONSTRUIRE UN GRAPHE DE PROJECTION OPTIMAL

Nous allons utiliser le principe de la programmation dynamique [9] pour déterminer un sous-graphe partiel de $G=(S, V)$ qui soit une arborescence de projection optimale. Avant d'appliquer l'algorithme on réduira, au maximum, le graphe $G=(S, V)$.

Rappelons l'expression du coût associé à une arborescence de projection $H=(X, U)$:

$$C(H) = \sum_{(x, y) \in U} c(x, y),$$

avec :

$$c(x, y) = \alpha \cdot [x] + (1 - \alpha) \sum_{i | a_i \in y} f_i.$$

$\forall s \in S$, notons $F^*(s)$ le coût d'une arborescence de projection optimale associée à $\mathcal{R}[s]$.

L'algorithme est initialisé en posant $F^*(s) = 0$ pour tout s tel que $|s| = 1$, puis on calcule $F^*(s)$ pour les sommets associés à deux attributs, à trois attributs... et enfin $F^*(A)$ par la formule d'optimisation séquentielle suivante :

$\forall s \in S$ tel que $|s| \geq 2$:

$$F^*(s) = \text{MIN}_{P_s} \left\{ \sum_{p \in P_s} [F^*(p) + c(s, p)] \right\},$$

où P_s est un sous-ensemble des successeurs de s (dans $G=(S, V)$) qui est une partition des attributs de s , non clés.

La valeur de $F^*(s)$ est donc la valeur d'une solution optimale du problème de partitionnement suivant :

minimiser :

$$\sum_{p \in \Gamma_G(s)} [F^*(p) + c(s, p)] t_p,$$

avec :

$$\sum_{\substack{p \mid a_{s_l} \in P \\ p \in \Gamma_G(s)}} t_p = 1, \quad l = 1, \dots, \mathcal{L},$$

$$\forall p \in \Gamma_G(s), \quad t_p = 0 \text{ ou } 1.$$

où $\{a_{s_1}, \dots, a_{s_{\mathcal{L}}}\}$ est l'ensemble des attributs non clés de s .

Nous pouvons donc utiliser, pour déterminer une arborescence de projection optimale, un des nombreux algorithmes publiés pour résoudre le problème de partitionnement [3, 10].

Exemple : Reprenons le graphe de la figure 9 et déterminons une arborescence de projection optimale dans les deux cas suivants :

$\alpha = 1$. La figure 10 donne le graphe porteur d'au moins une arborescence de projection optimale sur lequel on peut lire, entre parenthèses, le coût de chaque arc. Les arcs tracés en gras représentent une arborescence de projection optimale.

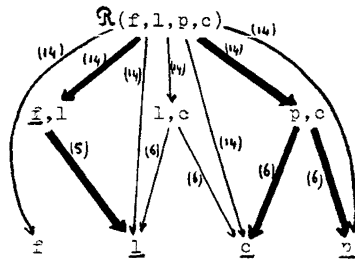


Figure 10

$\alpha = 0,2$. La figure 11 donne le graphe porteur d'au moins une arborescence de projection optimale sur laquelle on peut lire, entre parenthèses, le coût de chaque arc dans l'hypothèse où les fréquences d'accès aux différents attributs sont : $f_x = 0,2$, $f_l = 0,1$, $f_p = 0,4$, $f_c = 0,3$.

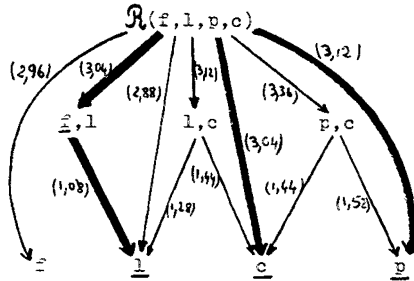


Figure 11

Les arcs tracés en gras correspondent à une arborescence de projection optimale.

Perspective : Nous avons, à la suite de ce travail, entrepris l'étude de l'implantation optimale de l'ensemble d'un schéma relationnel par un système de projecteurs.

REMERCIEMENTS

Je remercie MM. A. Cabanes et E. Pichat qui sont à l'origine de cette étude.

BIBLIOGRAPHIE

1. C. BERGE, *Graphes et Hypergraphes*, Dunod, Paris, 1970.
2. A. BILLIONNET, *Optimal Implementation of Selectors in a Data-Base*, Communication au quatrième Congrès Européen de Recherche Opérationnelle, Cambridge, juillet 1980.
3. A. BILLIONNET, *Une nouvelle méthode pour le problème de partitionnement fondée sur une évaluation par excès de la solution*, R.A.I.R.O., série verte, vol. 15, n° 2, mai 1981, p. 139 à 152.
4. A. CABANES et E. PICHAT, *La définition logique d'une base de données relationnelle; accès et irrédondance des données*, Rapport de recherche de l'Institut d'Informatique d'Entreprise, Paris, juin 1979.
5. CODASYL, *Data Base Task Group Report ACM*, New York, 1971.
6. E. F. CODD, *A Relational Model of Data for Large Shared Data Banks*, CA CM, 13, juin 1970, p. 377 à 387.
7. C. J. DATE, *An Introduction to Data Base Systems*, Addison Wesley, Reading, mars 1977.
8. P. DE, W. D. HASEMAN et C. H. KRIEBEL, *Towards an Optimal Design of a Network Data Base from Relational Descriptions*, Operations Research, vol. 26, n° 5, septembre-octobre 1978.
9. J. L. LAURIERE, *Éléments de programmation dynamique*, Gauthier-Villars, Paris, 1979.
10. R. S. GARFINKEL et G. L. NEMHAUSER, *Integer programming*, chap. 8, John Wiley and Sons, 1972.
11. M. E. SENKO, *Data Structures and Data Accessing in Data Base Systems, Past, Present, Future*, I.B.M. Systems Journal, 16, n° 3, 1977, p. 208 à 257.