Revue française d'informatique et de recherche opérationnelle. Série rouge

E. Pichat

Un algorithme donnant les paves maximaux d'une partie d'un produit de treillis distributifs

Revue française d'informatique et de recherche opérationnelle. Série rouge, tome 5, nº R3 (1971), p. 29-38

<http://www.numdam.org/item?id=M2AN_1971__5_3_29_0>

R.I.R.O.
(5e année, Nº R-3, 1971, p. 29-38)

UN ALGORITHME DONNANT LES PAVES MAXIMAUX D'UNE PARTIE D'UN PRODUIT DE TREILLIS DISTRIBUTIFS

par E. PICHAT (1)

Résumé. — *Étant donné un ensemble fini A d'éléments du produit cardinal de treillis distributifs G_i, cet article donne une méthode de détermination des pavés maximaux compatibles avec A. Il généralise et résoud une conjecture de Tison.*

Dans le cas où chaque treillis G_i est le treillis de Boole à quatre éléments, cet algorithme d'exclusion permet de rechercher les monômes premiers d'une fonction booléenne. Dans le cas où G *est le produit de deux treillis booléens, il détermine les rectangles maximaux inclus dans une partie d'un rectangle ; il permet donc de trouver les ensembles d'articulation minimaux d'un hypergraphe, les décompositions simples additives, multiplicatives et disjonctives maximales d'une fonction booléenne, les implantations d'un graphe dans un autre graphe.*

Étant donné un ensemble fini A d'éléments (ou *pavés*) du produit cardinal G de treillis distributifs G_1, G_2, ..., G_I, cet article donne une méthode de détermination des pavés maximaux compatibles avec A (la notion de compatibilité est précisée dans la première partie); d'autres méthodes ont déjà été exposées dans (Pichat [2]).

Cette méthode ou *algorithme de sélection*, exposé dans la deuxième partie, généralise et résoud une conjecture de (Tison). Pour cela, on autorise l'adjonction en cours d'algorithme de pavés compatibles avec A; la restriction à un pavé devient un homomorphisme relativement au processus de sélection; un raisonnement par induction achève la démonstration.

La troisième partie fournit des applications. Dans le cas où chaque treillis G_i est le treillis de Boole à quatre éléments, l'algorithme de sélection permet en particulier de rechercher les monômes premiers d'une fonction booléenne.

(1) Institut d'Informatique d'Entreprise (C.N.A.M., Paris). Institut de Mathématiques Appliquées de Grenoble.

Dans le cas où G est le produit de deux treillis booléens, il détermine les rectangles maximaux inclus dans une partie d'un rectangle; il permet donc de trouver les ensembles d'articulation minimaux d'un hypergraphe (Berge), les décompositions simples additives, multiplicatives et disjonctives maximales d'une fonction booléenne (Pichat [1]), les implantations (Malgrange).

I. PRELIMINAIRES

1.1 Définitions

Soient $G_1, G_2, \ldots, G_I$ I treillis distributifs, $\underset{i}{\cdot}$ et $\underset{i}{+}$ les opérations *borne inférieure* et *borne supérieure* (appelées aussi respectivement produit et réunion) de chacun d'eux ($i = 1, 2, \ldots, I$); si b_i, c_i et d_i, sont des éléments arbitraires de G_i, rappelons (Dubreil, Dubreil-Jacotin) :

$\underset{i}{\cdot}$ et $\underset{i}{+}$ sont associatifs, commutatifs, idempotents

$$[b_i \underset{i}{\cdot} (b_i \underset{i}{+} c_i) = b_i,] b_i \underset{i}{+} (b_i \underset{i}{\cdot} c_i) = b_i$$

$$b_i \underset{i}{\cdot} (c_i \underset{i}{+} d_i) = (b_i \underset{i}{\cdot} c_i) \underset{i}{+} (b_i \underset{i}{\cdot} d_i) [, b_i \underset{i}{+} (c_i \underset{i}{\cdot} d_i) = (b_i \underset{i}{+} c_i) \underset{i}{\cdot} (b_i \underset{i}{+} d_i)]$$

Une *relation d'ordre*, notée $\underset{i}{\leqslant}$, peut être introduite dans G_i par :

$$b_i \underset{i}{\leqslant} c_i \Leftrightarrow b_i \underset{i}{\cdot} c_i = b_i \qquad [\Leftrightarrow b_i \underset{i}{+} c_i = c_i]$$

Soit $G = G_1 \times G_2 \times \ldots \times G_I$ l'ensemble *produit* (Bourbaki) (cardinal [Dubreil-Jacotin, Lesieur, Croisot]) *des treillis* $G_1, G_2, \ldots, G_I$; ses éléments seront appelés des *pavés*. Il est muni d'opérations :

$$b \underset{i}{*} c = (b_1 \underset{1}{\cdot} c_1, \ldots, b_{i-1} \underset{i-1}{\cdot} c_{i-1}, b_i \underset{i}{+} c_i,$$
$$b_{i+1} \underset{i+1}{\cdot} c_{i+1}, \ldots, b_I \underset{I}{\cdot} c_I), i = 1, 2, \ldots, I$$

et de *la relation d'ordre, notée* $\leqslant$, *produit des relations d'ordre* $\underset{i}{\leqslant}$; donc, pour tous pavés $b = (b_1, \ldots, b_i, \ldots, b_I)$ et $c = (c_1, \ldots, c_i, \ldots, c_I)$ de G, on a :

$$b \leqslant c \Leftrightarrow b_i \underset{i}{\leqslant} c_i \qquad \text{pour tout } i = 1, 2, \ldots, I.$$

Enfin, nous dirons qu'un pavé b est *compatible avec un* ensemble A de pavés (ou, improprement, est *pavé de* A) si lui-même ou un pavé qui lui est supérieur peuvent être engendrés à partir de A à l'aide des opérations $\underset{i}{*}$. Un ensemble B de pavés est dit *compatible avec* A si chacun de ses éléments est compatible avec A; inversement, A sera appelé une *couverture* de B.

Remarquons que quand $I = 1$, il existe une seule opération $\underset{i}{*}$ qui est $\underset{1}{+}$.

1.2. Propriétés élémentaires

a) $\leqslant$ *induit sur l'ensemble $\mathcal{T}(G)$ des ensembles de pavés de G, un pré-ordre :* si B et C désignent des parties de G

$B \leqslant C \Leftrightarrow$ pour tout pavé b de B, il existe un pavé c de C tel que $b \leqslant c$.

b) *Les opérations $\underset{i}{*}$ sont isotones.*

$$b \leqslant c \Rightarrow b \underset{i}{*} d \leqslant c \underset{i}{*} d$$

cela résulte immédiatement de :

$$b_i \underset{i}{\leqslant} c_i \Rightarrow b_i \underset{i}{\cdot} d_i \underset{i}{\leqslant} c_i \underset{i}{\cdot} d_i \quad \text{et} \quad b_i \underset{i}{+} d_i \underset{i}{\leqslant} c_i \underset{i}{+} d_i$$

où b_i, c_i et d_i désignent des éléments arbitraires de G_i.

c) *La relation de compatibilité est une relation de pré-ordre dans $\mathcal{T}(G)$.*

d) *L'ensemble produit G des treillis distributifs G_i muni de l'ordre $\leqslant$ produit des ordres des G_i et des opérations*

borne inférieure $\cdot$ faisant correspondre à deux pavés b et c le pavé

$$b \cdot c = (b_1 \underset{1}{\cdot} c_1, b_2 \underset{2}{\cdot} c_2, \ldots, b_I \underset{I}{\cdot} c_I)$$

borne supérieure $\cdot$ faisant correspondre à deux pavés b et c le pavé

$$b + c = (b_1 \underset{1}{+} c_1, b_2 \underset{2}{+} c_2, \ldots, b_I \underset{I}{+} c_I),$$

est un treillis distributif.

1.3. Restriction de pavés

Si $b = (b_1, b_2, \ldots, b_I)$ est un pavé de G, nous appellerons *restriction à b du pavé $c = (c_1, c_2, \ldots, c_I)$*, le pavé

$$R_b c = (b_1 \underset{1}{\cdot} c_1, b_2 \underset{2}{\cdot} c_2, \ldots, b_I \underset{I}{\cdot} c_I),$$

c'est-à-dire la borne inférieure $b \cdot c$ de b et c dans le treillis produit G.

a) *La restriction à un pavé b de G, R_b, est un homormophisme relativement aux opérations $\underset{i}{*}$:*

$$R_b(c \underset{i}{*} d) = R_b c \underset{i}{*} R_b d$$

C'est une conséquence immédiate d'une part de la distributivité de $\underset{i}{\cdot}$ par rapport à $\underset{i}{+}$, d'autre part de l'idempotence, de l'associativité et de la commutativité de $\underset{i}{\cdot}$ pour tout i :

$$R_b(c \underset{i}{*} d) = b(c \underset{i}{*} d) = (bc) \underset{i}{*} (bd) = R_b c \underset{i}{*} R_b d$$

b) *L'application « restriction à un pavé b » de G dans G est croissante :*

$$c \leqslant d \Rightarrow R_b c \leqslant R_b d$$

C'est une conséquence immédiate de la propriété *a* précédente.

c) *Si un pavé c est compatible avec un ensemble A de pavés, sa restriction $R_b c$ est compatible avec l'ensemble des restrictions des pavés de A.*

C'est une conséquence des deux propriétés précédentes.

II. UN ALGORITHME DETERMINANT LES PAVES MAXIMAUX D'UNE PARTIE D'UN PRODUIT DE TREILLIS DISTRIBUTIFS.

Étant donné un ensemble fini A de pavés, montrons qu'il est possible d'engendrer les pavés compatibles avec A *maximaux* (ou *premiers*) relativement à $\leqslant$, à l'aide de l'algorithme suivant :

II.1. Algorithme de sélection

On part d'un ensemble A de pavés, appelés **pavés de droite** *et écrits* **en file** (ou ensemble totalement ordonné ou suite ou fichier ou vecteur [Harrand]).

On **sélectionne** *chaque pavé de droite c, c'est-à-dire qu'on le fait passer de droite à gauche en adjoignant aux pavés de droite* $\{ d^k \}$ *leurs* $\underset{i}{*}$ *— composés avec c,* $\{ c \underset{i}{*} d^k \}$ *pour tout i et pour tout* $\{ d^k \}$, *s'ils ne sont pas inférieurs ou égaux à un pavé de la file* (*pavé de droite* ou **pavé de gauche**).

On peut adjoindre à droite un pavé s'il est compatible avec A.

On peut supprimer un pavé de droite inférieur ou égal à un pavé de droite ou de gauche.

L'algorithme existe et est déterminé quand il n'y a plus de pavés à droite.

Remarquons qu'un tel algorithme appelé *algorithme de sélection S(A)* n'est pas uniquement déterminé (en particulier à cause de l'adjonction possible de pavés à droite).

Nous appellerons *ensemble de pavés d'un algorithme de sélection* tout ensemble de ses files de pavés de gauche et de pavés de droite (ils seront notés (0), (1), ..., (ν), ...) et *restriction au pavé b d'un ensemble de pavés* (ν) la réunion de l'ensemble des restrictions à b des pavés de gauche de (ν) et de l'ensemble des restrictions à b des pavés de droite de (ν) (cette restriction sera notée $R_b\ \nu$). Plus explicitement, si (ν) à la configuration

$\{g^j\}$	$\{d^k\}$	(ν)
ensemble de pavés de gauche	ensemble de pavés de droite	

($R_b\ \nu$) aura la configuration

$$\{R_b g^j\} \qquad \{R_b d^k\} \qquad (R_b \nu)$$

***a*) Lemme**

Étant donnés un algorithme de sélection $S(A)$ appliqué à un ensemble A de pavés et un pavé b, il existe un algorithme de sélection $S(R_bA)$ appliqué à R_bA ayant pour suite d'ensembles de pavés les restrictions à b des ensembles de pavés de $S(A)$.

Démonstration :

Raisonnons par induction. Nous supposons que ($R_b\ \nu$) est obtenu à partir de (R_b 0) à l'aide d'un algorithme $S(R_b\ A)$ et que ($R_b\ \nu$) est la restriction de (ν), c'est-à-dire : tout pavé de gauche (respectivement de droite) de ($R_b\ \nu$) est la restriction d'un pavé de gauche (respectivement de droite) de (ν) et tout pavé de gauche (respectivement de droite) de (ν) a un pavé de gauche (respectivement de droite) de ($R_b\ \nu$) pour restriction ; montrons que si ($\nu + 1$) s'obtient à partir de (ν) par l'une des opérations suivantes : adjonction d'un pavé à droite ou à gauche, suppression d'un pavé de droite, il existe des opérations compatibles avec $S(R_b\ A)$ permettant de transformer ($R_b\ \nu$) en la restriction ($R_b\ \nu + 1$) de ($\nu + 1$) à b.

$$\begin{tikzcd}
0 = \{\emptyset, A\} \arrow[r, "R_b"] \arrow[d, dotted, "S(A)"] & (R_b 0) = \{\emptyset, R_b A\} \arrow[d, dotted, "S(R_b A)"] \\
(\nu) \arrow[r, "R_b"] \arrow[d, "S(A)"] & (R_b \nu) \arrow[d, "S(R_b A)\ ?"] \\
(\nu+1) \arrow[r, "R_b"] & (R_b \nu + 1)
\end{tikzcd}$$

Figure 1

1. Adjonction du pavé c aux pavés de droite de (ν) : alors $R_b\ c$, s'il n'est pas un pavé de droite de ($R_b\ \nu$), doit être adjoint à la file de pavés de droite de ($R_b\ \nu$) (c'est licite d'après *I*-3*c*).

2. Adjonction du pavé c aux pavés de gauche de (ν) : alors $R_b\ c$, s'il n'est pas pavé de gauche de $(R_b\ \nu)$, doit être adjoint à l'ensemble de pavés de gauche de $(R_b\ \nu)$; chacune des adjonctions $R_b c \underset{i}{*} R_b d^k = R_b(c \underset{i}{*} d^k)$ impliquées par la sélection de $R_b\ c$ sera par la suite automatiquement réalisée comme restriction à l'adjonction de $c \underset{i}{*} d^k$ impliquée par la sélection de c.

3. Suppression du pavé de droite c de (ν) : alors $R_b\ c$, s'il est un pavé de droite de $(R_b\ \nu)$ restriction du seul pavé c de (ν), doit être supprimé de l'ensemble de pavés de droite de $(R_b\ \nu)$ (c'est licite si c est inférieur ou égal à un pavé e, puisque $R_b\ c$ est alors inférieur ou égal au pavé $R_b\ e$ de $(R_b\ \nu)$; c'est aussi licite si la suppression de c est le prélude de sa sélection).

Ce lemme permet donc d'appeler la suite des restrictions à b des files de pavés de $S(A)$ *algorithme* $S(R_b\ A)$ *restriction à b de l'algorithme* $S(A)$.

***b*) Théorème**

Soit b un pavé compatible avec A.S(A) sélectionne un pavé supérieur ou égal à b.

Démonstration : par induction sur la *dimension* (ou somme des hauteurs [Dubreil-Jacotin, Lesieur, Croisot] de leurs composantes) des pavés. En effet, montrons que pour tout pavé b et pour toute couverture B de b formée de pavés inférieurs ou égaux à b, tout algorithme $S(B)$, quand il est terminé, admet b comme pavé de gauche, en supposant cette propriété vérifiée pour les pavés de dimension inférieure à la dimension de b. Les pavés dont toutes les composantes ont des hauteurs au plus égales à un vérifiant cette propriété, il en résultera que les algorithmes restrictions de $S(A)$ aux pavés b compatibles avec A, admettent ces pavés b comme pavés à gauche, donc que $S(A)$ admet comme pavés à gauche des pavés supérieurs ou égaux aux b.

$S(B)$ fera passer à gauche soit b, et le théorème est démontré, soit tous les pavés *immédiatement inférieurs* à b (pavés dont I-1 composantes sont égales aux composantes correspondantes de b et dont une est immédiatement inférieure à la composante correspondante de b) : plaçons-nous au moment où va passer à gauche un pavé immédiatement inférieur à b appelé $c = (c_1, b_2, ..., b_I)$ et où ne seront pas encore passés à gauche un et un seul pavé immédiatement inférieur à b, distinct de c et ayant pour expression $e = (e_1, b_2, ..., b_I)$, et au plus I-1 pavés immédiatement inférieurs à b

$$f^\alpha = (b_1, b_2, ..., b_{\alpha-1}, f_\alpha, b_{\alpha+1}, ..., b_I),$$

α parmi 2, 3, ..., I; il y a donc à droite c et de quoi couvrir e.

On peut éliminer tout pavé de droite dont une i$^{\text{ème}}$ composante, avec i distinct de 1 et des α, est inférieure à b_i ou dont une $\alpha^{\text{ème}}$ composante est inférieure à b_α sans être inférieure ou égale à f_α. Les pavés de droite dont une $\alpha^{\text{ème}}$ composante est inférieure ou égale à f_α ne peuvent alors engendrer, comme nouveaux pavés, que des pavés à $\alpha^{\text{ème}}$ composante inférieure ou égale à f_α. Il existe donc nécessairement pour couvrir e des pavés dont chaque $i^{\text{ème}}$ composante, pour $i = 2, 3, \ldots, I$, est b_i; la combinaison de l'un d'eux de première composante non inférieure ou égale à c_1 avec c, lors de la sélection de c, entraîne la formation de b.

II.2. Algorithme de sélection pour déterminer les pavés maximaux

Appelons *ensemble réduit* ou ensemble obtenu par *réduction* d'un ensemble A de pavés l'ensemble de pavés obtenu à partir de A d'abord en ne gardant qu'un exemplaire des pavés de A, puis en supprimant les pavés inférieurs à d'autres; il sera noté Max A.

L'algorithme de sélection $S(A)$ devient, grâce au théorème précédent, un algorithme de recherche des pavés maximaux compatibles avec A : ce sont, après réduction, les pavés de gauche trouvés en fin d'algorithme.

Pour que l'ensemble de pavés de droite devienne vide, on peut en particulier :

— supprimer totalement les opérations arbitraires d'adjonction de pavés de droite;

— ou supprimer les opérations d'adjonction de pavés de droite inférieurs ou égaux à un pavé déjà connu; cette façon de procéder a l'avantage sur la précédente d'accélérer la convergence de l'algorithme par l'utilisation de pavés nouveaux ;

— ou effectuer des opérations d'adjonction de pavés de droite « pas trop nombreuses » vis-à-vis de la cadence de sélection; que l'ensemble de pavés de droite devienne vide dans les deux premières options provient de ce que les pavés compatibles avec A sont en nombre fini et de la croissance de l'ensemble de pavés inférieurs ou égaux à un pavé de gauche.

La réduction à tout instant de l'ensemble des pavés (et non seulement de l'ensemble des pavés de droite et de l'ensemble des pavés de gauche indépendamment l'un de l'autre) est possible à cause de l'isotonie des opérations $\underset{i}{*}$ et parce qu'un pavé de droite peut jouer tout rôle joué par un pavé qui lui est inférieur; elle paraît opérationnellement intéressante.

Énonçons l'algorithme de sélection en effectuant toute réduction dès qu'elle est possible :

Algorithme de sélection

On part d'un ensemble réduit Max A de pavés, écrit en file.

1. On « sélectionne » successivement chaque élément de la file en commençant par le premier à partir de la gauche, puis le deuxième, ..., tant qu'il en existe, en recherchant les $\underset{i}{*}$ — composés qu'il admet avec les éléments à droite; un tel $\underset{i}{*}$ — composé une fois formé,

— s'il est inférieur ou égal à un élément de la file, on n'en tient pas compte;

— sinon, on supprime les éléments de la file qui lui sont inférieurs et on l'adjoint en fin de file.

2. On peut toujours en fin de file adjoindre des éléments non inférieurs ou égaux à un élément de la file compatibles avec A.

Les pavés restants sont les pavés maximaux compatibles avec A.

REMARQUE 1

Pour sélectionner un pavé p, il est suffisant (mais non obligatoirement intéressant) de rechercher ses $\underset{i}{*}$ — composés avec les éléments qui sont à sa droite lors de son transfert de l'ensemble $\{d^k\}$ de pavés de droite à l'ensemble de pavés de gauche, puisque $p \underset{j}{*} (p \underset{i}{*} d^k)$ est inférieur ou égal à p ou à $p \underset{i}{*} d^k$ et puisque les opérations $\underset{i}{*}$ sont isotones. Par contre il est nécessaire de rechercher les $\underset{i}{*}$ — composés de p avec les pavés éventuellement adjoints en vertu de la clause 2 de l'algorithme, s'ils ont entraînés la suppression de pavés de droite.

REMARQUE 2

Il peut être intéressant pour gagner de la place en mémoire et pour éviter les ruptures de séquences, de combler, aussitôt formé, le vide créé par la suppression d'un élément à droite du pavé sélectionné p (respectivement à gauche) en y portant le dernier élément de la liste ou le nouvel $\underset{i}{*}$ — composé introduit (respectivement le premier élément de la liste). Mieux, si un $p \underset{i}{*} d^k$ est supérieur à p, il peut remplacer p; le processus de sélection de p est alors immédiatement arrêté, faisant place au processus de sélection de son remplaçant (la justification théorique est l'adjonction de $p \underset{i}{*} d^k$ à l'ensemble de pavés de droite).

III. APPLICATIONS

Supposons que :

— les treillis G_i sont booléens;

— nous ne nous intéressons qu'aux pavés dont aucune composante n'est nulle (ou le plus petit élément) : il est alors superflu de considérer au cours de l'algorithme de sélection les pavés ayant au moins une composante nulle, puisque leurs $\underset{i}{*}$ — composés sont soit inférieurs ou égaux à l'un des pavés qui leur a donné naissance, soit ont une composante nulle,

et considérons les deux cas particuliers suivants :

III.1. Cas où $I = 2$

Ce problème est posé dans (Malgrange). L'algorithme de sélection permet de déterminer les pavés (ou *rectangles*) maximaux inclus dans une partie d'un rectangle.

Reprenons l'exemple de (Kaufmann), p. 364. L'algorithme de sélection donne :

(A, bdef) (B, ac) (C, bde) (D, bcf) (AC, bde) (AD, bf) (BD, c) (ACD, b)

Figure 2

Les accolades indiquent les ensembles de rectangles de droite successifs avant la sélection de leur rectangle de tête; le trait supérieur indique la réduction de l'ensemble de rectangles. Dans cet exemple, les bornes supérieures sont notées multiplicativement.

III.2. Cas où chaque treillis G_i est le treillis de Boole à quatre éléments

L'algorithme de sélection donne l'algorithme de (Tison), p. 4-19 de détermination des monômes premiers (Kuntzmann) d'une fonction booléenne : Il a été programmé et s'avère sur calculatrice numérique moins rapide que l'algorithme de recherche de monômes premiers par consensus effectués successivement par rapport à chacune des variables, contrairement à ce qu'on pourrait penser. Cette anomalie s'explique par une disposition particulière des monômes qu'utilise (Benzaken) : pour calculer les consensus par rapport à une variable x d'un ensemble de monômes, sont envisagés les seuls couples du produit de l'ensemble de monômes contenant x et de l'ensemble des monômes contenant x'; ainsi la considération de nombreux couples de monômes est remplacée par un tri initial des monômes en ceux contenant x, ceux contenant x' et ceux ne contenant ni x ni x'.

Dans le cas général (produit de treillis distributifs à plus de quatre éléments), la comparaison doit être avantageuse à l'algorithme de sélection.

REFERENCES BIBLIOGRAPHIQUES

C. BENZAKEN, *Programme de calcul des monômes premiers d'une fonction booléenne.* Institut de Mathématiques Appliquées, Grenoble, 1964.

C. BERGE, *Graphes et hypergraphes,* Dunod, 1970, 502 p.

N. BOURBAKI, *Éléments de mathématiques. Ensembles ordonnés.* Fascicule XX, 2e éd., Hermann, 1963, 148 p.

P. DUBREIL, M. L. DUBREIL-JACOTIN, *Leçons d'algèbre moderne,* Dunod, 1961, 393 p.

M. L. DUBREIL-JACOTIN, L. LESIEUR et R. CROISOT, *Leçons sur la théorie des treillis des structures algébriques ordonnées et des treillis géométriques,* Gauthier-Villars, 1953, 385 p.

Y. HARRAND, *Traitement des files et des listes,* Dunod, 1967, 120 p.

A. KAUFMANN, *Méthodes et modèles de la recherche opérationnelle.* Tome 2, 2e éd., Dunod, 1968, 609 p.

J. KUNTZMANN, *Algèbre de Boole,* 2e éd., Dunod, 1968, 361 p.

Y. MALGRANGE, *Recherche des sous-matrices premières d'une matrice à coefficients binaires. Applications à certains problèmes de graphe,* p. 231-242 dans Deuxième Congrès de l'AFCALTI, octobre 1961, Gauthier-Villars, 1962, 524 p.

E. PICHAT [1], *Décompositions simples de fonctions booléennes,* Revue Française d'Informatique et de Recherche Opérationnelle, n° 7, 1968, p. 61-70,

E. PICHAT [2], *Algorithms for finding the maximal elements of a finite universal algebra.* Proc. IFIP Congress 1968, Edinburgh, booklet A, p. 96-101, ou Information processing 68, North-Holland, Amsterdam, 1969, p. 214-218.

P. TISON, *Théorie des consensus.* Thèse, Grenoble, 1965.