



**HAL**  
open science

## Synthèses booléennes dans les réseaux cellulaires

Jean-Antoine Montagnon

► **To cite this version:**

Jean-Antoine Montagnon. Synthèses booléennes dans les réseaux cellulaires. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1971. Français. NNT: . tel-00282891

**HAL Id: tel-00282891**

**<https://theses.hal.science/tel-00282891>**

Submitted on 28 May 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

n° d'ordre

T H E S E

présentée à

LA FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR INGENIEUR

par

*J.A. MONTAGNON*

SYNTHESES BOOLEENNES DANS LES RESEAUX CELLULAIRES

-----

Thèse soutenue le 19 février 1971 devant la commission d'examen

<i>Monsieur J. KUNTZMANN</i>	Président
<i>Monsieur R. PERRET</i>	Examineur
<i>Monsieur C. BENZAKEN</i>	Examineur
<i>Madame G. SAUCIER</i>	Examineur



## LISTE DES PROFESSEURS

Doyen Honoraire : Monsieur M.MORET  
Doyen : Monsieur E.BONNIER

### PROFESSEURS TITULAIRES

MM.	NEEL Louis	Physique Expérimentale
	KRAVTCHENKO Julien	Mécanique Rationnelle
	CHABAUTY Claude	Calcul différentiel et intégral
	BENOIT Jean	Radioélectricité
	CHENE Marcel	Chimie Papetière
	FELICI Noël	Electrostatique
	KUNTZMANN Jean	Mathématiques Appliquées
	BARBIER Reynold	Géologie Appliquée
	SANTON Lucien	Mécanique des Fluides
	OZENDA Paul	Botanique
	FALLOT Maurice	Physique Industrielle
	KOSZUL Jean-Louis	Mathématiques
	GALVANI Octave	Mathématiques
	MOUSSA André	Chimie Nucléaire
	TRAYNARD Philippe	Chimie Générale
	SOUTIF Michel	Physique Générale
	CRAYA Antoine	Hydrodynamique
	REULOS René	Théorie des Champs
	BESSON Jean	Chimie Minérale
	AYANT Yves	Physique Approfondie
	GALLISSOT François	Mathématiques
Mlle	LUTZ Elisabeth	Mathématiques
MM.	BLANBERT Maurice	Mathématiques
	BOUCHEZ Robert	Physique Nucléaire
	LLIBOUTRY	Géophysique
	MICHEL Robert	Minéralogie et pétrographie
	BONNIER Etienne	Electrochimie et Electrometallurgie
	DESSAUX Georges	Physiologie Animale
	PILLET Emile	Physique Industrielle-Electrotechnique
	YOCCOZ Jean	Physique Nucléaire théorique
	DEBELMAS Jacques	Géologie Générale
	BERBER Robert	Mathématiques
	PAUTENET René	Electrotechnique
	MALGRANGE Bernard	Mathématiques Pures
	VAUQUOIS Bernard	Calcul Electronique
	BARJON Robert	Physique Nucléaire
	BARBIER Jean-Claude	Physique
	SILBERT Robert	Mécanique des Fluides
	BUYLE-BODIN Maurice	Electronique
	DREYFUS Bernard	Thermodynamique

MM.	KLEIN Joseph	Mathématiques
	VAILLANT François	Zoologie et Hydrobiologie
	ARNAUD Paul	Chimie
	SENGEL Philippe	Zoologie
	BARNOUD Fernand	Biosynthèse de la Cellulose
	BRISSONNEAU Pierre	Physique
	GAGNAIRE Didier	Chimie Physique
Mme	KOFLER Lucie	Botanique
MM.	DEGRANGE Charles	Zoologie
	PEBAY-PEROULA Jean Claude	Physique
	RASSAT André	Chimie Systématique
	DUCROS Pierre	Cristallographie Physique
	DODU Jacques	Mécanique Appliquée I.U.T.
	ANGLES D'AURIAC Paul	Mécanique des Fluides
	LACAZE Albert	Thermodynamique
	GASTINEL Noël	Analyse Numérique
	GIRAUD Pierre	Géologie
	PERRET René	Servo-mécanisme
	PAYAN Jean Jacques	Mathématiques Pures

#### PROFESSEURS SANS CHAIRE

MM.	GIDON Paul	Géologie
Mme	BARBIER Marie-Jeanne	Electrochimie
Mme	SOUTIF Jeanne	Physique
MM.	COHEN Joseph	Electrotechnique
	DEPASSEL R.	Mécanique des Fluides
	GLENAT René	Chimie
	BARRA Jean	Mathématiques Appliquées
	COUMES André	Electronique
	PERRIAUX Jacques	Géologie et Minéralogie
	ROBERT André	Chimie Papetière
	BIARREZ Jean	Mécanique Physique
	BONNET Georges	Electronique
	CAUQUIS Georges	Chimie Générale
	BONNETAIN Lucien	Chimie Minérale
	DEPOMMIER Pierre	Physique Nucléaire-Génie Atomique
	HACQUES Gérard	Calcul Numérique
	POLOUJADOFF Michel	Electrotechnique
Mme	KAHANE Josette	Physique
Mme	BONNIER Jane	Chimie
MM.	VALENTIN Jacques	Physique
	REBECQ Jacques	Biologie
	DEPORTES Charles	Chimie
	SARROT-REYNAULD Jean	Géologie
	BERTRANDIAS Jean Paul	Mathématiques
	AUBERT Guy	Physique

#### PROFESSEURS ASSOCIES

MM.	RODRIGUES Alexandre	Mathématiques Pures
	MORITA Susumu	Physique Nucléaire
	RADHAKRISHNA	Thermodynamique

MAITRES DE CONFERENCES

MM.	LANCIA Roland	Physique Atomique
Mme	BOUCHE Liane	Mathématiques
MM.	KAHANE André	Physique Générale
	DOLIQUE Jean Michel	Electronique
	BRIERE Georges	Pyysique
	DESRE Georges	Chimie
	LAJZEHOWICZ Joseph	Physique
	LAURENT Pierre	Mathématiques Appliquées
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	LONGEQUEUE Jean Pierre	Physique
	SOHM Jean Claude	Electrochimie
	ZADWORNY François	Electronique
	DURAND François	Chimie Physique
	CARLIER Georges	Biologie Végétale
	PFISTER Jean Claude	Physique
	CHIBON Pierre	Biologie Animale
	IDELMAN Simon	Physiologie animale
	BLOCH Daniel	Electrotechnique I.P.
	MARTIN-BOUYER Michel	Chimie (C.S.U.Chambery)
	SIBILLE Robert	Construction mécanique (I.U.T.)
	BRUGEL Lucien	Energétique I.U.T.
	BOUVARD Maurice	Hydrologie
	RICHARD Lucien	Botanique
	PELMONT Jean	Physiologie animale
	BOUSSARD Jean Claude	Mathématiques Appliquées(I.P.G.)
	MOREAU René	Hydraulique I.P.G.
	ARMAND Yves	Chimie I.U.T.
	BOLLIET Louis	Informatique I.U.T.
	KUHN Gérard	Energétique I.U.T.
	PEFFEN René	Chimie I.U.T.
	GERMAIN Jean Pierre	Mécanique
	JOLY Jean René	Mathématiques Pures

UNIVERSITÉ DE GRENOBLE  
SERVICE DE MATHÉMATIQUES  
APPLIQUÉES  
POLYCOPIES  
CEDEX n° 53  
38 - GRENOBLE - GARE

Mlle	PIERY Yvette	Biologie Animale
MM.	BERNARD Alain	Mathématiques Pures
	MOHSEN Tahsin	Biologie (C.S.U.Chambery)
	CONTE René	Mesures Physiques I.U.T.
	LE JUNTER Noël	Génie Electrique Electronique I.U.T.
	LE ROY Philippe	Génie Mécanique I.U.T.
	ROMIER Guy	Technique Statistiques Quantitatives I.U.T
	VIALON Pierre	Géologie
	BENZAKEN Claude	Mathématiques Appliquées
	MAYNARD Roger	Physique
	DUSSAUD René	Mathématiques (C.S.U.Chambery)
	BELORIZKY Elie	Physique (C.S.U Chambery)
Mme	LAJZEROWICZ Jeanine	Physique (C.S.U Chambery)
M.	JULLIEN Pierre	Mathématiques Pures
Mme	RINAUDO Marguerite	Chimie
MM.	BLIMAN Samuel	E.I.E.
	BEGUIN Claude	Chimie Organique
	NEGRE Robert	I.U.T.

#### MAITRES DE CONFERENCES ASSOCIES

MM.	YAMADA Osamu	Physique du Solide
	NAGAO Makoto	Mathématiques Appliquées
	MAREZIO Massimo	Physique du Solide
	CHEECKE John	Thermodynamique
	BOUDOURIS Georges	Radioélectricité
	ROZMARIN Georges	Chimie Papetière

*Je tiens à exprimer toute ma reconnaissance à Monsieur le Professeur J. KUNTZMANN, Directeur du Service de Mathématiques Appliquées de l'Université de Grenoble, qui a dirigé et inspiré ce travail, et qui par ses précieux conseils m'a permis de le mener à bien.*

*Je suis particulièrement sensible à l'honneur qu'il m'a fait en acceptant de présider le Jury.*

*Je remercie vivement*

*Monsieur R. PERRET, Directeur du Laboratoire d'Automatisme, Professeur à la Faculté des Sciences de Grenoble,*

*Monsieur C. BENZAKEN, Maître de Conférences à la Faculté des Sciences de Grenoble,*

*Madame G. SAUCIER, Docteur es Sciences,*

*qui ont bien voulu accepter de faire partie du Jury.*

*Je tiens à remercier la D.G.R.S.T. qui par son aide financière a soutenu la présente recherche, le S.C.T.I. et la C.I.I. qui m'ont permis de la mener à terme.*

*Je remercierai aussi Monsieur J.C. SAILLARD qui m'initia à la programmation et Monsieur A. BEAN dont l'expérience me fut précieuse.*

*Mes remerciements iront encore à Mademoiselle BICAÏS dont j'ai apprécié la compétence pour la frappe de ces feuilles, ainsi qu'au Service de Tirage.*





## CHAPITRE - I

### INTRODUCTION AUX SYNTHÈSES CELLULAIRES

#### I-1 - LES SYNTHÈSES BOOLEENNES EN GENERAL : ROLE DES SYNTHÈSES CELLULAIRES

Une synthèse sera dite cellulaire, modulaire, itérative, répétitive, etc... si le réseau correspondant offre une disposition régulière des opérateurs (cellules) utilisés et de leurs liaisons.

Par régulière nous entendons la répétition d'un motif de base ou d'un certain nombre de ses variantes.

L'idée de rendre cellulaire la réalisation de fonctions booléennes remonte assez loin dans le temps (les matrices de diode par exemple), mais elle bénéficie actuellement d'un regain d'intérêt.

Cet intérêt s'explique par l'évolution de la technologie, la complexité des circuits qui s'accroît de façon concomitante. La manière de les réaliser (masques, photorépétition etc...), de les tester pose de nouveaux problèmes. Cet esprit de modularité se rencontre d'ailleurs à plusieurs niveaux hiérarchiques dans la conception de calculatrices.

De nombreux mathématiciens ont travaillé sur l'aspect séduisant des réseaux cellulaires, et la littérature spécialisée fait foi de cette floraison de recherches.

Une synthèse booléenne répond au problème suivant : élaboration d'une (ou plusieurs) fonction(s) à l'aide d'opérateurs reliés entre eux.

A ce stade il y a déjà au moins deux choix :

- 1 - celui de la méthode (par composition, par destruction, etc...)
- 2 - celui de l'ensemble des opérateurs de base qui seront utilisés (ET/OU, NI,U, Majorité etc...).

A ces deux choix principaux viennent généralement se greffer des contraintes plus particulières :

- 3 - nombre d'entrées/sorties des opérateurs : illimité ou non
- 4 - dispose-t-on comme entrées des variables sous les deux formes, directe et complémentée, ou sous une seule.
- 5 - nombre de "couches" du réseau, c'est-à-dire nombre maximal d'opérateurs traversés lorsqu'on va de l'entrée à la sortie du réseau.
- 6 - planéarité du réseau.
- 7 - nombre de types d'opérateurs différents.
- 8 - test du réseau.
- 9 - nombre maximal de cellules utilisées.

Dans leur cahier des charges, les synthèses booléennes inscrivent toutes ces alternatives. Les synthèses cellulaires ont pour originalité de fixer un cadre à la disposition relative des cellules :

- 10 - disposition régulière d'un motif de base.
- 11 - les liaisons entre cellules sont distribuées dans un certain nombre de directions bien précisées au départ, on parle très souvent des "cellules voisines" d'une cellule donnée, c'est-à-dire les seules cellules susceptibles d'être reliées à la cellule en question.

Les synthèses cellulaires essaient ainsi de résoudre un problème très délicat : l'implantation du réseau logique réalisant la fonction. Ce point est traité directement par la synthèse qui sous-entend ce problème, puisqu'elle régit elle-même la disposition des cellules.

En particulier le problème de la planarité peut être résolu au départ et ne se pose plus par la suite..

Si les synthèses cellulaires apportent une solution à certains problèmes épineux, elles prêtent bien sûr le flanc à certaines critiques :

- \* le nombre de cellules utilisées est souvent très grand. Mais selon certains, ceci ne devrait jouer qu'une importance décroissante grâce à l'intégration à grande échelle (LSI en américain).
- \* nombre de couches plus élevé que les circuits classiques à deux/trois couches. Ceci touche à la rapidité du signal transmis.

Nous ne nous attacherons pas ici à la défense ou à l'accusation des synthèses cellulaires. Chacun peut y trouver ce qu'il veut bien y mettre.

## I-2 - LES SYNTHÈSES CELLULAIRES CLASSIQUES

Nous donnons les traits généraux de travaux aujourd'hui classiques d'où dérivent presque toutes les recherches actuelles.

I-2-1 - Matrice de diodes - Réseau microcellulaire - Espaliers

Tous ces noms désignent une même structure qui semble avoir un nouvel intérêt avec l'apparition des transistors MOS.

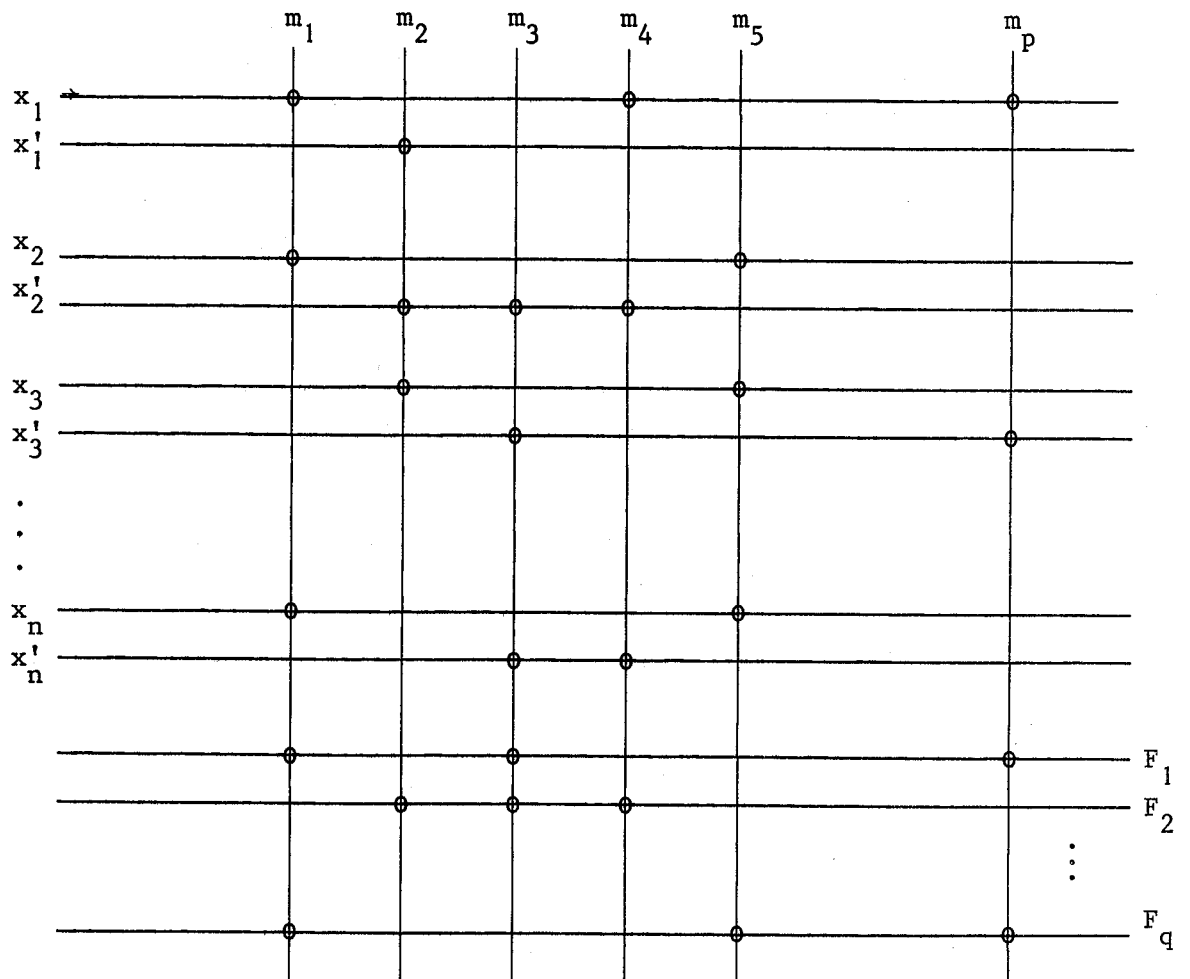
Symboliquement nous pouvons la représenter par un ensemble de lignes et colonnes. A chacune des intersections figure une diode ou un transistor qui sera utilisé ( o ) ou non utilisé ( + ).

Nous pourrions ainsi réaliser la synthèse de q fonctions simples

$F_i(x_1, \dots, x_n)$  exprimées comme sommes d'un certain nombre de p monômes.

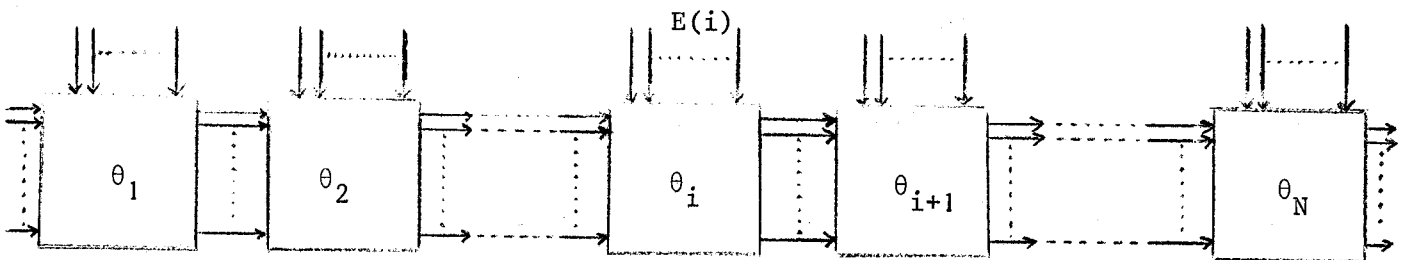
$m_j \in$  ensemble des  $3^n$  monômes de n variables.

Il existe des algorithmes permettant de minimiser le nombre de colonnes ;  
ou la taille globale du réseau.



### I-2-2 - Les réseaux unidimensionnels - Structures en ligne

Ces réseaux sont formés de  $N$  opérateurs situés sur une ligne, c'est-à-dire que les sorties du  $i^{\text{ème}}$  ne peuvent attaquer que les entrées du  $(i+1)$ -ème opérateur, à l'exclusion de tout autre ( $i = 1, \dots, N-1$ ).



L'opérateur  $\theta_i$  dispose de

- \*  $E(i)$  entrées extérieures
- \*  $S(i-1)$  entrées internes au réseau.

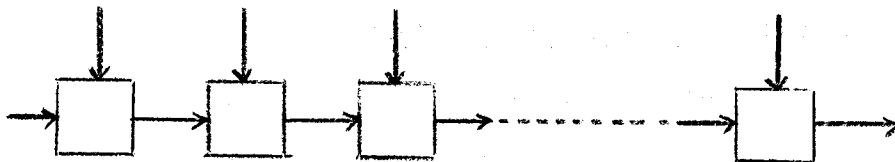
La (ou les) sortie(s) de l'opérateur  $\theta_N$  élabore(nt) la fonction simple (ou générale) désirée.

#### I-2-2-1 - Les cascades dites de Maitra

Avec  $E(i) = 1 \quad \forall i = 1, \dots, N$

et  $S(j) = 1 \quad \forall j = 0, 1, \dots, N$

on obtient les cascades les plus simples



Chaque opérateur peut élaborer un certain nombre des seize fonctions de deux variables (cinq généralement).

Ces cascades ont fait l'objet de très nombreuses études ces vingt dernières années.

Elles offrent une structure simple mais ne réalisant qu'un faible nombre  $R_n$  des fonctions de  $n$  variables, nombre qui va en s'évanouissant avec  $n$  croissant :

$$R_n = 0,8(3 \times 6^{n-1} + 2).$$

Or il y a  $2^{2n}$  fonctions de  $n$  variables.

Remarque : Si nous faisons  $E(i) = m \quad \forall i = 1, \dots, N$

et  $S(j) = 1 \quad \forall j = 0, 1, 2, \dots, N$

nous aurons  $R_{n,m}$  fonctions de  $n$  variables réalisées :

$$R_{n,m} = 2^I \times A^{n-1-B} \frac{A^{n-1}-1}{A-1}$$

avec

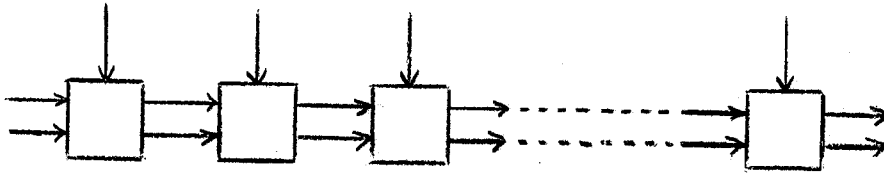
$$\left\{ \begin{array}{l} I = 2^m \\ A = 2^{I-1}(2^I - 1) \\ B = 2^I(2^I - 2). \end{array} \right.$$

Or il y a  $2^{2nm}$  fonctions de  $nm$  variables.

#### I-2-2-2 - Les cascades à deux liaisons artérielles

Elles correspondent à  $E(i) = 1 \quad \forall i = 1, \dots, N$

$S(j) = 2 \quad \forall j = 0, \dots, N$



Chacun des opérateurs peut produire un certain nombre des 65.536 couples de fonctions de trois variables.

Cette structure permet de réaliser toute fonction de  $n$  variables avec un nombre d'opérateurs inférieur ou égal à  $N_n$  avec

$$N_n \leq N_{n-1} + (n-1)2^{n-4}$$

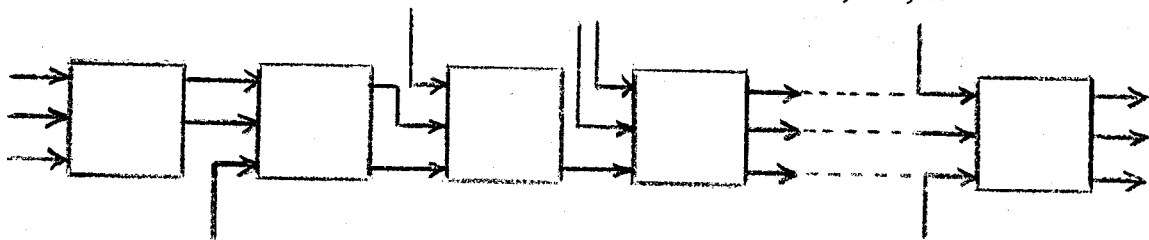
$$N_3 = N_2 = N_1 = 1$$

On a pour les premières valeurs de  $n$  :

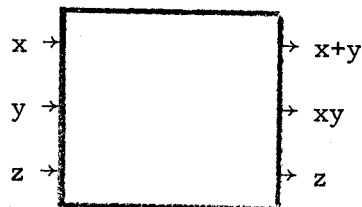
$n$	3	4	5	6	7	8	9	10
$N_n$	1	2	10	30	78	190	446	1022

### I-2-2-3 - Cascades à triples liaisons artérielles

Elles correspondent à  $E(i) + S(i-1) = 3 \quad \forall i = 1, \dots, N.$



Chacun des opérateurs étant identique à l'opérateur ci-dessous





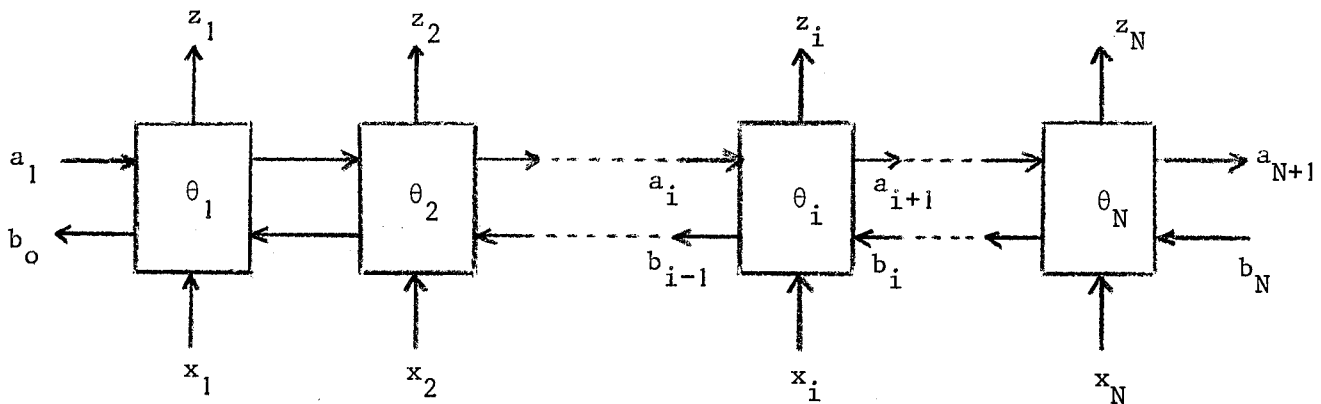
Cette structure permet d'élaborer toute fonction.

A la différence des réalisations précédentes les liaisons n'y sont plus aussi régulières. Par contre toutes les cellules sont identiques.

#### I-2-2-4 - Les réseaux bilatéraux

Dans les réseaux précédents l'information traversait tous les opérateurs, dans un seul sens : de la gauche vers la droite.

Dans les réseaux bilatéraux l'information circule dans les deux sens.



Chaque cellule  $\theta_i$  dispose de 3 entrées  
et 3 sorties

- 1 entrée et une sortie liées à l'extérieur du réseau
- 2 entrées qui sont les sorties des cellules  $\theta_{i-1}$  et  $\theta_{i+1}$
- 2 sorties qui attaquent les cellules  $\theta_{i-1}$  et  $\theta_{i+1}$ .

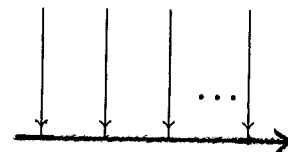
Ces réseaux ont fait l'objet d'études surtout dans le domaine séquentiel.

### I-2-3 - Les réseaux bidimensionnels

Ce sont généralement des extensions dans le plan des réseaux à une seule dimension.

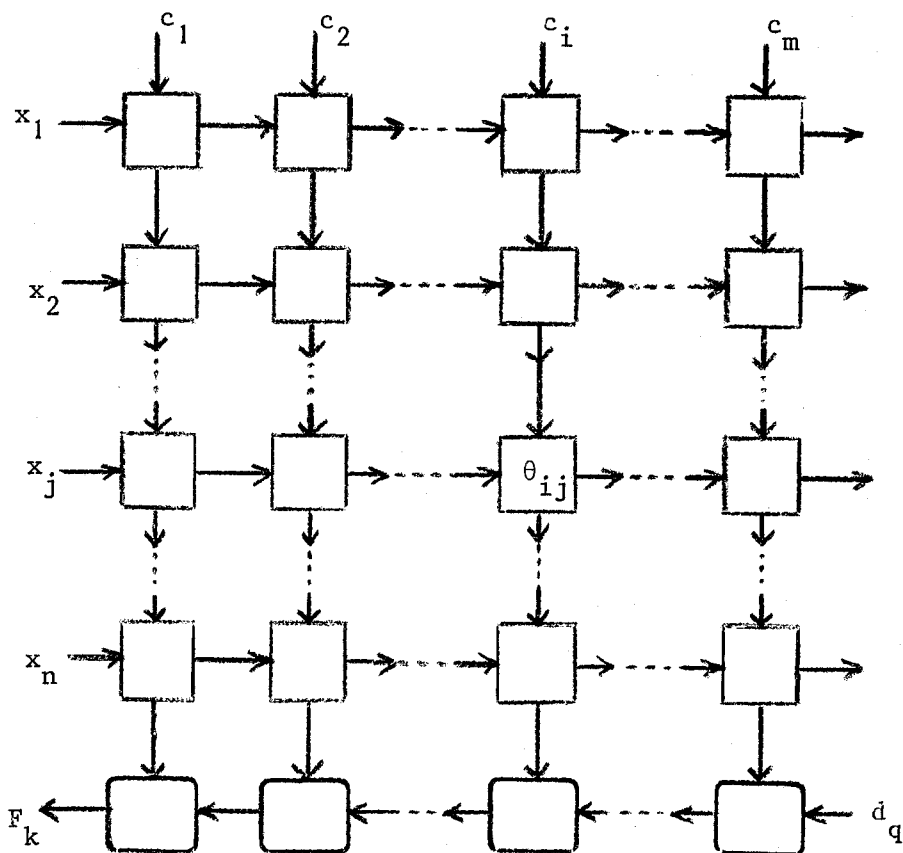
La plupart d'entre eux sont à ranger sous la dénomination "structures en peigne".

Ces structures sont de fausses structures à deux dimensions, elles représentent en fait la juxtaposition de plusieurs structures en lignes qui oeuvrent chacune séparément.



#### I-2-3-1 - Le réseau canonique - Cut-point

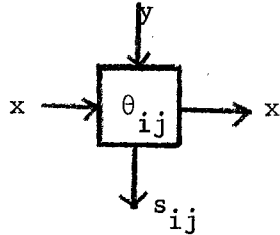
C'est l'extension de la cascade de Maitra



Les opérateurs  $\theta_{ij}$   $i = 1, \dots, m$   
 $j = 1, \dots, n$

n'élaborent en fait la fonction que verticalement.

Leur sortie horizontale sert à traduire les variables d'entrée à l'intérieur du réseau.



avec  $s_{ij}$  une des seize fonction  
(cinq généralement) de deux variables.

Chaque colonne du réseau construit une "composante" de la fonction. Cette composante peut être :

- \* un monôme canonique de la fonction
- \* un de ses monômes premiers
- \* une fonction croissante inférieure à la fonction synthétisée
- \* etc...

Ce réseau peut avoir plus de  $n$  lignes lorsque ses colonnes (ou cascades) ne sont plus irrédondantes (une même variable d'entrée peut attaquer plus d'un opérateur dans une colonne donnée).

Universel (il peut synthétiser toute fonction) ce réseau n'en présente pas moins un gros handicap : son nombre de cellules est élevé ; de plus toutes ne sont pas semblables.

Mais il secrète lui-même son propre défaut : les liaisons horizontales rentent passives au point de vue synthèse.

Remarque : Des réseaux plus généraux ont été étudiés en utilisant des opérateurs ayant  $m$  entrées et sorties verticales ainsi que  $p$  entrées et sorties horizontales.

I-2-3-2 - Les réseaux dérivés de l'arbre dichotomique

Très souvent les synthèses booléennes prennent une forme arborescente.  
Prenons par exemple une synthèse avec l'opérateur Majorité

$$\text{Maj}(x,y,z) = xy+yz+zx$$

Soit  $f(x_1, x_2, \dots, x_n)$  la fonction à synthétiser.

Mettons en évidence la première variable :

$$f(x_1, x_2, \dots, x_n) = x_1 f_1(1, x_2, \dots, x_n) + x_1' f_0(0, x_2, \dots, x_n).$$

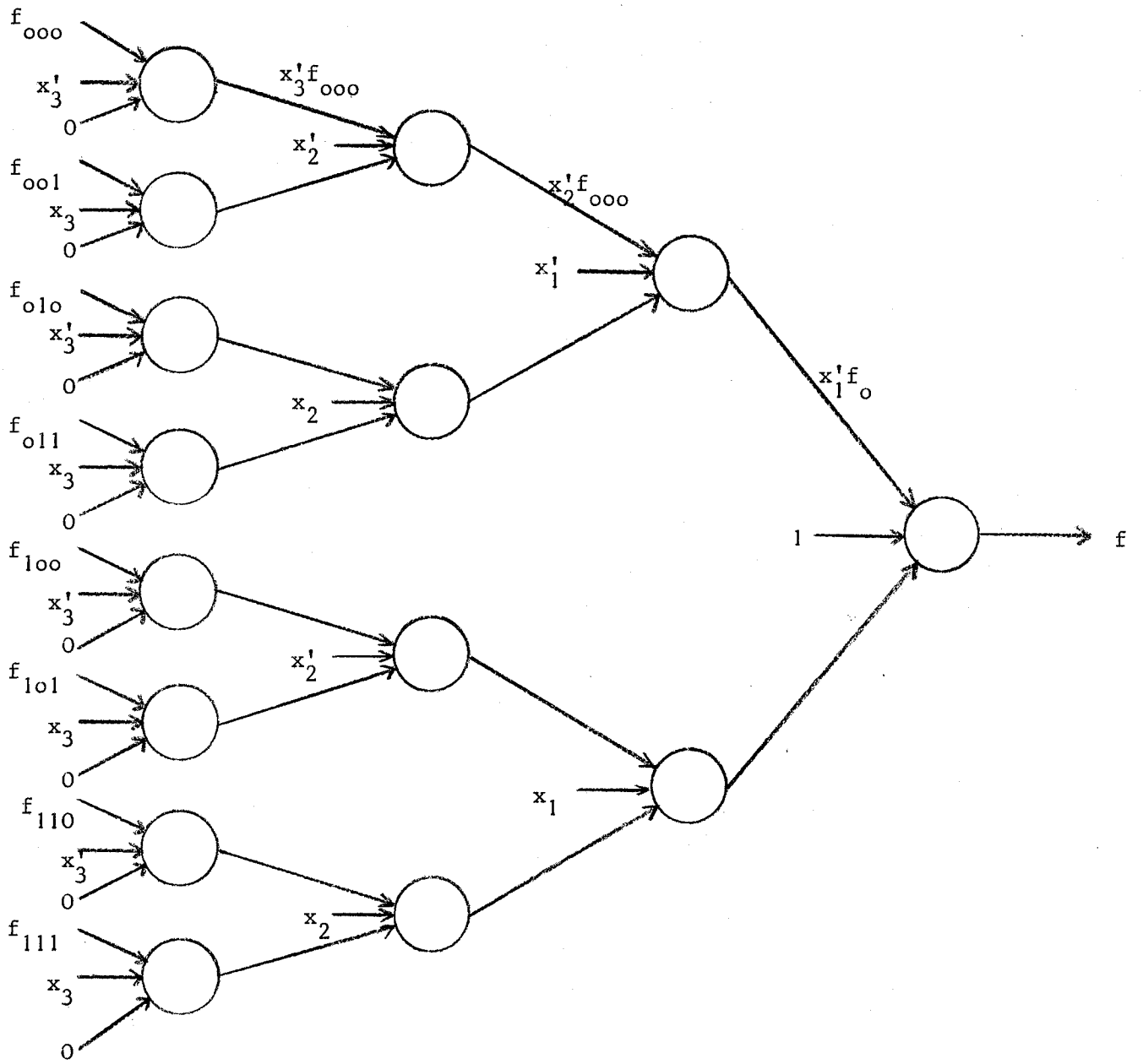
Nous aurons ainsi deux fonctions de  $(n-1)$  variables :

$$f_1(1, x_2, \dots, x_n) = x_2 f_{11}(1, 1, x_3, \dots, x_n) + x_2' f_{10}(1, 1, x_3, \dots, x_n)$$

$$f_0(0, x_2, \dots, x_n) = x_2 f_{01}(0, 1, x_3, \dots, x_n) + x_2' f_{00}(0, 0, x_3, \dots, x_n).$$

Et ainsi de suite pour les  $n$  variables.

Pour  $n = 3$  nous obtiendrons l'arbre de la figure ci-après :



Ce genre de réseaux arborescents permet la synthèse de toute fonction booléenne.

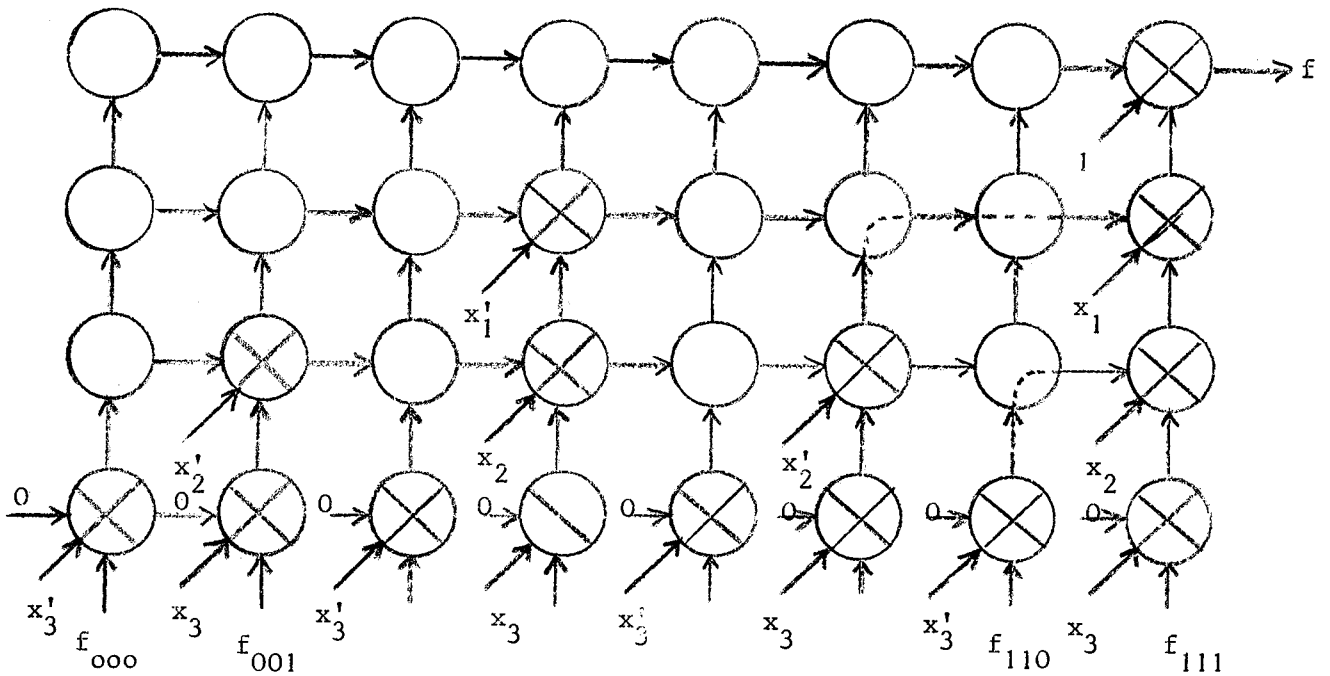
C'est pourquoi on a cherché à placer l'arbre sur un réseau itératif. On aurait ainsi un réseau régulier universel.


Citons quelques immersions de l'arbre lexicographique complet dans des réseaux cellulaires.


I-2-3-2-1 - Equivalence couche de l'arbre - Colonne du réseau

Chaque couche de l'arbre, c'est-à-dire l'ensemble des opérateurs ayant la même variable  $\tilde{x}_i$  d'entrée, est représentée dans une colonne du réseau.

Pour l'exemple précédent, nous obtenons la figure ci-après :



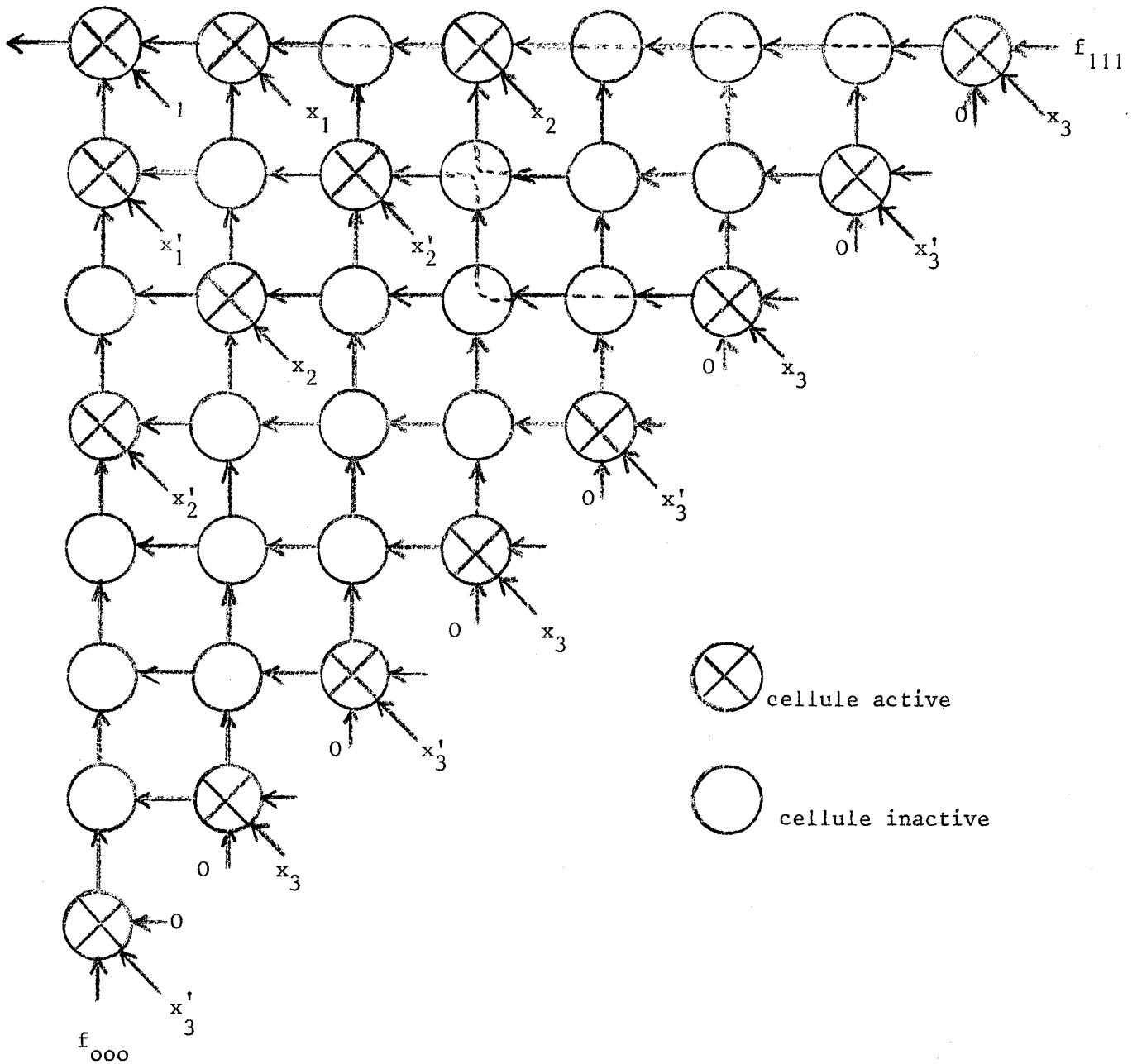
 cellule active

 cellule inactive

I-2-3-2-2 - Equivalence couche de l'arbre - Diagonale du réseau  
.....

On obtient alors le réseau, figure ci-dessous.

Ce réseau comprend  $2^{n-1}(2^n+1)$  cellules et de nombreuses cellules y sont encore inactives.



En conclusion ces immersions donnent des réseaux trop grands et en utilisant mal les cellules. De plus, ils ne sont pas planaires.

I-2-3-3 - Réseaux et classes de fonctions

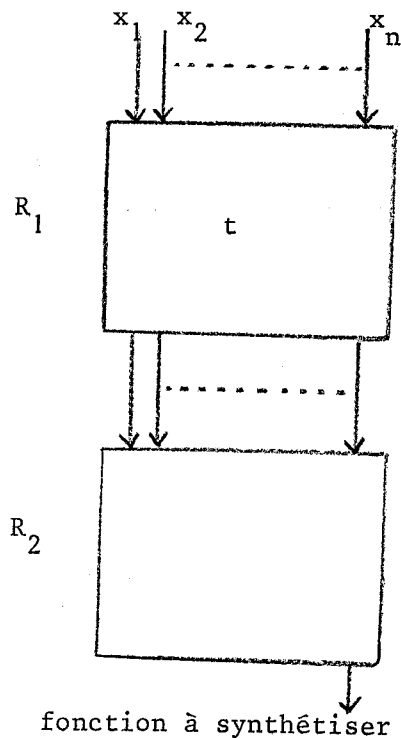
Soit  $F_n$  l'ensemble des fonctions de  $n$  variables, à deux valeurs.  
Sur cet ensemble on fait agir un groupe  $T$  de transformations.  
Ce groupe forme un certain nombre de classes d'équivalences  
 $E_i$  ( $i = 1, \dots, p$ ) qui constituent une partition de  $F_n$  :

deux fonctions  $f_j, f_k \in F_n$  sont équivalentes s'il existe une transformation  $t \in T$  telle que l'on ait :

$$f_j = t(f_k).$$

La synthèse se traduit alors par deux réseaux  $R_1$  et  $R_2$  :

- \* le réseau  $R_1$  synthétise un représentant déterminé de la classe d'équivalence contenant la fonction à synthétiser.
- \* le réseau  $R_2$  réalise la transformation  $t$  qui fait passer du représentant à la fonction désirée.





Le but de telles méthodes est d'obtenir un nombre de classes d'équivalence suffisamment petit pour n'avoir qu'un nombre restreint de réseaux standards  $R_2$  à fabriquer.

Malheureusement, quelque soit le groupe T choisi, le nombre de réseaux standards ne reste abordable (50) que jusqu'à cinq variables. Ensuite il croît très vite.

Groupe T	Nombre de classes d'équivalences pour					
	n=1	2	3	4	5	6
$T_1$ complémentations des variables	3	7	46	4336	134281.216	
$T_2$ permutations des variables	4	12	80	3984	37.333.248	$>25.10^{15}$
$T_3 = T_1 \cdot T_2$ (symétries de l'hypercube)	3	6	22	402	1228158	$>4.10^{14}$
$T_4 =$ groupe linéaire	4	8	20	92	2744	950998216
$T_5 =$ groupe affine	3	5	10	32	382	15.768.919
$T_6 = T_5$ (complémentation sur la fonction)	2	3	6	18	206	7888299
$T_7 =$ RAG (cf. LECHNER)	1	2	3	8	48	$>69.000$

CHAPITRE - II

CARACTERES GENERAUX DES SYNTHESES PROPOSEES

II-1 - LE POINT DE DEPART

Nous avons cherché à améliorer certaines directions de recherche, et plus spécialement, ce qui avait été réalisé dans le domaine de la représentation de l'arbre sur un réseau cellulaire régulier.

Un deuxième axe de travail, défini par Monsieur Kuntzmann, est basé sur le théorème d'amélioration concernant la synthèse de fonctions croissantes au moyen d'opérateurs croissants. Ce qui permet de synthétiser toute fonction booléenne.

Dans une première partie, nous ne traiterons que le cas de l'arbre lexicographique.

II-2 - AVANTAGES DES SYNTHESES ARBORESCENTES

Les synthèses arborescentes semblent les plus naturelles, la fonction est réalisée par construction ou destruction en  $n$  étapes, chacune des étapes correspondant à une des variables.

Les algorithmes sont généralement assez simples.

UNIVERSITÉ DE GRENOBLE  
SERVICE DE MATHÉMATIQUES  
APPLIQUÉES  
POLYCOPIES  
CEDEX n° 53  
38 - GRENOBLE - GARE

II-3 - LES RESEAUX DERIVES DE L'ARBRE

Pour les méthodes de synthèse proposées ici nous utiliserons donc des réseaux cellulaires dérivés de l'arbre dont ils conservent nombre de propriétés :

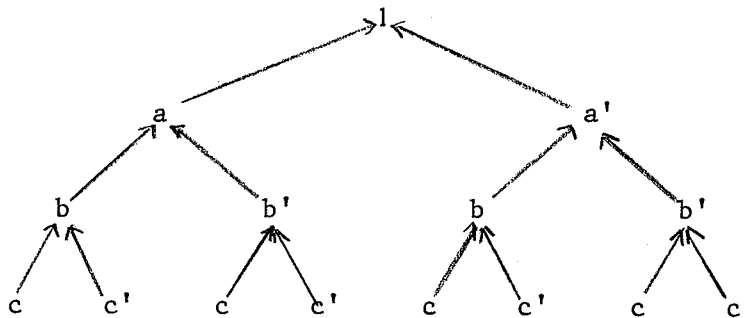
- \* ils comportent les mêmes  $2^n$  chemins,
- \* chaque sommet (ou arête) correspondant à la variable  $\tilde{x}_i$  est relié(e) aux deux sommets (arêtes) correspondant à  $\tilde{x}_{i+1}$  et  $\tilde{x}'_{i+1}$ ,
- \* ils sont planaires.

Nous donnons ici une représentation symbolique de ces réseaux, faisant ressortir les caractéristiques principales.

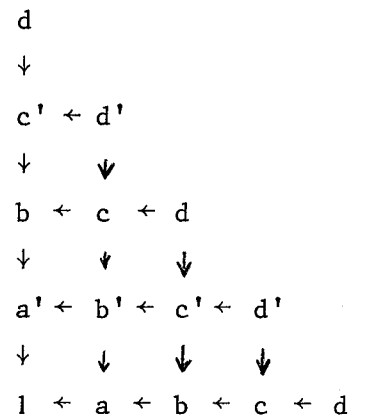
II-3-1 - Le réseau



Chaque couche de l'arbre correspond à une diagonale du réseau qui comporte ainsi (n+1) couches.



arbre lexicographique

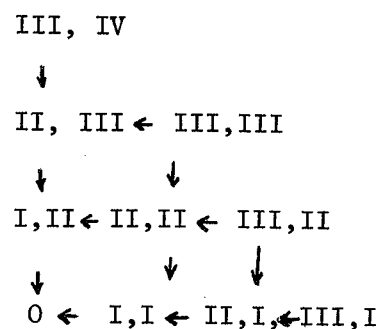
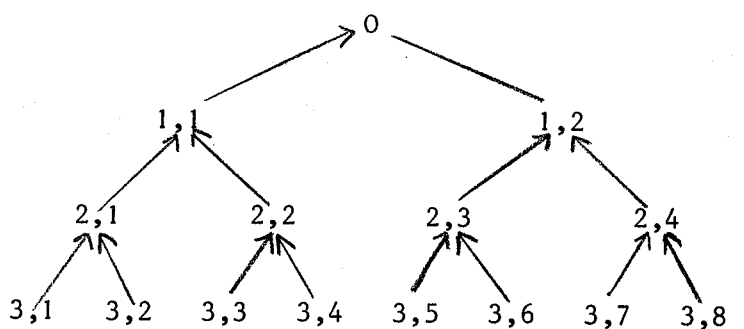


réseau



Nous ne représenterons pas toujours les arcs qui précisent la circulation dans le réseau

Il n'y a que surjection entre les cellules de l'arbre et celles du réseau  $\triangle$ . Reprenons l'exemple précédent :



1,1	→	I, I
1,2	→	I, II
2,1	→	II, I
2,2	}	→ II, II
2,4		
2,3	→	II, III

3,1	→	III, I
3,2	}	→ III, II
3,4		
3,8		
3,3	}	→ III, III
3,7		
3,5		
3,6	→	III, IV

L'arbre comportait  $(2^{n+1}-1)$  cellules.

Le réseau  $\triangle$  en contient  $(n+1)(n+2)/2$ .

On peut définir un "coefficient de contraction" de l'arbre par

$$C_{\triangle} = \frac{(n+1)(n+2)}{2(2^{n+1}-1)}$$

Comme pour l'arbre nous aurons des :

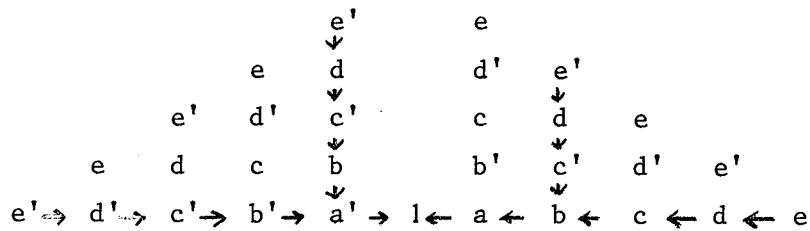
- \* feuilles : sommets n'ayant aucun prédécesseur
- \* racine : sommet n'ayant aucun successeur.

II-3-2 - Réseau



Ce réseau est symbolisé par un triangle isocèle.

Chaque couche de l'arbre y correspond à deux parallèles aux côtés adjacents à la base.



Ce réseau comporte  $n(n+1)+1$  cellules.

Il y a toujours surjection entre les cellules de l'arbre et celles du réseau :

1,1	→	I,I		3,1	→	III,I
1,2	→	I,II		3,2	} →	III,II
				3,4		
2,1	→	II,I		3,3	→	III,III
2,2	→	II,II		3,6	→	III,IV
2,3	→	II,III		3,5	} →	III,V
				3,7		
2,4	→	II,IV		3,8	→	III,VI

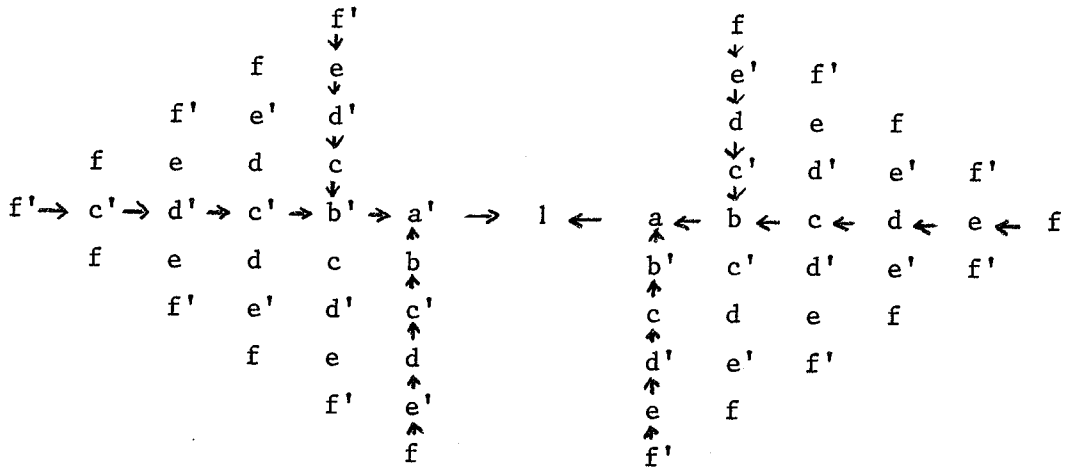
avec :

			III, IV		III, III			
		III, V	II, III		II, II	III, II		
III, VI	II, IV	I, II	0	I, I	II, I	III, I		

III-3-3 - Le réseau

Ce réseau a la forme d'un losange (carré) et chaque couche de l'arbre y correspond à une ligne parallèle au périmètre.

Ce réseau comprend  $(2n^2 - 2n + 3)$  cellules




Nous donnons ci-dessous quelques valeurs des coefficients de contraction pour différents nombres (n) de variables.

n	0	1	2	3	4	5	6	7	8	9	10
100 C	100	100	86	67	49	33	22	14	8,7	5,4	3,2
100 C	100	100	100	87	68	49	34	22	14	8,9	5,4
100 C	100	100	100	100	87	68	49	34	22	14	9



CHAPITRE - III

PROBLEMES POSES PAR LA SYNTHÈSE SUR LES RESEAUX PRECEDENTS

Dans la suite nous supposons que le réseau utilisé est le réseau  Les problèmes sont en effet analogues dans les trois réseaux, au coefficient de contraction près.

III-1 - ECRITURE DES FONCTIONS

Nous considérons les fonctions booléennes de n variables données par leur première ou leur seconde forme de Lagrange :

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^{i=p} \mu_i = \prod_{j=1}^{j=p} \nu_j$$

avec

$$\mu_i = \tilde{x}_1 \cdot \tilde{x}_2 \cdot \dots \cdot \tilde{x}_n$$

$$\nu_j = \tilde{x}_1 + \tilde{x}_2 + \dots + \tilde{x}_n$$

où

$$\tilde{x}_k = \begin{cases} x_k \\ \text{où} \\ x'_k \end{cases}$$

Nous verrons plus loin qu'une écriture lexicographique peut suffire.

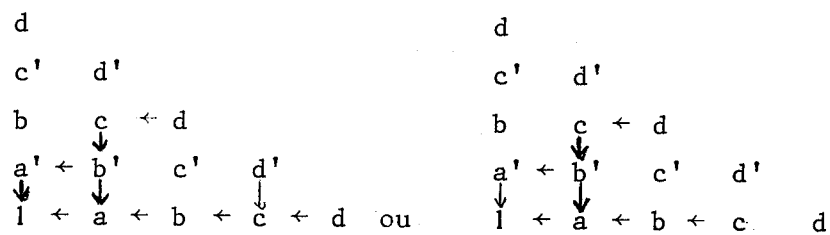
Il y a bijection entre les  $2^n$   $\mu_i$  (ou  $\nu_j$ ) possibles et les  $2^n$  chemins du réseau.



III-2 - MECANISME DE LA SYNTHESE

Synthétiser la fonction F sur le réseau en question revient à "activer" par certains opérateurs définis plus loin, les p (ou q) chemins du réseau qui correspondent aux p monômes  $\mu_i$  (ou q  $\nu_j$ ).

Exemple :  $F_1(a,b,c,d) = abcd + abcd' + a'b'cd + ab'cd$



III-3 - COMPATIBILITE

Le réseau a été obtenu par une contraction de l'arbre, aussi ne remplit-il plus toutes les possibilités offertes par l'arbre. En particulier :

les  $2^n$  chemins ne sont plus indépendants.

III-3-1 - Intersection de deux chemins

Soit deux chemins  $C_i$  et  $C_j$  :

$$C_i = \{1, x_1^{i1}, \dots, x_n^i\}$$

$$C_j = \{1, x_1^{j1}, \dots, x_n^{jn}\}$$

avec  $i_k, j_k \in \{0,1\} \quad \forall k = 1, \dots, n$   
 et

$$\begin{cases} x^1 = x \\ x^0 = x' \end{cases}$$

En plus de la racine du réseau, ces deux chemins peuvent avoir un sommet commun  $x_k^{i_k} \equiv x_k^{j_k}$ . Nous appellerons alors :

$$t_{ik} = \{1, x_1^{i_1}, \dots, x_{k-1}^{i_{k-1}}\}$$

$$t_{jk} = \{1, x_1^{j_1}, \dots, x_{k-1}^{j_{k-1}}\}$$

les TETES de ces chemins par rapport au sommet commun et

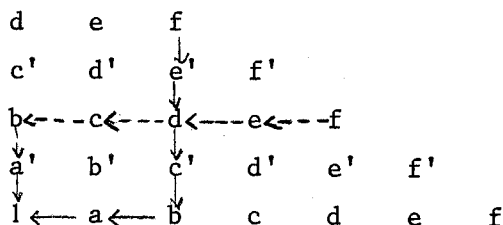
$$q_{ik} = \{x_k^{i_k}, \dots, x_n^{i_n}\}$$

$$q_{jk} = \{x_k^{j_k}, \dots, x_n^{j_n}\}$$

les QUEUES de ces chemins.

Exemple :  $C_1 = \{1, a, b, c', d, e', f\}$

$C_2 = \{1, a', b, c, d, e, f\}$



$$\left. \begin{aligned} x_k^{ik} &= d \\ t_{1d} &= \{1, a, b, c'\} \\ t_{2d} &= \{1, a', b, c\} \\ q_{1d} &= \{e', f\} \\ q_{2d} &= \{e, f\} \end{aligned} \right\}$$

III-3-2 - Compatibilité de deux monômes au sens strict

Sur l'ensemble des  $2^n$  monômes nous définissons une relation binaire de compatibilité  $C$  :

DEFINITION

Deux monômes  $\mu_i$  et  $\mu_j$  sont dits compatibles au sens strict si et seulement si nous sommes dans une des trois configurations suivantes :

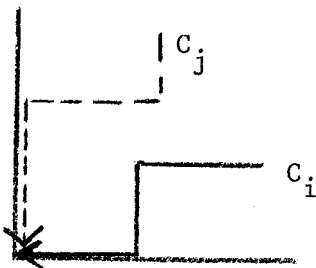
les deux chemins  $C_i$  et  $C_j$  qui les représentent sur le réseau

1 - n'ont d'autre intersection que la racine  $C_i \wedge C_j = \{1\}$

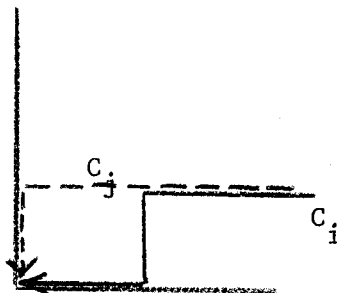
2 - ont même tête :  $t_{ik} = t_{jk} = C_i \wedge C_j$

3 - ont même queue :  $q_{ik} = q_{jk} = C_i \wedge C_j$

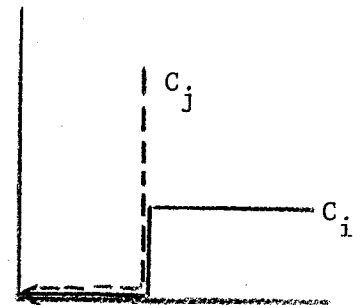
Nous écrirons :  $\mu_i C \mu_j$



configuration 1



configuration 3



configuration 2

Dans le cas contraire, c'est-à-dire si l'on a

$$C_i \wedge C_j \neq \{1\}$$

et

$$t_{ik} \neq t_{jk}$$

et

$$q_{ik} \neq q_{jk}$$

les deux monômes  $\mu_i$  et  $\mu_j$  seront dit incompatibles

$$\mu_i \not\subset \mu_j$$

Exemple Reprenons les deux monômes du paragraphe III-3-1

$$\mu_1 = abc'de'f$$

$$\mu_2 = a'bcdef$$

ils sont incompatibles car nous avons

$$\left\{ \begin{array}{l} C_1 \wedge C_2 = \{d\} \neq \{1\} \\ t_{1d} \neq t_{2d} \\ q_{1d} \neq q_{2d} \end{array} \right.$$

L'union des deux chemins  $C_1 \vee C_2$  représente 4 monômes :

$$\mu_1, \mu_2$$

$$\mu_3 = abc'def$$

$$\mu_4 = a'bcd'e'f.$$

Cette relation de compatibilité  $C$  est :

\* réflexive  $\mu_i C \mu_i$  car :

$$C_i \wedge C_i = C_i \neq \{1\}$$

mais  $q_i = \{\emptyset\}$  l'ensemble vide et  $t_i = t_i = C_i$

\* symétrique  $\mu_i C \mu_j \Rightarrow \mu_j C \mu_i$ .

Elle n'est pas transitive par contre :

$$\mu_i C \mu_j$$

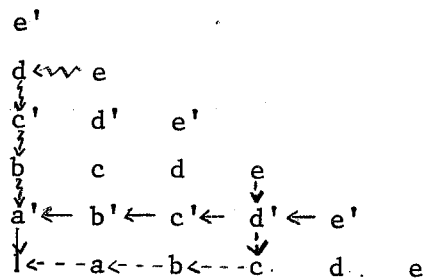
et

$$\mu_j C \mu_k$$

n'entraînent pas obligatoirement  $\mu_i C \mu_k$

Exemple

$$\left\{ \begin{array}{l} \mu_1 = a'b'c'd'e' \\ \mu_2 = a'bc'de \\ \mu_3 = abcd'e \end{array} \right.$$



Nous avons :

$$\mu_1 C \mu_2 \text{ car } C_1 \wedge C_2 = \{1, a'\} = q_{1a} = q_{2a}$$

$$\mu_2 C \mu_3 \text{ car } C_2 \wedge C_3 = \{1\}$$

Mais :

$$C_1 \wedge C_3 = \{1, d'\}$$

$$(t_{1d} = \{1, a', b', c'\}) \neq (t_{2d} = \{1, a, b, c\})$$

$$(q_{1d} = \{e\}) \neq (q_{2d} = \{e'\})$$

et par suite nous n'avons pas  $\mu_1 \subset \mu_3$ .

Cette non-transitivité nous obligera à étudier toutes les compatibilités deux à deux.

### III-3-3 - Conséquences pour la synthèse

Comme il y a bijection entre les monômes d'une fonction et les chemins du réseau, on comprend aisément que certaines fonctions ne soient pas synthétisables puisqu'il n'y a pas toujours compatibilité entre les différents monômes.

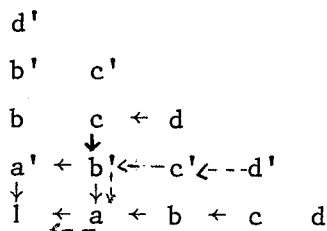
En reprenant la fonction synthétisable du paragraphe III-2

$$F_1(a, b, c, d) = abcd + abcd' + a'b'cd + ab'cd$$

et en lui ajoutant le monôme  $(ab'c'd')$  nous obtenons une nouvelle fonction

$$F_2(a, b, c, d) = F_1(a, b, c, d) + ab'c'd'$$

qui n'est plus synthétisable car le monôme  $(ab'c'd')$  n'est pas compatible avec les monômes  $(a'b'cd)$  et  $(ab'cd)$ .



III-4 - ROLE DE L'ORDRE DES VARIABLES

Nous avons perdu de nombreuses possibilités en contractant l'arbre, mais un nouveau phénomène auquel l'arbre était étranger, va jouer un rôle important dans notre réseau : l'ordre des variables.

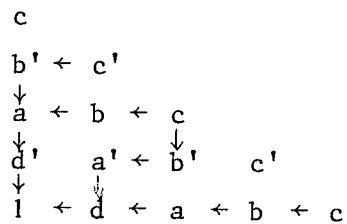
Dans l'exemple ci-dessus la première ligne du réseau est représentée par

(labcd)

Ceci constitue en fait un choix arbitraire de représentation du réseau.

Comme il y a bijection entre l'ensemble des réseaux et l'ensemble de leurs premières lignes, nous représenterons symboliquement un réseau par la configuration de sa première ligne.

La fonction  $F_2$  de l'exemple précédent n'est pas synthétisable, on l'a vu, sur le réseau (labcd). Elle l'est par contre dans le réseau (ldabc) comme le montre la figure ci-dessous :



Le nouvel ordre des variables qui figure dans la première ligne a ici levé toutes les incompatibilités.

Il faut donc souligner que la notion de compatibilité est liée à l'ordre des variables :

on avait  $(ab'c'd) \alpha_{1abcd} (a'b'cd)$

mais on a  $(ab'c'd') C_{1dabc} (a'b'cd)$ .

On conçoit ainsi qu'en jouant sur l'ordre des variables on pourra empêcher un grand nombre de possibilités perdues en contractant l'arbre.

### III-5 - COMPATIBILITE DE DEUX MONOMES RELATIVEMENT A UN ENSEMBLE DE MONOMES

DEFINITION 2 Deux monômes  $\mu_i$  et  $\mu_j$  appartenant à un ensemble  $E = \{\mu_k / k = 1, \dots, p\}$  seront compatibles relativement à cet ensemble  $E$  si et seulement si l'ensemble des chemins formés par l'union  $C_i \vee C_j$  des deux chemins associés à  $\mu_i$  et  $\mu_j$  représente un sous-ensemble de  $E$ .

Nous écrirons  $\mu_i C_E \mu_j$ .

Remarquons au passage que la compatibilité de deux monômes au sens strict est un cas particulier de la définition 2 :

cas où l'on a  $E = \{\mu_i, \mu_j\}$ .

#### III-5-1 - Cas de deux chemins sécants en $\tilde{x}_k$

Soit

$$C_1 = t_{1k} \vee \{\tilde{x}_k\} \vee q_{1k}$$

$$C_2 = t_{2k} \vee \{\tilde{x}_k\} \vee q_{2k}$$



$C_1$  et  $C_2$  représentent les deux monômes

$$\mu_1 = T_{1k} \cdot \tilde{x}_k \cdot Q_{1k}$$

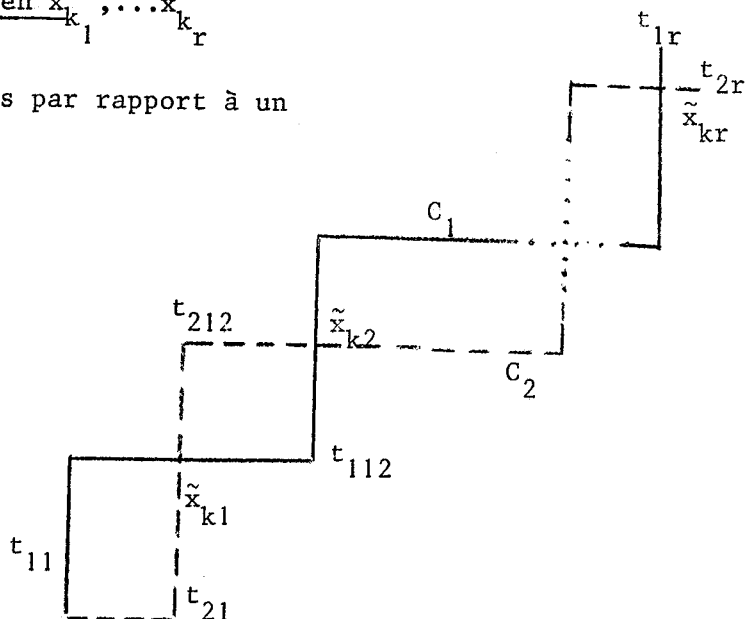
$$\mu_2 = T_{2k} \cdot \tilde{x}_k \cdot Q_{2k}$$

$\mu_1$  et  $\mu_2$  seront compatibles par rapport à l'ensemble

$$E = \{\mu_1, \mu_2, T_{1k} \cdot \tilde{x}_k \cdot Q_{2k}, T_{2k} \cdot \tilde{x}_k \cdot Q_{1k}\}.$$

III-5-2 - Cas de deux chemins sécants en  $\tilde{x}_{k_1}, \dots, \tilde{x}_{k_r}$

$\mu_1$  et  $\mu_2$  seront ici compatibles par rapport à un ensemble  $E$  de  $2^r$  monômes



$$E = \{T_{i_1 1} \cdot \tilde{x}_{k_1} \cdot T_{i_2 2} \cdot \tilde{x}_{k_2} \cdot \dots \cdot \tilde{x}_{k_r} \cdot T_{i_r r} / i_j \in \{1, 2\}\}.$$

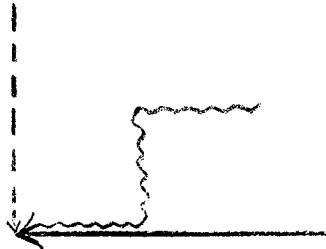
III-6 - CAS DE TROIS MONOMES

Le problème de la compatibilité se complique lorsque l'on considère plus de deux monômes.

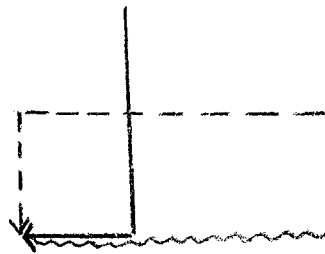
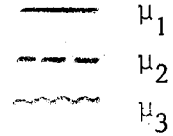
Etudions à titre d'exemple, le cas de trois monômes  $\mu_1, \mu_2, \mu_3$ .

Ils ont quatre configurations possibles :

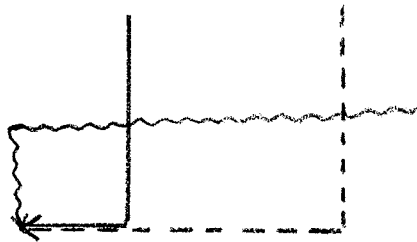
\* configuration 1 : aucune intersection



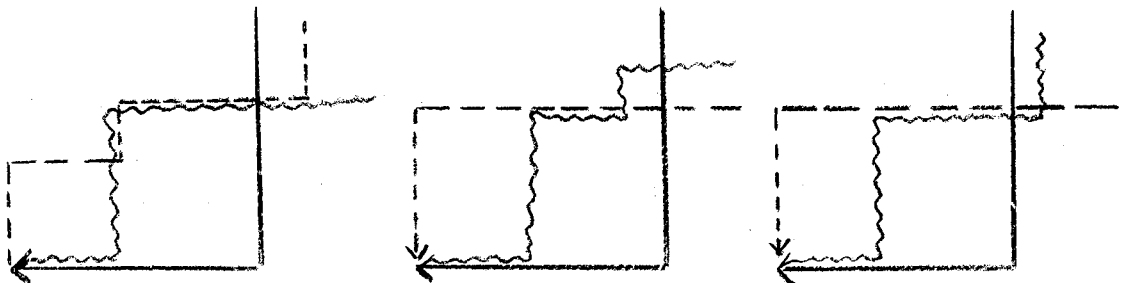
\* configuration 2 :  $\mu_1$  et  $\mu_2$  sécants



\* configuration 3 :  $\mu_1$  et  $\mu_2$  non sécants coupent  $\mu_3$



\* configuration 4 :  $\mu_1, \mu_2, \mu_3$  tous sécants deux à deux.



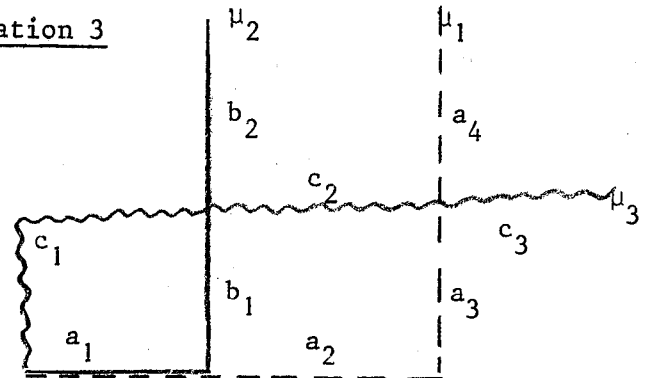
Les deux premières configurations n'offrent que des cas dégénérés. Seuls les deux dernières présentent des nouveautés.

III-6-1 - Examinons le cas de la configuration 3

$$\mu_1 = a_1 \cdot a_2 \cdot a_3 \cdot a_4$$

$$\mu_2 = a_1 \cdot b_1 \cdot b_2$$

$$\mu_3 = c_1 \cdot c_2 \cdot c_3$$



\* Nous avons  $\mu_1 \subset \mu_2$

\*  $\mu_1$  et  $\mu_3$  seront compatibles par rapport à l'ensemble

$$E_{13} = \{\mu_1, \mu_3, \mu_4 = a_1 \cdot a_2 \cdot a_3 \cdot c_3, \mu_5 = c_1 \cdot c_2 \cdot a_4\}$$

\*  $\mu_2$  et  $\mu_3$  le seront par rapport à l'ensemble

$$E_{23} = \{\mu_2, \mu_3, \mu_6 = c_1 \cdot b_2, \mu_7 = a_1 \cdot b_1 \cdot c_2 \cdot c_3\}$$

En prenant  $E = E_{13} \vee E_{23}$  nous aurons

$$\mu_i \subset_E \mu_j \quad \forall i, j = 1, 2, 3.$$

Pourtant, bien que compatibles deux à deux, ces trois monômes ne forment pas un ensemble compatible. L'union des chemins les représentant contient le chemin associé au monôme

$$\mu_8 = a_1 \cdot b_1 \cdot c_2 \cdot a_4$$

monôme qui n'appartient pas à E.

Est-ce à dire que la condition de compatibilité deux à deux pour  $p$  monômes n'implique pas la compatibilité globale des  $p$  monômes.

On pourrait le penser en constatant que  $\mu_1, \mu_2, \mu_3$  rendus compatibles deux à deux relativement à  $E$ , ne forment pas un ensemble compatible.

Mais ce serait oublier que pour rendre  $\mu_1, \mu_2$  et  $\mu_3$  compatibles nous avons introduit certains monômes dont nous n'avons pas étudié la compatibilité deux à deux relativement à  $E$ .

Nous avons effectivement

$$\left\{ \begin{array}{l} \mu_1 \not\subset_E \mu_7 \\ \mu_2 \not\subset_E \mu_5 \\ \mu_5 \not\subset_E \mu_7 \end{array} \right.$$

Pour lever ces incompatibilités il faut considérer l'ensemble

$$E' = E \vee \{\mu_8\}$$

Nous aurons alors

$$\mu_i \subset_{E'} \mu_j \quad \forall i, j = 1, \dots, 8.$$

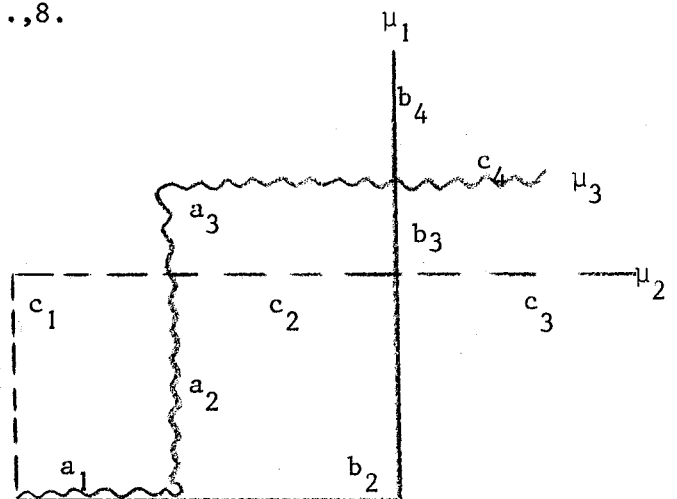
#### III-6-2 - Pour la configuration 4

Le même phénomène se reproduit

$$\mu_1 = a_1 \cdot b_2 \cdot b_3 \cdot b_4$$

$$\mu_2 = a_1 \cdot a_2 \cdot a_3 \cdot a_4$$

$$\mu_3 = c_1 \cdot c_2 \cdot c_3$$



Pour avoir  $\mu_i \underset{E}{C} \mu_j \quad \forall i, j = 1, 2, 3$  il faut considérer l'ensemble  $E = \{\mu_k / k = 1, \dots, 9\}$  avec

$$\begin{cases} \mu_4 = c_1 \cdot c_2 \cdot b_3 \cdot b_4 \\ \mu_5 = a_1 \cdot b_2 \cdot c_3 \end{cases} \quad \begin{cases} \mu_6 = c_1 \cdot a_3 \cdot a_4 \\ \mu_7 = a_1 \cdot a_2 \cdot c_2 \cdot c_3 \end{cases}$$

$$\begin{cases} \mu_8 = a_1 \cdot a_2 \cdot a_3 \cdot b_4 \\ \mu_9 = a_1 \cdot b_2 \cdot b_3 \cdot a_4 \end{cases}$$

Mais E ne forme pas pourtant un ensemble de monômes compatibles deux à deux. Nous aurons :

$$\mu_1 \underset{E}{\not C} \mu_6 \quad \mu_1 \underset{E}{\not C} \mu_7 \quad \mu_2 \underset{E}{\not C} \mu_4 \text{ etc...}$$

### III-7 - CONSTELLATIONS

#### DEFINITION 3

*p monômes  $\mu_i$  seront dits compatibles globalement si et seulement si l'ensemble de ces p monômes est identique à l'ensemble des monômes représentés par l'union des p chemins  $C_i$ .  
Un tel ensemble sera appelé "constellation".*

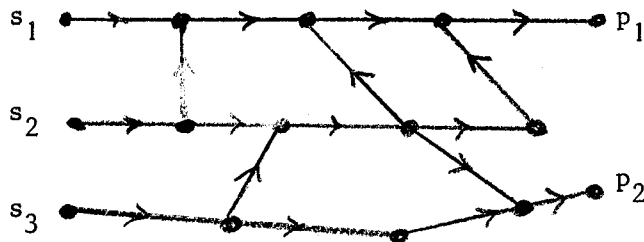
#### Remarque -

Plus généralement on définit une constellation sur un graphe orienté comportant

\* des sources : sommets n'ayant aucun prédécesseur

\* des puits : sommets n'ayant aucun successeur

comme étant l'ensemble de tous les chemins allant de certaine(s) source(s) à certain(s) puit(s).



Une constellation sera dite maximale si elle n'est contenue dans aucune autre

#### THEOREME

*CNS de compatibilité :*

*Pour que p monômes forment une constellation il faut et il suffit qu'ils soient compatibles deux à deux relativement à l'ensemble des p monômes.*

*Cette condition est nécessaire :*

si les p monômes forment une constellation, ils sont automatiquement compatibles deux à deux puisque les chemins associés à ces deux monômes sont inclus dans les chemins représentant les p monômes.

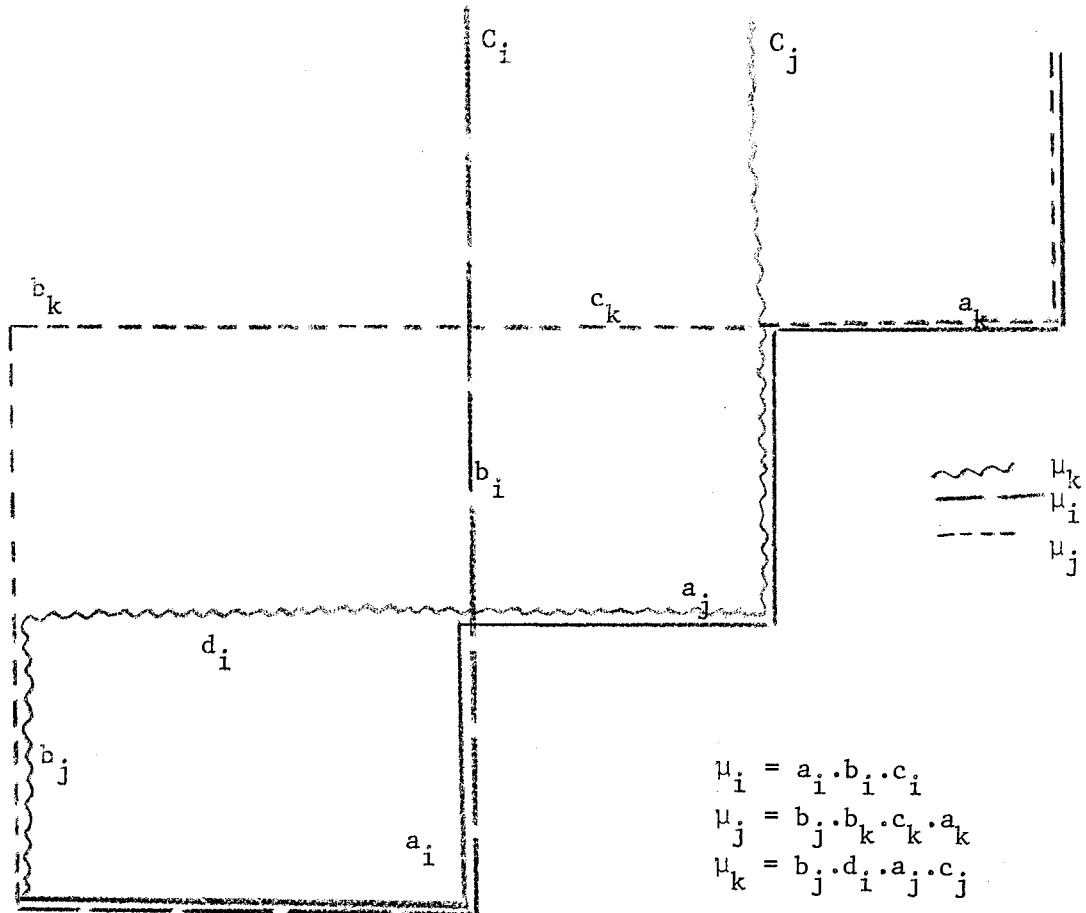
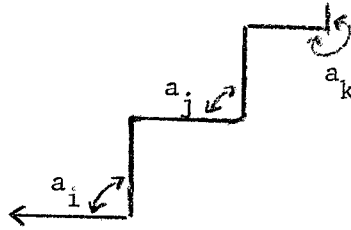
*La condition est suffisante :*

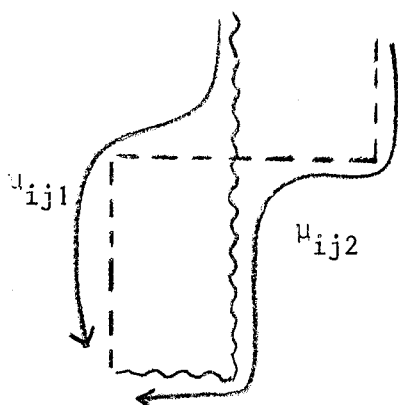
supposons les p monômes compatibles deux à deux et supposons qu'au moins un des chemins représentatifs n'a pas son image dans l'ensemble des p monômes.

Ce chemin est composé de sous-chemins appartenant à au moins trois monômes  $\mu_i, \mu_j, \mu_k$  différents, car s'il n'était composé que de sous-chemins appartenant à deux monômes différents seulement, cela signifierait que deux monômes ne sont pas compatibles, ce qui est contraire à l'hypothèse.

Soit  $(a_i \cdot a_j \cdot a_k)$  le monôme n'appartenant pas à l'ensemble des  $p$  monômes :

$$\begin{aligned} a_i &\in \mu_i \\ a_j &\in \mu_j \\ a_k &\in \mu_k \end{aligned}$$





$\mu_i$  et  $\mu_j$  étant compatibles, les  $p$  monômes devront inclure

$$\begin{cases} \mu_{ij1} = b_j \cdot b_k \cdot c_i \\ \mu_{ij2} = a_i \cdot b_i \cdot c_k \cdot a_k \end{cases}$$

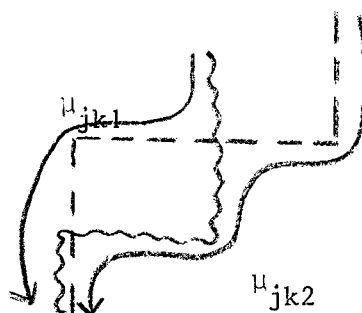
Pour  $\mu_j$  et  $\mu_k$  nous devons avoir

$$\begin{cases} \mu_{jk1} = b_j \cdot b_k \cdot c_k \cdot c_j \\ \mu_{jk2} = b_j \cdot d_i \cdot a_j \cdot a_k \end{cases}$$

A ce stade les  $p$  monômes comprennent

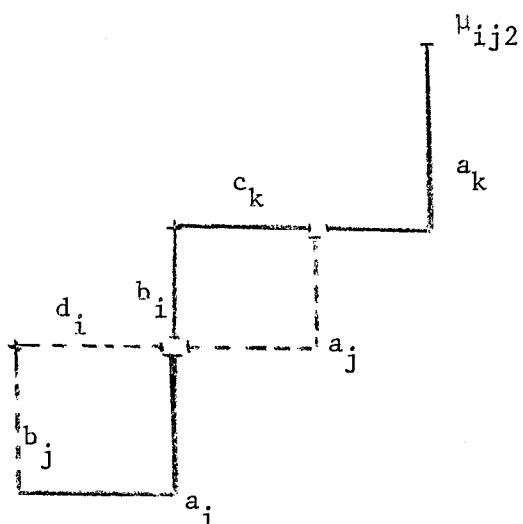
$$\mu_i, \mu_j, \mu_k, \mu_{ij2}, \mu_{jk2}$$

et les deux derniers doivent en particulier être compatibles d'après l'hypothèse.



Or  $\mu_{ij1} \stackrel{C}{\sim} \mu_{jk2}$  entraîne l'existence du monôme :

$a_i \cdot a_j \cdot a_k$  parmi les  $p$  monômes, ce que nous avons supposé faux au départ de cette démonstration.



Nous démontrons ainsi par l'absurde qu'il est impossible d'avoir un chemin ne correspondant à aucun des  $p$  monômes considérés lorsqu'il y a compatibilité deux à deux au sens large.



Nous verrons l'intérêt de ces constellations par la suite.

Il est à remarquer que l'ensemble des constellations maximales ne forme pas une partition de l'ensemble des monômes considérés : en général l'intersection de deux constellations maximales n'est pas vide.

Exemple

$$F = a \oplus b \oplus c \oplus d = \sum_{i=1}^{i=8} \mu_i$$

$\mu_1 = a'bcd$	$\mu_5 = ab'c'd'$
$\mu_2 = ab'cd$	$\mu_6 = a'bc'd'$
$\mu_3 = abc'd$	$\mu_7 = a'b'cd'$
$\mu_4 = abcd'$	$\mu_8 = a'b'c'd$

Cette fonction doit être synthétisée sur le réseau (1,a,b,c,d).

Nous obtenons les constellations maximales

$$C_1 = \{\mu_i / i = 1, 2, 4, 5, 6\}$$

$$C_2 = \{\mu_j / j = 3, 4, 6, 7, 8\}$$

$$C_1 \wedge C_2 = \{\mu_4, \mu_6\} \neq \{\emptyset\}$$

Corollaire 1

Soit  $E_1 = \{\mu_i / i = 1, \dots, p\}$  un ensemble de monômes inférieurs au produit  $\tilde{x}_{k-1} \cdot \tilde{x}_k$  ( $p \neq 0$ ), et  $E_2 = \{\pi_j / j = 1, \dots, q\}$  un ensemble de monômes inférieurs au produit  $\tilde{x}'_{k-1} \cdot \tilde{x}_k$  ( $q \neq 0$ ).

Ces deux ensembles ont le sommet  $\tilde{x}_k$  en commun sur le réseau.

Si on a  $p \neq q$  alors la réunion  $E_1 \vee E_2$  ne peut pas former une constellation.

Démonstration

Supposons donc  $p \neq q$  sans perdre en généralité nous pouvons supposer  $p > q$ .

Il existe donc un monôme

$$\mu_\ell = T_a \cdot \tilde{x}_{k-1} \cdot \tilde{x}_k \cdot Q_\ell \in E_1$$

et l'on peut affirmer que l'on a

$$\pi_\ell = T_b \cdot \tilde{x}'_{k-1} \cdot \tilde{x}_k \cdot Q_\ell \in E_2$$

Comme  $q$  est différent de zéro nous avons nécessairement

$$\mu_j = T_a \cdot \tilde{x}_{k-1} \cdot \tilde{x}_k \cdot Q_j \in E_1$$

$$\pi_j = T_b \cdot \tilde{x}'_{k-1} \cdot \tilde{x}_k \cdot Q_j \in E_2$$

avec  $j \neq \ell$

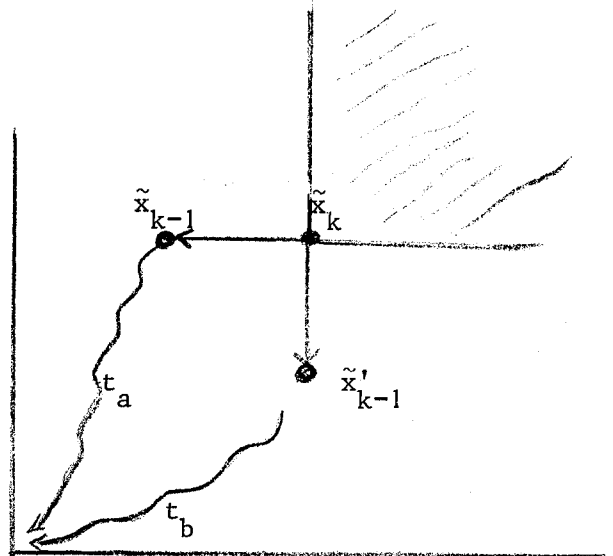
Pour que  $E_1 \vee E_2$  forme une constellation il faudrait que  $\mu_\ell$  et  $\pi_j$  en particulier soient compatibles relativement à  $E_1 \vee E_2$ , ce qui nécessiterait

$$\pi_\ell \in E_2$$

or ceci est faux par hypothèse.

Corollaire 2

Soit  $m_t$  un monôme dont le chemin associé coupe des chemins représentant la constellation  $E_1 \vee E_2$  précédente dans des couches d'ordre supérieur à  $k$ . Pour que  $\{m_t\} \vee E_1 \vee E_2$  forme une constellation, il suffit de vérifier que  $\{m_t\} \vee E_1$  forme une constellation.



Démonstration

Le chemin  $C_t$  associé à  $m_t = T_t \cdot \tilde{x}_t \cdot Q_t$  coupe des chemins de la constellation  $E_1 \vee E_2$  en un point  $\tilde{x}_t$  appartenant à une couche d'ordre  $t$  supérieur à  $k$ .

$C_t$  coupera au moins deux chemins, l'un correspondant à un monôme  $\mu_t$  de  $E_1$  et l'autre à un monôme  $\pi_t$  de  $E_2$

$$\mu_t = T_c \cdot \tilde{x}_t \cdot Q_n$$

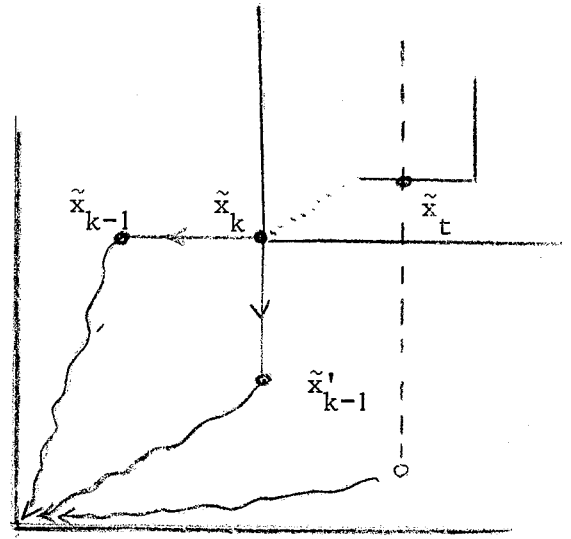
$$\pi_t = T_d \cdot \tilde{x}_t \cdot Q_n$$

Pour que  $\{m_t\} \vee E_1 \vee E_2$  forme une constellation, d'après le théorème précédent, il est en particulier nécessaire que  $m_t$  soit compatible avec chacun des monômes  $\mu_t$  et  $\pi_t$  relativement à  $\{m_t\} \vee E_1 \vee E_2 = E$ .

Montrons que la compatibilité avec l'un d'entre eux,  $\mu_t$  par exemple, entraîne automatiquement la compatibilité avec l'autre  $\pi_t$

$$m_t C_E \mu_t \Rightarrow \begin{cases} T_t \cdot \tilde{x}_t \cdot Q_n \in E \\ T_c \cdot \tilde{x}_t \cdot Q_t \in E_1 \end{cases}$$

$$E_1 \vee E_2 \text{ est une constellation} \Rightarrow T_d \cdot \tilde{x}_t \cdot Q_t \in E_2$$



Or le fait que l'on ait

$$T_t \cdot \tilde{x}_t \cdot Q_n \in E$$

$$T_d \cdot \tilde{x}_t \cdot Q_t \in E_2 < E$$

entraîne que  $m_t$  et  $\pi_t$  sont compatibles par rapport à  $E$ .



CHAPITRE - IV

LES OPERATEURS UTILISES

IV-1 - FORME GENERALE DES OPERATEURS UTILISES

Tous les opérateurs utilisés ont le même aspect extérieur :

3 entrées u, v, z

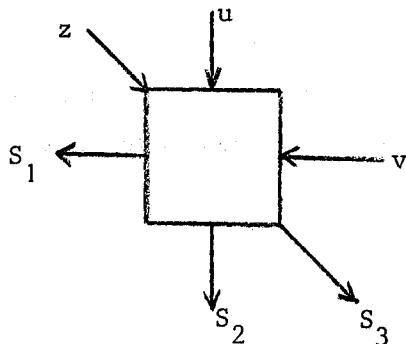
3 sortie  $S_1$ ,  $S_2$ ,  $S_3$ .

A chaque sommet du réseau (symbolisé jusqu'ici par une des variables  $\tilde{x}_i$ ) est placé un opérateur.

Les trois directions représentées sur le croquis ci-dessus sont utilisées pour relier les opérateurs entre eux.

Toutes les entrées ou sorties de l'opérateur ne sont pas toujours utilisées pour la synthèse d'une fonction donnée.

Si une entrée n'est pas utilisée nous la supposons égale à "1" ou "0" suivant l'opérateur utilisé.



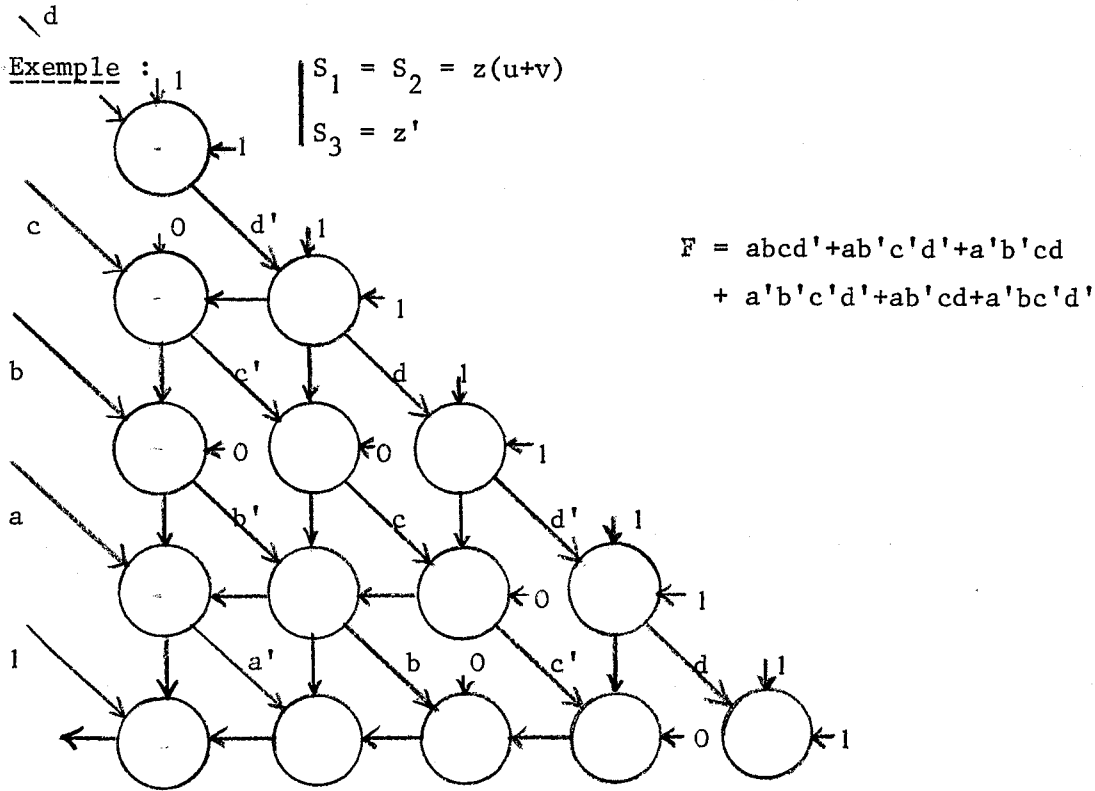
IV-2 - FONCTIONS DE SORTIE DES OPERATEURS

Les fonctions de sortie  $S_1, S_2, S_3$  doivent permettre au réseau de disposer des  $2^n$  chemins représentant les  $2^n$  monômes canoniques car il est nécessaire que l'on ait toujours bijection entre les deux ensembles.

Chaque opérateur de couche  $i$ , donc lié à une variable  $\tilde{x}_i$ , doit offrir deux liaisons représentant les cheminement  $\tilde{x}_i \tilde{x}_{i+1}$

et

$$\tilde{x}_i \tilde{x}'_{i+1}.$$



Les sorties  $S_1, S_2$  servent à élaborer la fonction à synthétiser en transformant l'information qui est introduite par les trois entrées. La sortie  $S_3$  permet d'irriguer l'intérieur du réseau avec les variables d'entrée.

Nous aurons ainsi généralement

$$\left| \begin{array}{l} S_1 = S_1(z, u, v) \\ S_2 = S_2(z, u, v) \\ S_3 = S_3(z) = \begin{cases} z \\ z \end{cases} \end{array} \right.$$

#### IV-3 - NOMBRE D'OPERATEURS DIFFERENTS

Chacune des deux fonctions  $S_1$ ,  $S_2$  dépendant de trois variables, nous aurons

$$\frac{256 \times 257}{2} \text{ couples } (S_1, S_2) \text{ non ordonnés.}$$

En tenant compte des deux choix possibles pour  $S_3$  nous aurons 65.792 opérateurs possibles.

En fait, le nombre d'opérateurs que nous conserverons ici sera beaucoup plus petit.

Pour réduire les calculs booléens formels nous exigerons que les fonctions  $S_1$ ,  $S_2$  soient croissantes par rapport à  $u$  et  $v$ .

Il y a 6 fonctions croissantes de 2 variables :

$$uv, u+v, u, v, 0, 1.$$

Elles correspondent à 34 fonctions  $S_1$  (ou  $S_2$ ), soit 1190 opérateurs possibles.



fonctions représentées par $S_1/S_2$			
0	1	z	z'
u	v	uz	uz'
zu+v	zv+u	vz	vz'
z'u+v	z'v+u	(u+v)z	(u+v)z'
(z+v)u	(z+u)v	(z+u)(z+v)	(z'+u)(z'+v)
(z'+v)u	(z'+u)v	(z+u)(z'+v)	(z+v)(z'+u)
z'+u	z'+v	z+u	z+v
uv	u+v	uvz	uvz'
zu+z'v	zv+z'u		

En fait, toutes ces disponibilités ne sont pas disponibles simultanément. Elles se partagent en deux groupes relatifs aux deux formes PS et SP pour l'écriture d'une fonction.

Pour une fonction écrite sous la forme SP seules 23 des 34 fonctions sont utilisables.

Pour une fonction écrite sous la première forme de Lagrange (ou sous une forme lexicographique) seules 4 fonctions conviennent :

$$\left\{ \begin{array}{l} z(u+v) \\ z'(u+v) \\ zu+z'v \\ z'u+zv. \end{array} \right.$$

Soit 20 opérateurs différents seulement.

Ce choix peut encore être restreint :

- \* certains opérateurs présentent des symétries
- \* certains réseaux composés d'opérateurs différents offrent en réalité un agencement identique des chemins.

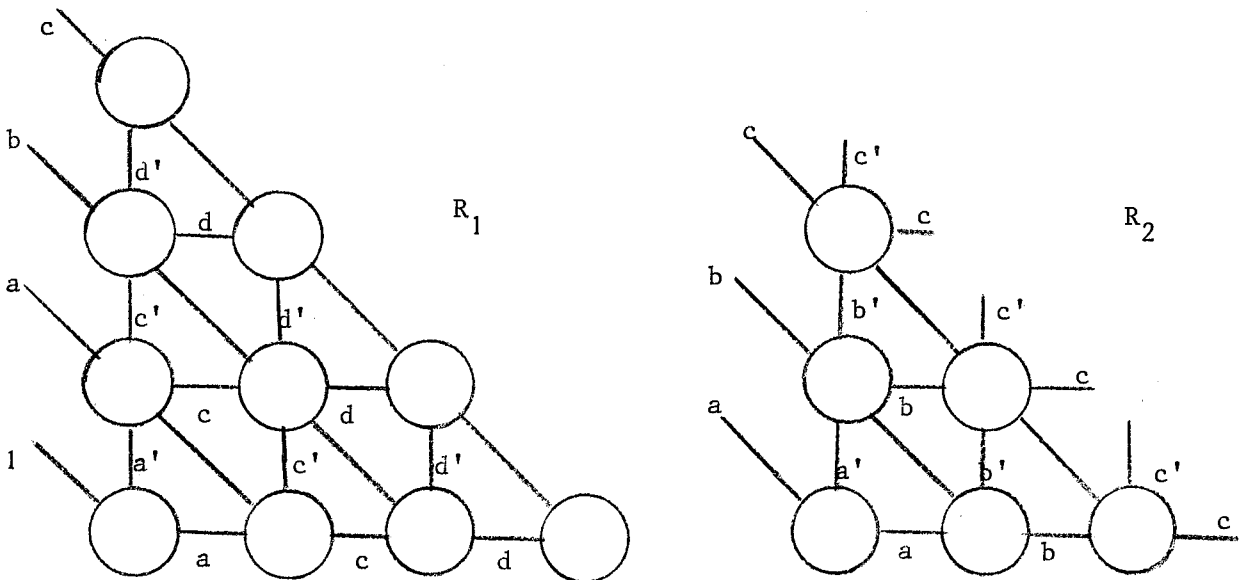
Exemple : Le réseau  $R_1$  est formé des opérateurs

$$\begin{cases} S_1 = z(u+v) \\ S_2 = z'(u+v) \\ S_3 = z \end{cases}$$

Quant à  $R_2$  :

$$\begin{cases} S_1 = S_2 = zu+z'v = U_{zuv} \\ S_3 = z. \end{cases}$$

Ces deux réseaux sont identiques car toute fonction synthétisable dans l'un l'est aussi dans l'autre. On le voit aisément sur la configuration identique des chemins :



Pour des raisons de simplicité d'algorithme et d'efficacité technique (masques...) nous nous sommes surtout intéressés aux méthodes n'utilisant qu'un seul type d'opérateur par réseau.

Les programmes ont été écrits pour l'opérateur

$$\theta_1 \begin{cases} S_1 = S_2 = z(u+v) \\ S_3 = z' \end{cases}$$

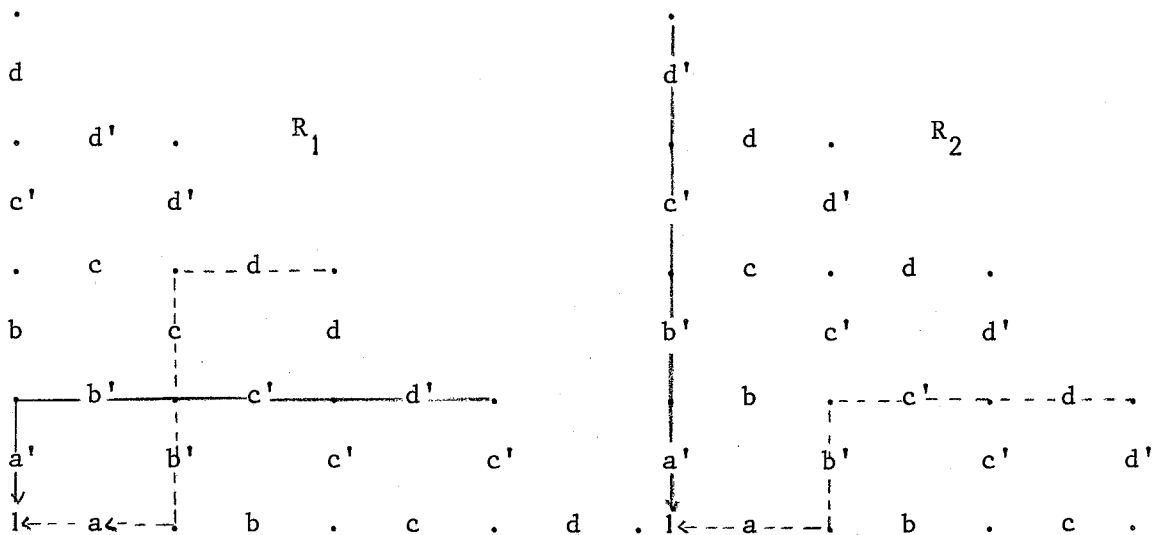
qui a jusqu'ici été utilisé implicitement dans tous les exemples.

Certains programmes ont été adaptés pour l'opérateur

$$\theta_2 \begin{cases} S_1 = z(u+v) \\ S_2 = z'(u+v) \\ S_3 = z. \end{cases}$$

L'adaptation se fait aisément pour une nouvelle cellule, il suffit de modifier le système d'intersection des chemins.

Exemple :



Opérateur  $\theta_1$

$\{1, a', b', c', d'\}$

et

$\{1, a, b', c, d\}$  sécants en  $\{b'\}$

Les monômes  $a'b'c'd'$  et  $ab'cd$  sont ici incompatibles sur  $R_1$ .

Opérateur  $\theta_2$

Les deux chemins ne sont plus sécants et les monômes sont compatibles sur  $R_2$ .



CHAPITRE V

SYNTHESE D'UNE FONCTION POUR UN ORDRE DONNE DES VARIABLES

CAS GENERAL SUR RESEAUX IRREDONDANTS

V-1 - DEFINITIONS

Nous avons vu plus haut que l'on pouvait représenter tout réseau par sa première ligne.

DEFINITION 4

*Nous appellerons réseau irrédondant un réseau représenté par*

$$(1, \tilde{x}_{i1}, \tilde{x}_{i2}, \dots, \tilde{x}_{in})$$

avec  $x_{ij} \neq \tilde{x}_{ik}$

$$\forall ij \neq ik \quad \text{pour} \quad ij, ik = 1, \dots, n$$

DEFINITION 5

*Nous appellerons réseau redondant un réseau représenté par*

$$(1, x_{i1}, x_{i2}, \dots, x_{in}, \dots, x_{ir})$$

avec  $r \geq n$

et où il existe au moins un ensemble  $J = \{j_1, j_2, \dots, j_s\}$  pour lequel on ait

$$\# \quad s \geq 2$$

$$\# \quad j^k \neq j^l \text{ et } s_{jk} = \tilde{x}_{jl} \quad \forall j^k, j^l \in J$$

et

$$J \subset \{i_1, i_2, \dots, i_r\}.$$

DEFINITION 6

Dans un réseau redondant nous dirons que la  $ik$ -ième couche est redondante si l'on a :

$$x_{ik} = \tilde{x}_{ij} \quad \text{avec } ij < ik.$$

DEFINITION 7

Dans un réseau redondant nous dirons que la  $ik$ -ième couche est irrédondante si l'on a

$$x_{ik} \neq \tilde{x}_{ij} \quad \forall ij < ik.$$

DEFINITION 8

Les deux couches  $ik$  et  $ij$  seront dites adjacentes si l'on a

$$ik = ij \pm 1.$$

V-2 - LA SYNTHÈSE

La fonction  $F(x_1, \dots, x_n)$  à synthétiser est donnée par une forme de Lagrange ou par une forme lexicographique correspondant à l'ordre imposé pour les variables.

$$x_{i_1}^{j_1} x_{i_2}^{j_2} \dots x_{i_n}^{j_n} .$$

L'opérateur utilisé étant précisé, on connaît alors sur le réseau la bijection qui aux monômes de  $F$  fait correspondre les chemins.

A  $F$  est associé le graphe des incompatibilités  $G_c(F)$  où

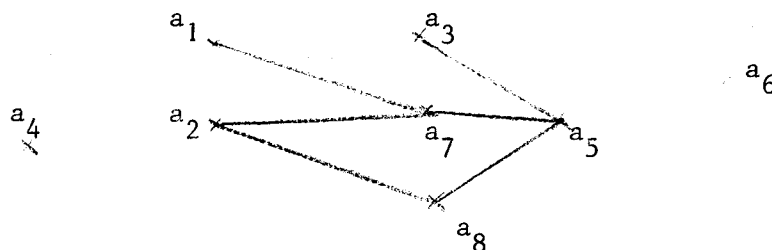
\* chaque sommet  $a_i$  représente un monôme  $\mu_i$  de  $F$

\*  $a_i$  et  $a_j$  sont adjacents (reliés par une arête) si  $\mu_i$  et  $\mu_j$  sont incompatibles.

Exemple : Reprenons l'exemple du paragraphe III-7 :

$$F(a,b,c,d) = a \otimes b \otimes c \otimes d$$

Le graphe  $G_c(F)$  sera le suivant pour le réseau  $(1, a, b, c, d)$



Dans le cas général la synthèse d'une fonction nécessite la connaissance de :

- 1 - la fonction
- 2 - l'opérateur utilisé dans le réseau
- 3 - l'ordre des variables.





Le but de la synthèse est alors de déterminer le nombre minimal de constellations maximales formées par les monômes de la fonction.

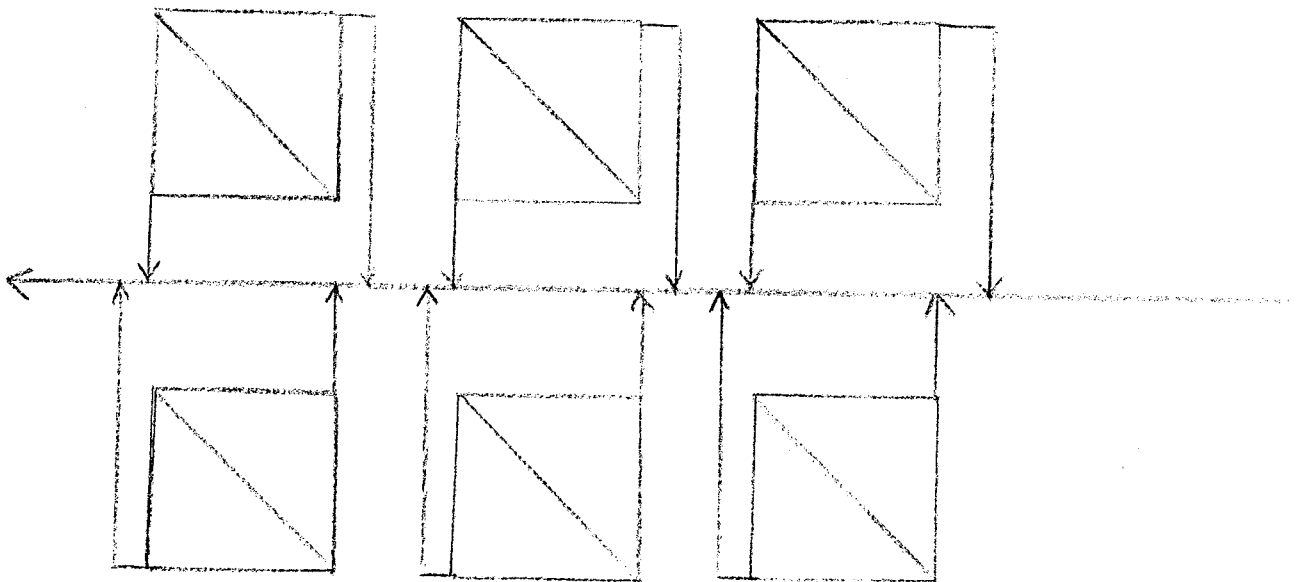
Cette recherche s'appuie sur le théorème des constellations démontré précédemment : on forme les compatibilités deux à deux ce qui est très bien représenté par les arêtes du graphe d'incompatibilité  $G_c(F)$ .

Au point de vue graphe nous sommes ainsi conduits à :

- 1 - déterminer les ensembles intérieurement stables maximaux.
- 2 - rechercher une couverture minimale.

C'est le calcul du nombre chromatique de  $G_c(F)$ .

Chacune des constellations ainsi formées étant synthétisable sur un réseau , nous obtenons un réseau formé de modules  avec une ligne collectrice donnant  $F(x_1, \dots, x_n)$ .



Exemple : Pour l'exemple précédent nous avons quatre ensembles intérieurement stables maximaux

$$I_1 = \{a_1, a_2, a_4, a_5, a_6\}$$

$$I_2 = \{a_3, a_4, a_6, a_7, a_8\}$$

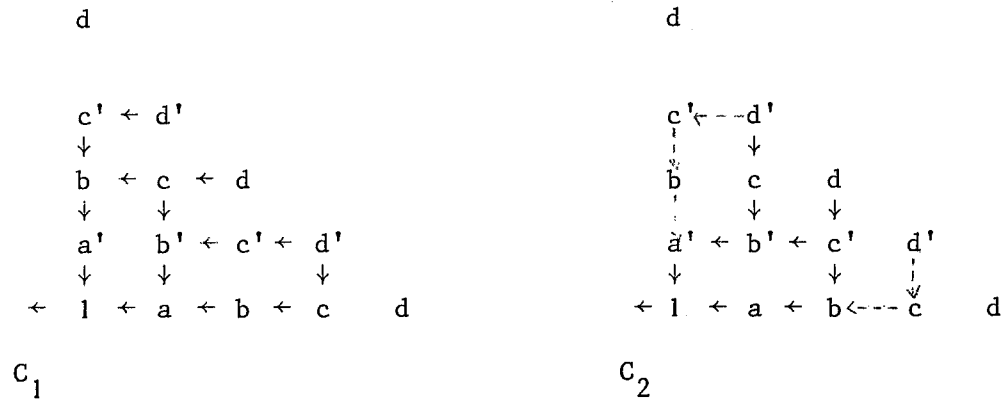
$$I_3 = \{a_1, a_2, a_3, a_4, a_6\}$$

$$I_4 = \{a_1, a_3, a_4, a_6, a_8\}$$

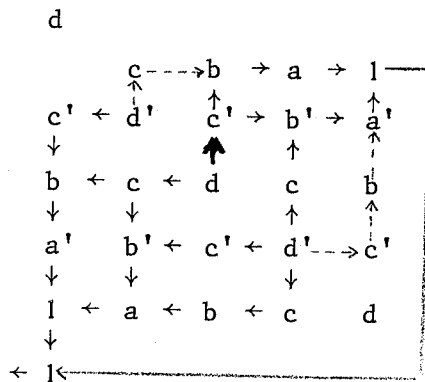
Une couverture minimale sera formée par

$$\{I_1, I_2\}$$

Deux réseaux  $\triangle$  suffiront donc à synthétiser la fonction



Les flèches en pointillé représentent  $I_1 \wedge I_2$ .  
On peut donner la représentation suivante :



UNIVERSITÉ DE GRENOBLE  
 SERVICE DE MATHÉMATIQUES  
 APPLIQUÉES  
 POLYCOPIES  
 CEDEX N° 53  
 38 - GRENOBLE - GARE

La méthode comprend trois étapes :

$$F(x_1, \dots, x_n) = \sum_{i=1}^{i=p} \mu_i = \prod_{j=1}^{j=q} \nu_j$$

- 1 - formation du graphe des incompatibilités
- 2 - recherche des ensembles intérieurement stables maximaux
- 3 - couverture minimale.

Ces algorithmes ont été programmés en PLI et utilisent des méthodes booléennes (travaux de M. Benzaken). Le listing figure en appendice. Nous donnons dans les pages suivantes un exemple :

$$F = a \oplus b \oplus c \oplus d \oplus e = \sum_{i=1}^{16} \mu_i$$

traité par le programme.

$$\mu_1 = ab'c'd'e'$$

$$\mu_9 = a'bcd'e'$$

$$\mu_2 = a'bc'd'e'$$

$$\mu_{10} = ab'c'de$$

$$\mu_3 = a'b'cd'e'$$

$$\mu_{11} = ab'cd'e$$

$$\mu_4 = a'b'c'de'$$

$$\mu_{12} = ab'cde'$$

$$\mu_5 = a'b'c'd'e$$

$$\mu_{13} = abc'd'e$$

$$\mu_6 = a'b'cde$$

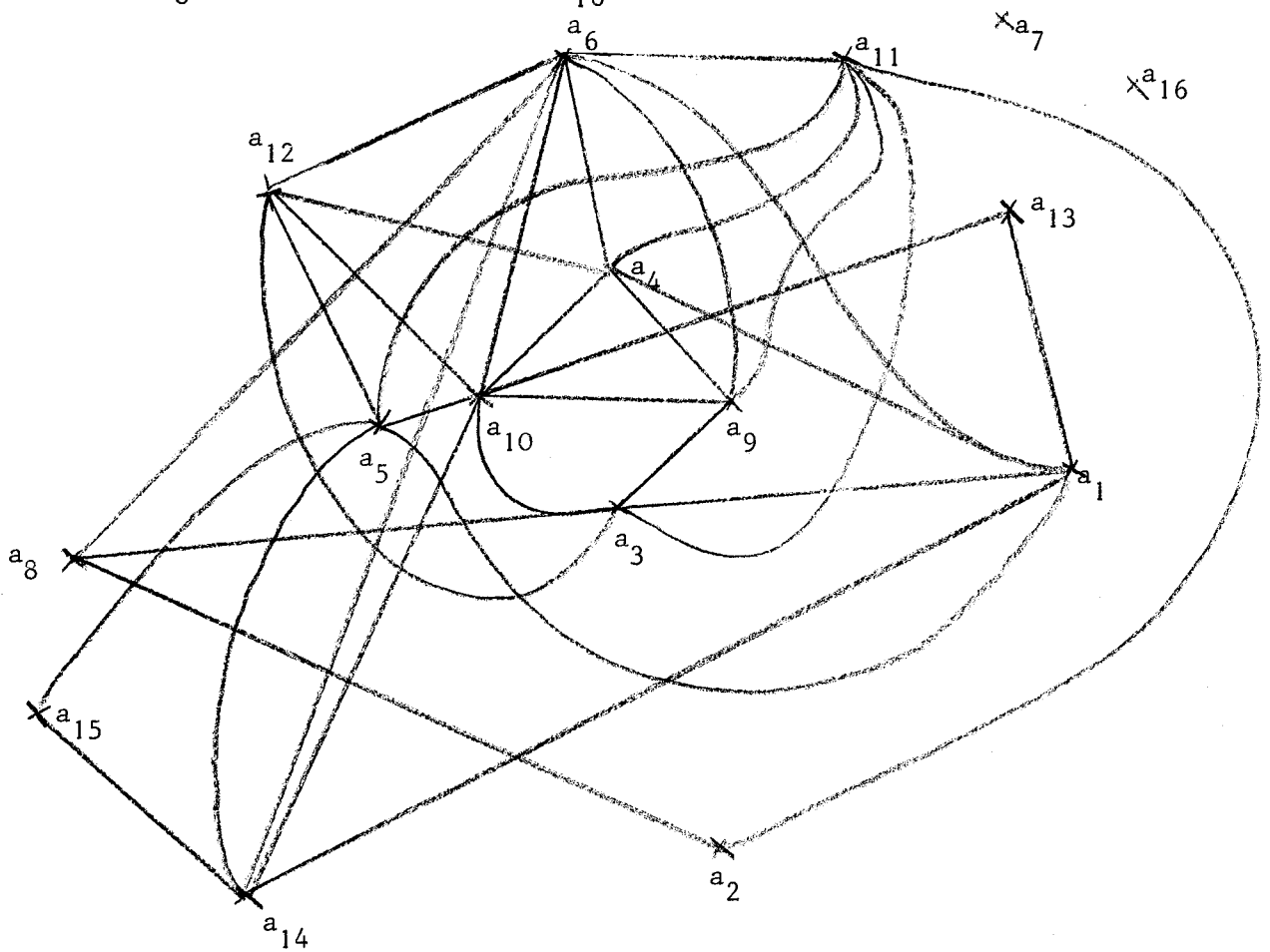
$$\mu_{14} = abc'de'$$

$$\mu_7 = a'b'c'de$$

$$\mu_{15} = abcd'e'$$

$$\mu_8 = a'bcd'e$$

$$\mu_{16} = abcde$$



Nous obtenons une couverture minimale de constellations.

Conclusion

Cette méthode permet la synthèse de toute fonction, mais elle donne un résultat assez lourd.

Nous verrons plus loin des méthodes de synthèse plus satisfaisantes.

CHAPITRE - VI

SYNTHESE DE FONCTIONS COMPLETES SUR UN RESEAU  IRREDONDANT

IV-2 - NOMBRE DE RESEaux CORRESPONDANTS A n VARIABLES

Nous avons vu que l'ordre des variables jouait un rôle très important pour la synthèse d'une fonction. En tenant compte des permutations et complémentations des n variables, il y a

$n!2^n$  écritures différentes de

$$\tilde{x}_{i_1} \tilde{x}_{i_2} \dots \tilde{x}_{i_n}$$

Exemple :  $n = 2$

$$2!2^2 = 8 \text{ soit}$$

ab	ba
a'b	ba'
ab'	b'a
a'b'	b'a'

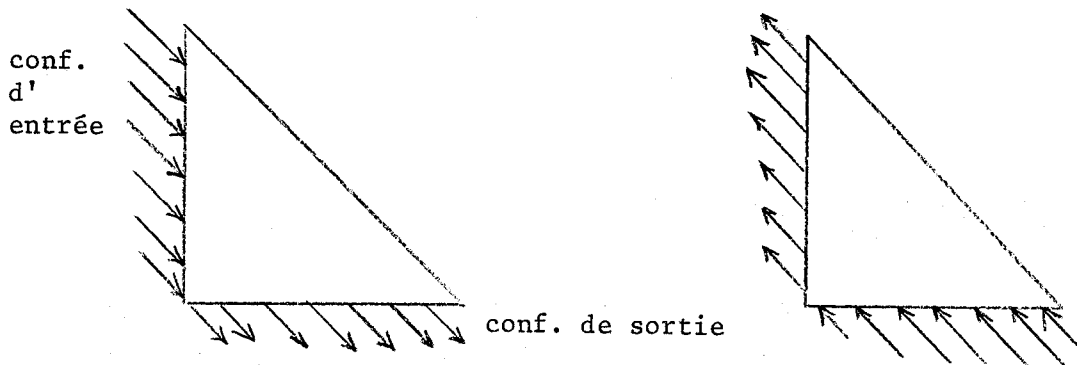
PROPRIETE 1

Les deux réseaux  $(1, \tilde{x}_{i_1}, \dots, \tilde{x}_{i_n})$  et  $(1, \tilde{x}_{i_1}, \dots, \tilde{x}'_{i_n})$  permettent de synthétiser les mêmes fonctions : l'accentuation de la dernière variable du réseau ne joue aucun rôle puisque la dernière couche du réseau ne peut provoquer aucune incompatibilité : pour deux chemins s'y coupant, leurs queues sont vides et par suite aucun problème de compatibilité ne se pose.

PROPRIETE 2

*Il existe toujours deux configurations différentes des variables d'entrée qui correspondent au même réseau, à une symétrie près par rapport à la première bissectrice.*

Ceci est évident lorsque l'on considère le réseau : en permutant les deux configurations d'entrée et de sortie, rien n'est changé.



\* pour l'opérateur 
$$\begin{cases} S_1, S_2 = z(u+v) \\ S_3 = z' \end{cases}$$

nous aurons les deux réseaux identiques :

$$(1, \tilde{x}_{i_1}, \tilde{x}_{i_2}, \tilde{x}_{i_3}, \dots, \tilde{x}_{i_{2j-1}}, \tilde{x}_{i_{2j}}, \dots, \tilde{x}_{i_n})$$

$$(1, \tilde{x}'_{i_1}, \tilde{x}'_{i_2}, \tilde{x}'_{i_3}, \dots, \tilde{x}'_{i_{2j-1}}, \tilde{x}'_{i_{2j}}, \dots, \tilde{x}'_{i_n})$$

\* pour l'opérateur 
$$\begin{cases} S_1 = z(u+v) \\ S_2 = z'(u+v) \\ S_3 = z \end{cases}$$

les deux réseaux

et 
$$(1, \tilde{x}_{i_1}, \tilde{x}_{i_2}, \dots, \tilde{x}_{i_n})$$

et 
$$(1, \tilde{x}'_{i_1}, \tilde{x}'_{i_2}, \dots, \tilde{x}'_{i_n})$$
 sont identiques.

Ces deux propriétés permettent de diviser par 4 le nombre de réseaux différents. Il nous en reste donc

$$p(n) = n!2^{n-2}$$

Exemple :  $n = 2$       $p(2) = 2$       $(1,a,b)$  et  $(1,b,a)$

$n = 3$       $p(3) = 6$       $(1,a,b,c)$     $(1,a,c,b)$   
     $(1,b,c,a)$     $(1,b,a,c)$   
     $(1,c,a,b)$     $(1,c,b,a)$

### VI-3 - SYNTHESE D'UNE FONCTION DONNEE

La synthèse d'une fonction est conditionnée par l'ordre des variables, le problème qu'elle pose va donc se concrétiser en une recherche d'un ordre lexicographique convenable.

Nous venons de dénombrer  $p(n)$  choix possibles. Il semble ainsi qu'il faille étudier chacun de ces cas. En fait, seul un très petit nombre va être effectivement consulté par l'algorithme.

Comme on le verra au paragraphe suivant, la synthèse se fera par étapes successives, couche par couche. Si à la  $k^{\text{ième}}$  couche une incompatibilité se déclare, il y a  $(n-k)! 2^{n-k-1}$  possibilités qui ne seront pas envisagées.

C'est en fait ce phénomène qui rend l'algorithme utilisable, seul un faible pourcentage des  $p(n)$  cas est exploré.

Nous donnons dans les pages suivantes quelques valeurs du rapport

$$\frac{\text{nombre de cas étudiés par l'algorithme}}{p(n)}$$

pour différents exemples.

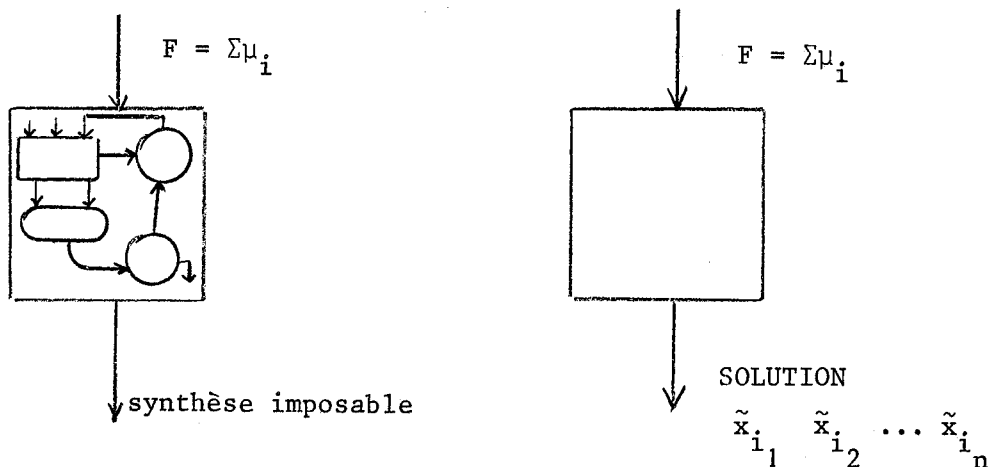


Toute fonction n'est pas synthétisable par cette méthode, contrairement à celle du chapitre précédent. Nous verrons plus loin ce qu'il y a lieu de faire en cas de synthèse impossible.

#### VI-4 - L'ALGORITHME

Le but de l'algorithme est de :

- \* donner l'ordre des variables permettant de représenter la fonction sur le réseau  $\Delta$  lorsque cette synthèse est possible.
- \* diagnostiquer l'impossibilité de réaliser la synthèse.



L'algorithme procède couche par couche, en partant de la première, et en se référant à un certain ordre des  $p(n)$  choix que l'on peut représenter par une arborescence :

#### VI-4-1 - Arbre des choix

Les variables sont rangées dans l'ordre suivant, formant une chaîne ascendante :

$$x_0 < x_1 < x'_1 < x_2 < x'_2 < \dots < x_n < x'_n.$$

A l'instant  $t$  la variable affectée à la couche  $i$  est  $\tilde{x}_t$ . S'il faut revenir sur ce choix à l'instant  $(t+\Delta t)$ , nous attacherons à la couche  $i$  la variable

$$\begin{aligned} \tilde{x}_{t+\Delta t} &= x'_t & \text{si } \tilde{x}_t &= x_t \\ &= x_{t+1} & \text{si } \tilde{x}_t &= x'_t. \end{aligned}$$

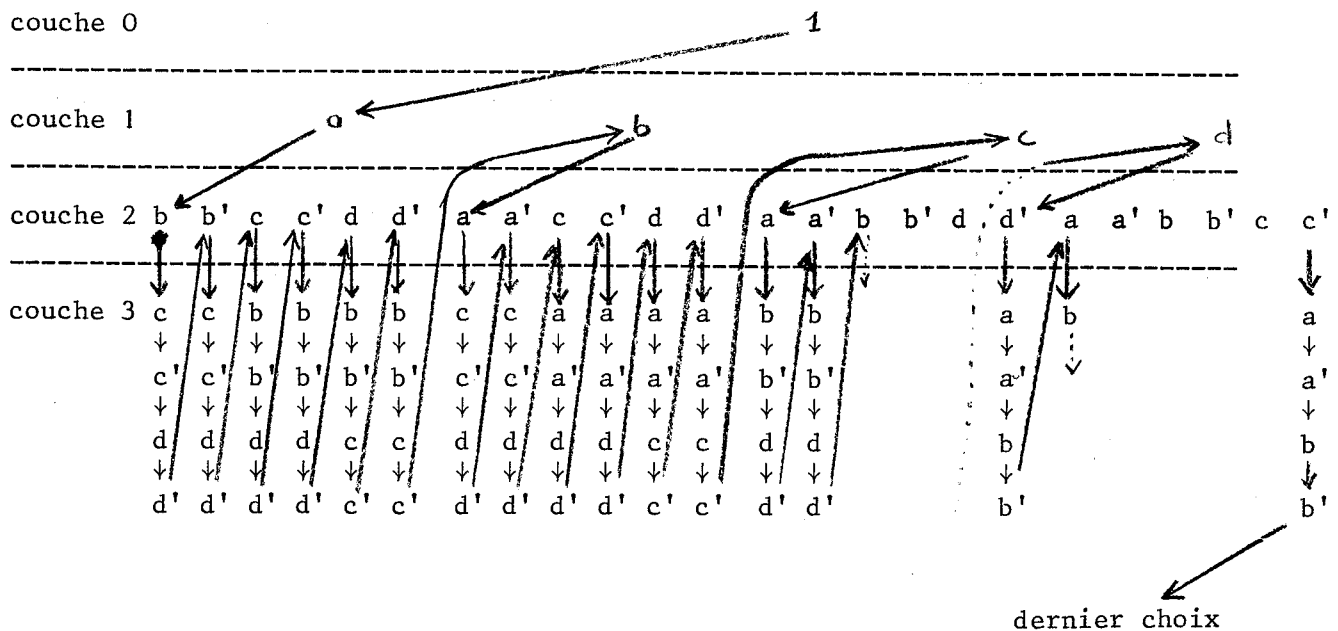
Dans le cas  $\tilde{x}_t = x'_t$  nous aurons

$$\tilde{x}_{t+\Delta t} = x_{t+1}$$

si et seulement si on n'a pas déjà affecté  $\tilde{x}_{t+1}$  à une couche d'ordre inférieure à  $i$ . Ceci parce que nous considérons un réseau irrédondant.

Si l'on a  $\tilde{x}_t = x'_n$  alors il faut revenir sur le choix de la variable affectée à la couche  $(i-1)$ .

Exemple : arbre des choix pour  $n = 4$  :



Remarque - A la couche 1 seules les variables directes sont affectées pour tenir compte de la propriété 2.

VI-4-2 - Sur la couche i

On forme pour chaque sommet  $\tilde{x}_k$  ( $k = 1, \dots, i+1$ )

\* l'ensemble  $\mu_g = \{\mu_{g_i} / i = 1, \dots, \alpha\}$

des monômes contenant les lettres  $\tilde{x}_{k-1}$  et  $\tilde{x}_k$

\* l'ensemble  $\mu_b = \{\mu_{b_j} / j = 1, \dots, \beta\}$

des monômes contenant les lettres  $\tilde{x}'_{k-1}$  et  $\tilde{x}_k$

et passant par le sommet  $\tilde{x}_k$  en question.

- 1 - Si  $\mu_g$  et  $\mu_b$  sont vides on passe au sommet suivant puisqu'il ne peut y avoir d'incompatibilités.
- 2 - Si un seul des deux ensembles est vide, il ne peut y avoir d'incompatibilité, on passe à la suite.
- 3 - Si aucun des deux ensembles n'est vide :
  - 3.1 Un test rapide sur les valeurs de  $\alpha$  et  $\beta$  nous dira, il y a ou non incompatibilité : si l'on a  $\alpha \neq \beta$  alors l'union  $\mu_b \vee \mu_g$  ne peut former une constellation d'après le corollaire 1 du chapitre III.  
On passe au choix suivant.
  - 3.2 Si  $\alpha = \beta$  il y a lieu de voir si  $\mu_b \vee \mu_g$  forme une constellation.

3.2.1 - S'il n'y a pas constellation on passe au choix suivant.

3.2.2 - S'il y a constellation on conserve un seul des deux ensembles  $\mu_b/\mu_g$  ce qui est permis d'après le Corollaire 2 du chapitre III. Ceci allège beaucoup la synthèse, et on passe au sommet suivant.

Lorsque pour la couche  $i$  a été trouvé un choix convenable, on passe à la couche suivante.

La synthèse se termine de façon positive lorsqu'on a une affectation convenable pour la couche  $(n-1)$ . D'après la propriété 2 il n'y a pas lieu d'étudier le comportement de la dernière couche.

Si cette affectation convenable n'est pas trouvée, nous en déduisons que la synthèse est impossible.

#### VI-5 - REMARQUES SUR LE NOMBRE DE FONCTIONS SYNTHÉTISABLES

Chaque fois qu'une méthode de synthèse non universelle est proposée, il est intéressant d'étudier la proportion de fonctions synthétisables. Ce peut être un critère de qualité de la méthode.

##### VI-5-1 - Fonctions d'une seule variable ( $n = 1$ )

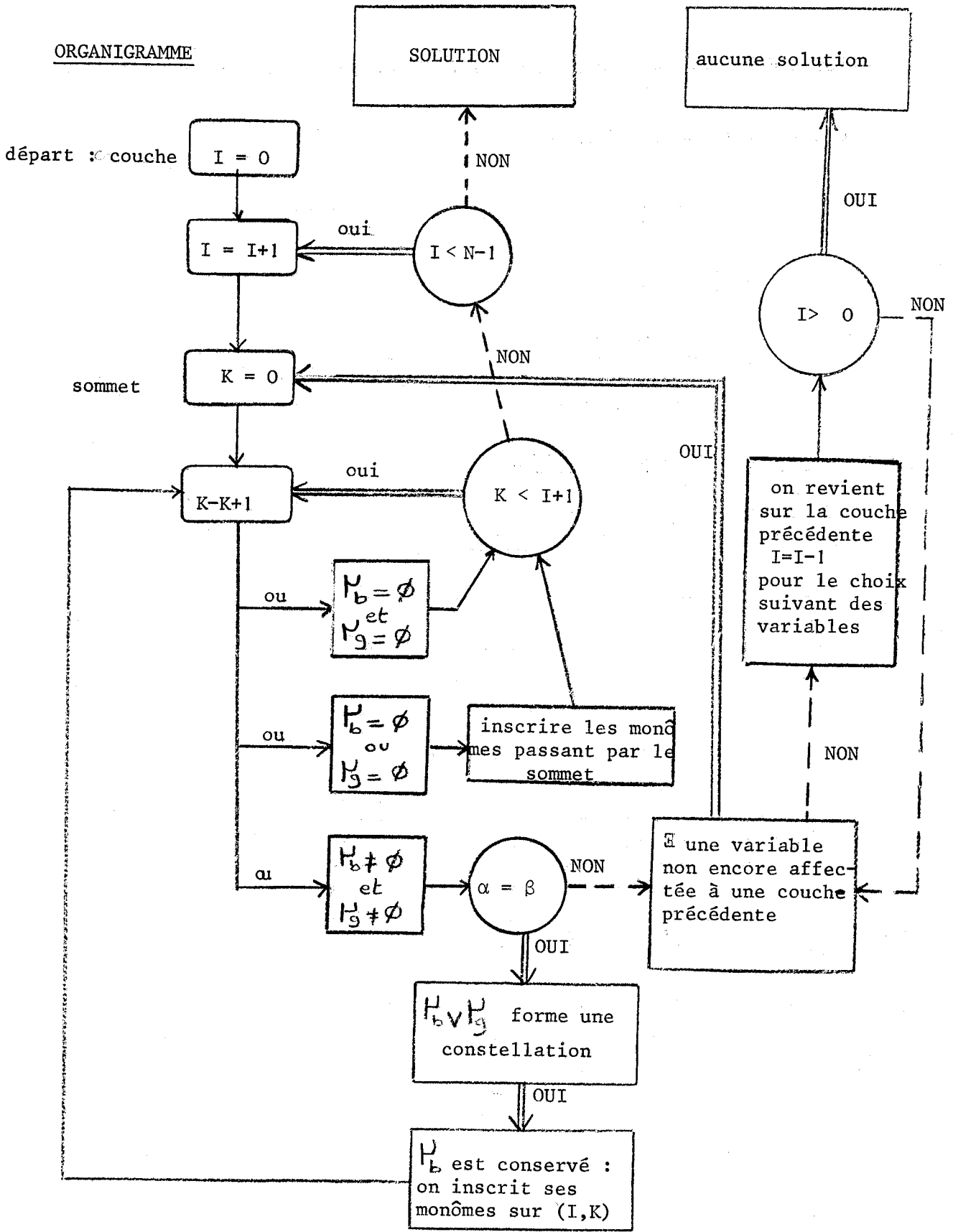
Elles sont toutes synthétisables sur le réseau. Nous n'étudierons bien sûr les fonctions qu'au moyen d'un représentant par classe d'équivalence.

Ces classes d'équivalence sont déterminées par le groupe produit du :

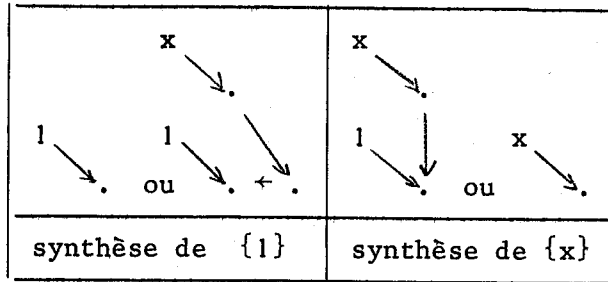
- groupe des permutations,
- du groupe des complémentations,

agissant sur l'ensemble des variables de la fonction et de la complémentations de la fonction elle-même.

ORGANIGRAMME

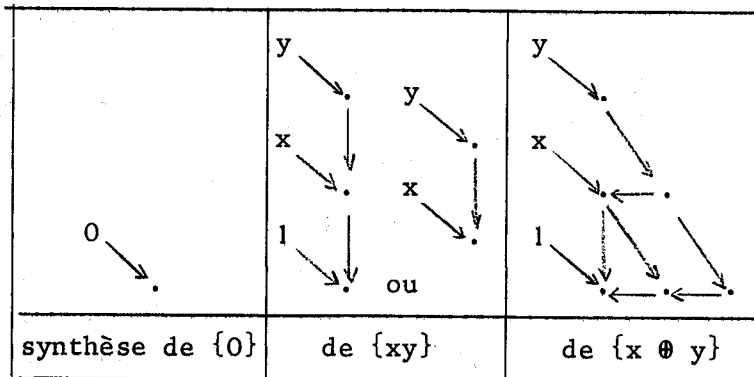


Pour  $n = 1$  nous avons deux classes d'équivalence représentées par  $\{1\}$   
 $\{x\}$   
 dont la synthèse est donnée ci-dessous :



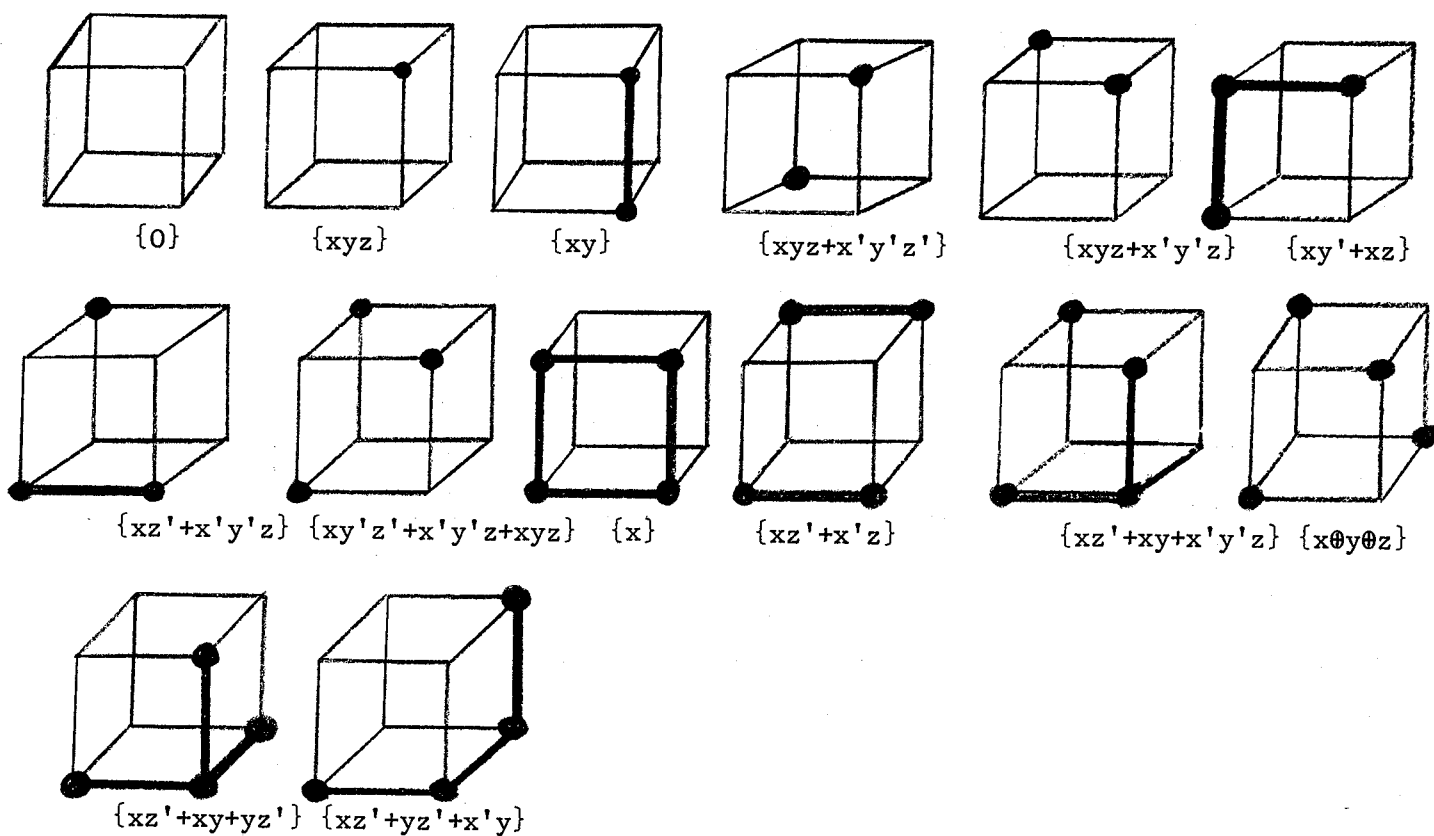
VII-5-2 - Fonctions de deux variables ( $n = 2$ )

Nous aurons 4 classes d'équivalentes :  $\{0\}$   $\{xy\}$   $\{x\}$   $\{x \oplus y\}$

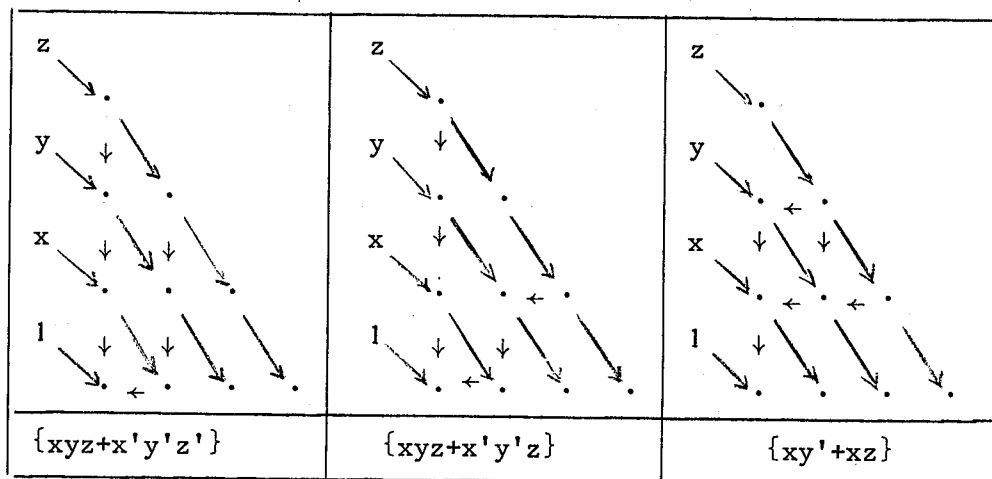


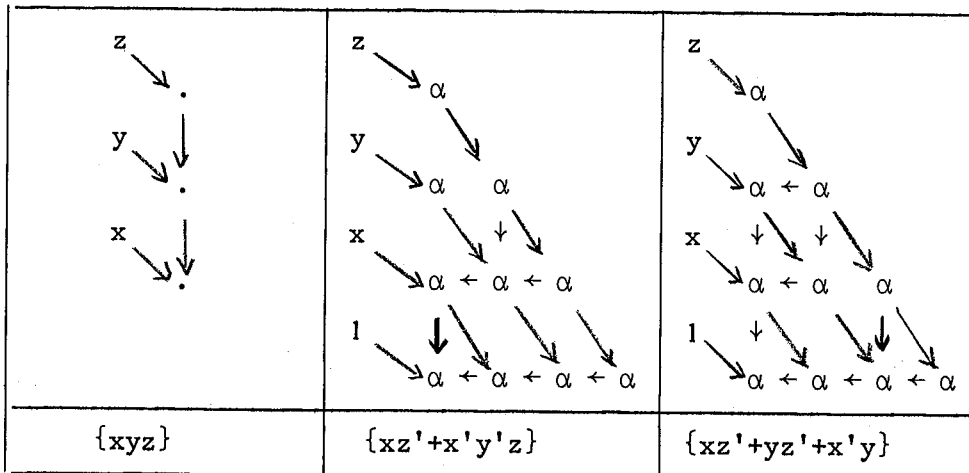
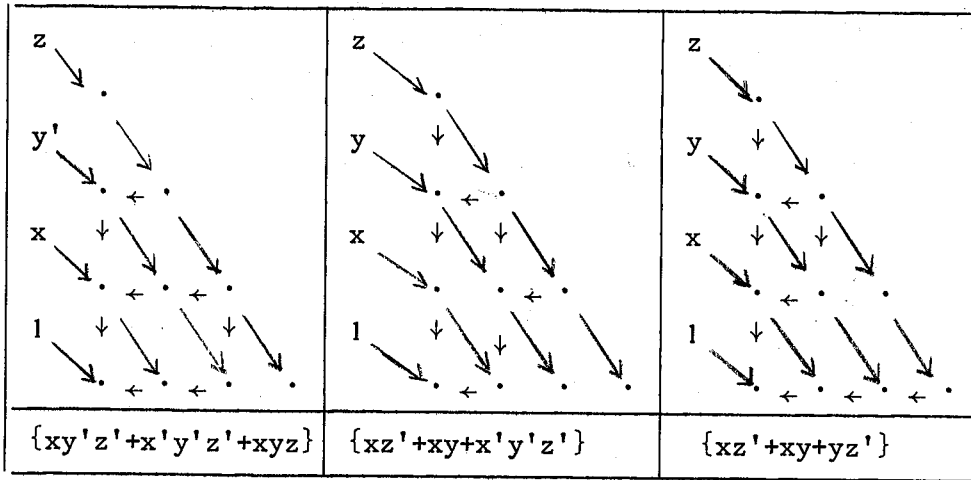
VII-5-3 - Fonctions de trois variables ( $n = 3$ )

Nous aurons 14 classes d'équivalence :



Nous nous occupons seulement des fonctions dépendant des 3 variables. Les autres ont été vues précédemment.





Toutes les synthèses précédentes ont été faites avec l'opérateur

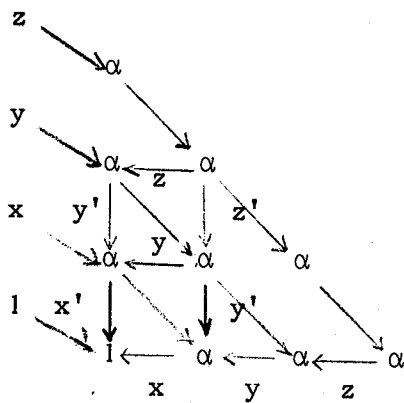
$$\begin{cases} S_1, S_2 = z(u+v) \\ S_3 = z' \end{cases}$$

que nous utilisons pour tous les exemples, sauf précision contraire



Seule la classe  $x \oplus y \oplus z$  n'est pas synthétisable avec cet opérateur.  
Elle l'est cependant avec l'opérateur

$$\left\{ \begin{array}{l} S_1 = z(u+v) \\ S_2 = z'(u+v) \\ S_3 = z. \end{array} \right.$$



VI-5-4 - Fonctions de 4 variables

Nous donnons ci-après la liste des classes de fonctions de 4 variables.  
Pour plus de quatre variables une étude exhaustive n'est pas envisageable  
puisque pour cinq variables nous dépassons le million de classes.

Pour  $n = 4$  nous avons certaines fonctions qui ne sont pas synthétisables  
avec un réseau irrédondant et avec l'opérateur utilisé  $\theta_1$ .

Par exemple :

$$* \quad a'(b\oplus d') + ac(b\oplus d) = a'(b\oplus d') + c(a'\oplus b\oplus d')$$

$$* \quad bc(a+d) + ab'c'd + a'b'(b\oplus c)$$

$$* \quad bcd + (a\oplus b)c'd' + a'b'(c\oplus d)$$

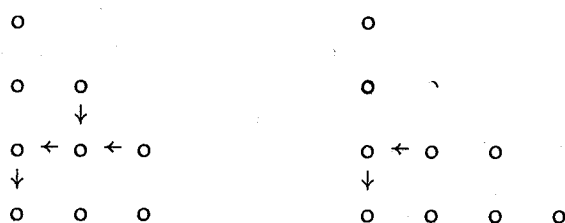
$$* \quad ab'c + a'bd + c'd'(a'\oplus b).$$

#### VI-5-5- Cas général

##### a) Réseau avec ordre de variables imposé

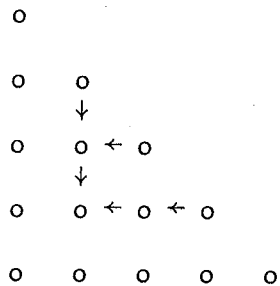
Le réseau comporte  $n(n+1)$  arêtes pouvant être active dans la synthèse (arêtes horizontales et verticales). Il y a donc  $2^{n(n+1)}$  sous-ensembles d'arêtes différents. Bien entendu, ce nombre ne correspond pas au nombre de fonctions synthétisables :

\* 2 sous-ensembles d'arêtes différents peuvent représenter la même fonction :



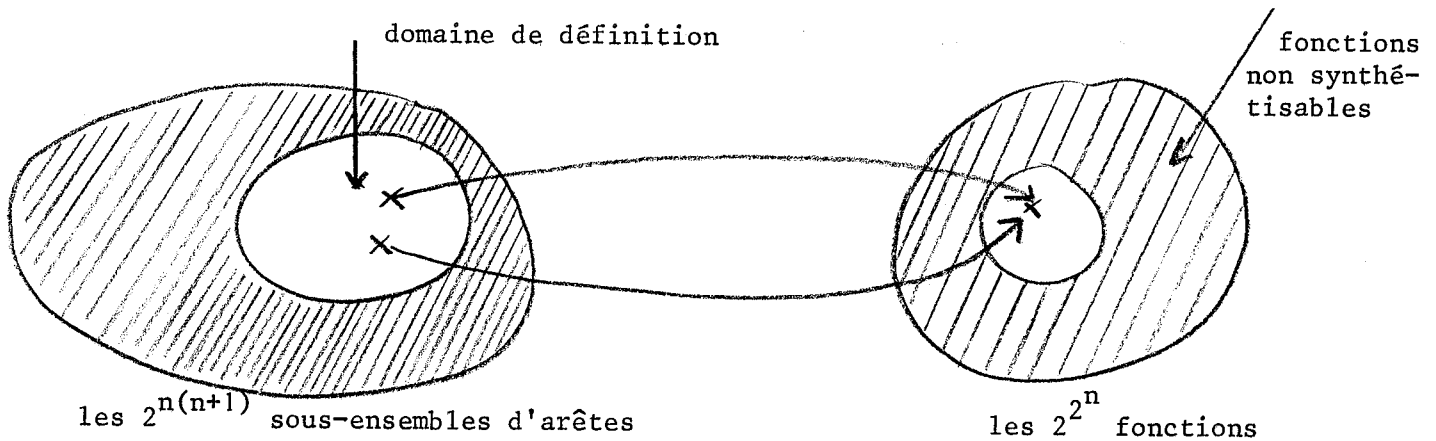
ces deux schémas représentent une même fonction.

\* certains sous-ensembles ne représentent aucune fonction :



La transformation  $f$  qui fait passer d'un sous-ensemble d'arêtes à une fonction booléenne

- 1 - n'est donc pas partout définie
- 2 - n'est pas injective
- 3 - n'est pas surjective.



Pour qu'un ensemble d'arêtes représente une fonction il faut et il suffit

- 1 - qu'il soit connexe
- 2 - que la racine du réseau lui appartienne.

Il y a là un problème de dénombrement que nous ne pousserons pas plus loin.

b) Réseaux avec ordre des variables non imposé

Une même fonction peut être représentée par plusieurs réseaux.

Exemple : un monôme  $\tilde{x}_1 \cdot \tilde{x}_2 \dots \tilde{x}_n$  pourra être représenté par les  $p(n)$  réseaux.

Le problème du dénombrement se complique ainsi avec l'étude des symétries parmi les variables des différentes fonctions.

VI-6 - RESULTATS

L'algorithme a été programmé en PL/1 pour les deux opérateurs  $\theta_1$  et  $\theta_2$ . Des exemples figurent dans les pages suivantes ainsi que le listing.

Nous donnons ci-dessous quelques temps de calcul.

	nombre de variables	nombre de monômes	ordre trouvé	temps en s. 360/67 ASP/MVT
1	6	8	synthèse impossible	1,30
2	6	10	ca'fbd	0,81
3	6	11	cfdea'	0,74
4	6	14	ca'fd'e	0,93
5	6	14	synthèse impossible	1,01
6	10	23	synthèse impossible	12,02
7	10	23	bc'ead'h'jif'	5,44
8	10	27	"	2,24
9	10	27	synthèse impossible	7,46
10	10	32	"	12,28
11	10	35	bc'ead'h'jif'	2,45
12	10	35	"	2,54
13	10	44	"	2,22
14	10	52	"	2,16
15	10	59	"	2,33
16	20	10	abopcdresftghij'klm	3,87
17	20	14	synthèse impossible	32,48
18	20	15	idem 16	3,53
19	20	20	"	3,36
20	20	26	"	3,77
21	20	26	plus de 60 s d'exécution	
22	20	32	"	
23	20	32	idem 16	5,40
24	30	5	abgl'c'defhijklmnopqrstuvwxyz345	2,40

VI-7 - REMARQUES AU SUJET DE  $F^*$  et  $F'$

1 - Si  $F(x_1, \dots, x_n)$  est synthétisable il en sera de même pour sa duale  $F^*(x_1, x_2, \dots, x_n)$  : il suffit d'employer l'opérateur dual  $(S_1^*, S_2^*)$  de l'opérateur  $(S_1, S_2)$  qui a servi à la synthèse de  $F$ .

2 - De même  $F'(x_1, \dots, x_n)$ , la fonction complémentaire de  $F$ , sera aussi synthétisable

\* en prenant comme opérateur  $\begin{cases} S_1^* \\ S_2^* \end{cases}$

et en changeant le réseau  $(1, \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$  en le réseau  $(1, \tilde{x}'_1, \tilde{x}'_2, \dots, \tilde{x}'_n)$ .

\* ou en gardant le même réseau et prenant l'opérateur

$$\begin{cases} S_1^c = S_1^*(u, v, z') \\ S_2^c = S_2^*(u, v, z'). \end{cases}$$

VI-8 - RECONNAISSANCE DE FONCTIONS IDENTIQUES

Soit deux fonctions  $F_1(x_1, \dots, x_n)$  et  $F_2(x_1, \dots, x_n)$ . On veut savoir si ces deux fonctions sont identiques à l'appellation des variables près, c'est-à-dire s'il existe une transformation  $t = \sigma.c$

$\sigma$  étant une permutation des  $n$  variables

et

$c$  étant une complémentation des  $n$  variables,

telle que l'on ait  $F_1 = t(F_2)$ .

En employant cette méthode de synthèse nous pourrions opérer de la façon suivante :

- \* synthétiser  $F_1$  par un réseau  $R_1$
- \* synthétiser  $F_2$  par un réseau  $R_2$ .

Si le réseau  $R_2$  est isomorphe au réseau  $R_1$  on sait que les deux fonctions sont identiques et l'on connaît du même coup la transformation  $t$ .

Si le réseau  $R_2$  n'est pas isomorphe au réseau  $R_1$ , on poursuit l'exploration des choix rendant  $F_2$  synthétisable, jusqu'à obtenir un réseau  $R_2$  isomorphe à  $R_1$ .

Si aucun des réseaux  $R_2$  n'est isomorphe à  $R_1$  on peut en déduire que  $F_1$  et  $F_2$  ne sont pas identiques à une transformation  $t$  près.

Ceci ne convient bien sûr qu'à des fonctions synthétisables pour l'instant.

CHAPITRE - VII

SYNTHESE DE FONCTIONS INCOMPLETES SUR UN RESEAU IRREDONDANT

La fonction à synthétiser  $F(x_1, \dots, x_n)$  est donnée par sa borne inférieure

$$F_{\text{inf}}(x_1, \dots, x_n) = \sum_{i=1}^{i=p} \mu_i = \prod_{j=1}^{j=q} \nu_j$$

et par le complément de sa borne supérieure

$$F'_{\text{sup}}(x_1, \dots, x_n) = \sum_{i=1}^{i=p'} \mu'_i = \prod_{j=1}^{j=q'} \nu'_j$$

La synthèse se complique mais la proportion des  $3^{2^n}$  fonctions incomplètes synthétisables est plus importante que dans le cas précédent.

VII-1 - SYNTHESE D'UNE FONCTION

L'arbre des choix pour l'ordre des variables est le même qu'au chapitre précédent.

La synthèse se fait toujours progressivement couche par couche, sommet par sommet.

Lorsqu'en un sommet il y a incompatibilité le traitement sera différent et l'on ne pourra plus comme précédemment éliminer un des deux ensembles  $\mu_b, \mu_g$  lorsqu'il y aura compatibilité.



Le mécanisme de la synthèse est ainsi alourdi mais il est généralement plus facile de trouver une solution puisque les sommets de l'hypercube sur lesquels la fonction est indéterminée donnent une plus grande liberté d'action.

## VII-2 - L'ALGORITHME

Pour chaque couche ( $i = 1, \dots, n-1$ ) on examine chaque sommet  $\tilde{x}_k$  ( $k = 1, \dots, i+1$ ) et l'on forme

\* l'ensemble  $\mu_g = \{\mu_{g_i} / i = 1, \dots, \alpha\}$  des monômes de  $F_{\text{inf}}$  qui contiennent les lettres  $\tilde{x}_k^{g_i}$  et  $\tilde{x}_{k-1}$

\* l'ensemble  $\mu_b = \{\mu_{b_j} / j = 1, \dots, \beta\}$  des monômes de  $F_{\text{inf}}$  qui contiennent les lettres  $\tilde{x}_k^{b_j}$  et  $\tilde{x}'_{k-1}$

qui passent par le sommet  $\tilde{x}_k$  en question.

- 1 - Si l'on a  $\alpha = \beta = 0$  on passe au sommet suivant.
- 2 - Si l'on a  $\alpha = 0$  et  $\beta \neq 0$   
ou  $\alpha \neq 0$  et  $\alpha = 0$  on ne peut avoir d'incompatibilité et on passe au sommet suivant.
- 3 - Si l'on a  $\alpha \neq 0$  et  $\beta \neq 0$  :
  - 3.1 - Si  $\alpha = \beta$  et si  $\mu_b \vee \mu_g$  forme une constellation on poursuivra la synthèse en passant au sommet suivant, comme précédemment, mais en conservant  $\mu_b$  et  $\mu_g$  pour une raison qui apparaîtra ci-dessous.

3.2. - Si  $\mu_b \vee \mu_g$  ne forme pas une constellation nous allons essayer de rendre compatibles

$$\mu_{g_i} \text{ et } \mu_{b_j} \quad \forall i = 1, \dots, \alpha \text{ et } \forall j = 1, \dots, \beta$$

relativement à un ensemble

$$E = \mu_g \vee \mu_b \vee \mu_a$$

en ajoutant un certain nombre de monômes à  $F_{inf}$ , monômes rangés dans  $\mu_a$ .

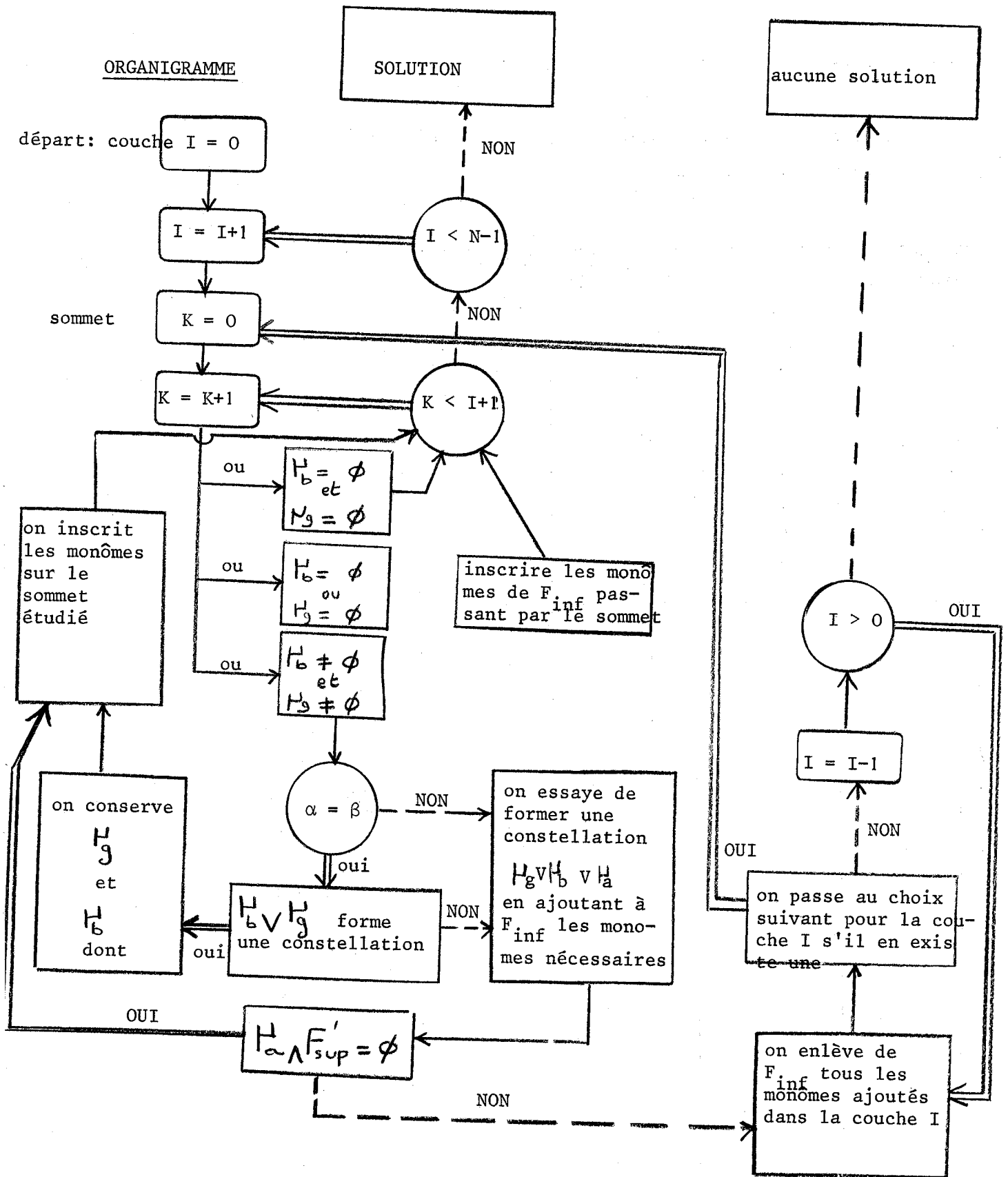
Mais on ne peut rajouter à  $F_{inf}$  que des monômes non couverts par  $F'_{sup}$ . C'est pour cette raison que l'on ne peut plus éliminer un des deux ensembles  $\mu_b, \mu_g$  lorsqu'il y a compatibilité ; il faut disposer de toutes les têtes passant par  $\tilde{x}_k$ .

3.2.1. - Si l'on a à rajouter que des monômes non couverts par  $F'_{sup}$  ( $\mu_a \wedge F'_{sup} = \emptyset$ ),  $F_{inf}$  devient  $F_{inf} \vee \mu_a$  et l'ensemble E formant une constellation, on passe au sommet suivant.

3.2.2. - Si au contraire  $\mu_a \wedge F'_{sup}$  n'est pas vide alors la synthèse n'est pas possible avec l'ordre des variables considéré. Il faut passer au choix suivant, mais auparavant, il convient de supprimer dans  $F_{inf}$  les monômes éventuellement rajoutés sur d'autres sommets de la couche i.

La synthèse se termine lorsque l'on a trouvé un choix convenable jusqu'à la couche (n-1). Sinon, après avoir testé toutes les possibilités, il faut conclure que la synthèse est impossible.

ORGANIGRAMME



VII-3 - RESULTATS

L'algorithme est programmé en PL/1. Le listing figure en appendice.  
 Nous donnons quelques exemples.

	nombre de variables	nombre de monômes de $F_{inf}$	nombre de monômes de $F_{sup}$	nombre de monômes rajoutés à $F_{inf}$	ordre trouvé	temps en s. ASP/MVT
1	6	9	11	2	ab'c'de	0,39
2	6	9	13	3	bacde'	0,58
3	6	9	16	4	cb'a'de'	0,91
4	6	9	11	3	ab'c'de	0,41
5	6	9	14	11	cb'a'de'	1,04
6	6	9	14	11	cb'dea'	3,50
7	6	9	15	7	ceab'd	9,30
8	6	9	16	7	ceb'ad	en CP/CI
8	6	9	17	2	cedfa'	
10	6	9	18	2	cfdea'	
11	6	9	19	8	cfe'b'a'	
12	6	9	20	1	decfa'	
13	6	9	21		synthèse impossible	
14	6	9	5	24	ab'cde	
15	6	9	8	3	ab'c'de	
16	6	9	8	9	ab'c'de'	
17	6	9	8	3	ab'c'df	
18	6	9	8	11	ab'c'd'e	
19	6	9	8	8	ab'c'd'e'	
20	6	9	8	8	ab'c'd'f'	
21	6	9	8	9	ab'c'ed'	
22	6	9	8	9	ab'c'ef'	
23	6	9	8	9	ab'c'fd'	

24	6	9	5	24	ab'cde
25	6	9	5	26	ab'cde'
26	6	9	5	24	ab'cdf
27	6	9	5	23	ab'cd'e

Les exemples 1 - 2 - 3 ont un point commun : ils ont la même borne inférieure. Quant aux compléments des bornes supérieures, ils se déduisent les uns des autres en y ajoutant les monômes ajoutés à  $F_{inf}$  dans l'exemple précédent.

De même les exemples 8 - 9 - 10 --- 13 mais à chaque étape on n'a adjoint à  $F'_{sup}$  qu'un seul des monômes ajoutés à  $F_{inf}$  dans l'étape précédente.

Les exemples 24 à 27 ont trait à la même fonction. On a cherché les ordres successifs la rendant synthétisable :

$$F = (F_{inf}, F'_{sup})$$

avec

$$F_{inf} = a'b'cd(e'\oplus f) + b'c'ef(a'\oplus d) + (ab'cd + a'bcd)(e\oplus f) + abc'd'ef$$

et

$$F'_{sup} = a'b'e'f'(c\oplus d) + a'b'c'(d'e' + def).$$

De même pour les exemples 15 à 23 où l'on a rajouté à  $F'_{sup}$  les monômes  $b'd'ef(a\oplus c) + abc'd'e'f$ .

PROGRAMME DE SYNTHÈSE :  
 FONCTIONS COMPLETES  
 RESEAUX IRREDONDANTS

```

(SUBRG):
JAM95 : PROCEDURE OPTIONS (MAIN) ;
/* SYNTHÈSE DE FONCTION COMPLETE SUR RESEAU IRREDONDANT */
  DCL SYSPRINT FILE STREAM PRINT ENVIRONMENT (F(120)),
    D94 FILE STREAM ENVIRONMENT (F(80)),
    D9 FILE STREAM ENVIRONMENT (F(80));
  GET FILE(D94) LIST(N,M);
BEGIN ;
DCL MONOM(M) CHARACTER(2*N);
DCL VAR(2*N) CHARACTER(2);
DCL PRIM(2*N) CHARACTER(2);
GET FILE(D94) LIST (VAR);
  GET FILE (D9) LIST(MONOM);
PUT EDIT('LES ',N,' VARIABLES DE LA FONCTION')
  (X(20),A(4),F(2),A(25));
PUT SKIP(3);
PUT LIST(VAR);
PUT EDIT('LES ',M,' MONOMES DE LA FONCTION')
  (SKIP(10),X(20),A(4),F(2),A(23));
PUT SKIP(3);
DO I=1 TO M;
  PUT SKIP LIST(MONOM(I));
END;
DO I=2 BY 2 TO (2*N);
PRIM(I)=VAR(I-1); PRIM(I-1)=VAR(I);
END;
DCL VOI(N,N) BIT(M);
DCL UN BIT(M) VARYING;
UN='1'B;
DO I=2 TO M;
UN=UN||'1'B;
END;
VOI(1,1)=UN;
DCL POINTE CHARACTER(4*N) VARYING;
DCL PREZ(2*N) BIT(M);
DO I=1 BY 2 TO (2*N);
DO J=1 TO M;
IF SUBSTR(MONOM(J),I,2)=VAR(I) THEN
DO;
SUBSTR(PREZ(I),J,1)='1'B;
SUBSTR(PREZ(I+1),J,1)='0'B;
END;
ELSE
DO ;
SUBSTR(PREZ(I),J,1)='0'B;
SUBSTR(PREZ(I+1),J,1)='1'B;
END;
END;
END;
END;
/*
  */

```

```

    INCVOI :
      PROCEDURE(IV,JV,RV);
      DCL RV LABEL;
      DO KV=1 TO IV-1;
      IF VOI(IV-1,KV)='0'B THEN GO TO FINKV;
      IF MOD(KV,2)=0 THEN
      BEGIN;
      MV=JV;
      IF MOD(JV,2)=0 THEN LV=JV-1;
      ELSE LV=JV+1;
      END;
      ELSE
      BEGIN;
      LV=JV;
      IF MOD(JV,2)=0 THEN MV=JV-1;
      ELSE MV=JV+1;
      END;
      /*
      /*
      DCL TEST1 ENTRY RETURNS(FIXED BINARY);
      TEST1 : PROCEDURE(UT) FIXED BINARY;
      DCL UT BIT(M);
      LT=0;
      DO I=1 TO M;
      IF SUBSTR(UT,I,1)='1'B THEN LT=LT+1;
      END;
      RETURN(LT);
      END TEST1;
      /*
      /*
      VOIE :
      PROCEDURE(KW,LW);
      DCL UW BIT(M);
      DCL QUEUES CHARACTER(2*M*N) VARYING;
      UW=BOOL(VOI(IV-1,KV),PREZ(LW),'0001'B);
      IF UW='0'B THEN GO TO VOIEND;
      IF VOI(IV,KW)='0'B THEN VOI(IV,KW)=UW;
      ELSE BEGIN;
      IF TEST1(UW)=TEST1(VOI(IV,KW)) THEN GO TO RV;
      QUEUES='';
      DO MW=1 TO M;
      IF SUBSTR(VOI(IV,KW),MW,1)='0'B THEN GO TO FIMW;
      DO MW=1 BY 2 TO (2*N);
      IF INDEX(POINTE,VAR(MW)) =0 THEN
      QUEUES=QUEUES||SUBSTR(MONOM(MW),MW,2);
      END;
      FIMW : END;
      DCL QUEUE CHARACTER(2*N) VARYING;
      DO MW=1 TO M;
      IF SUBSTR(UW,MW,1)='0'B THEN GO TO FIMW;
      QUEUE='';
      DO MW=1 BY 2 TO (2*N);
      IF INDEX(POINTE,VAR(MW)) =0 THEN
      QUEUE=QUEUE||SUBSTR(MONOM(MW),MW,2);
      END;
      IF INDEX(QUEUES,QUEUE)=0 THEN GO TO RV;

```

```

FIMW : END;
END;
VOIEND : END VOIE;
/*
/*
CALL VOIE(KV, LV);
CALL VOIE(KV+1, MV);
  FINKV : END;
  END INCVOI;
  DCL INCVOI ENTRY(FIXED BINARY, , );
/*
/*
PUT PAGE;
VERIFI : PROCEDURE(IVE, JVE, KVE, RVE, SVE);
DCL (RVE, SVE) LABEL ;
POINTE=SUBSTR(POINTE, 1, 4*(IVE-2));
JVE=JVE+KVE;
IF JVE>(2*N) THEN GO TO SVE;
IF INDEX(POINTE, VAR(JVE))=0 THEN GO TO RVE;
POINTE=POINTE||VAR(JVE)||PRIM(JVE);
DO J=1 TO IVE;
VOI(IVE, J)='0'B;
END;
END VERIFI;
DCL VERIFI ENTRY(FIXED BINARY,,FIXED BINARY,,);

/* */
/* */
/* */
/* */

I1=-1;
IUN :
CALL VERIFI(2, I1, 2, IUN, IMPOSS);
CALL INCVOI(2, I1, IUN);
I2=0;
IDEU :
CALL VERIFI(3, I2, 1, IDEU, IUN);
CALL INCVOI(3, I2, IDEU);
I3=0;
ITROI :
CALL VERIFI(4, I3, 1, ITROI, IDEU);
CALL INCVOI(4, I3, ITROI);
PUT EDIT('SOLUTION : ', POINTE)(PAGE, A(11), A);
GO TO FINI;
IMPOSS :
PUT SKIP;
PUT EDIT('AUCUNE SOLUTION')(PAGE, A);
FINI :
END ;
END JAM95;
EOF:

```



CHAPITRE - VIII

CAS DES FONCTIONS NON SYNTHETISABLES PAR UN RESEAU IRREDONDANT

Lorsqu'une fonction  $F(x_1, \dots, x_n)$  n'a pas pu être synthétisée sur un réseau  $\triangle$ , trois possibilités s'offrent alors :

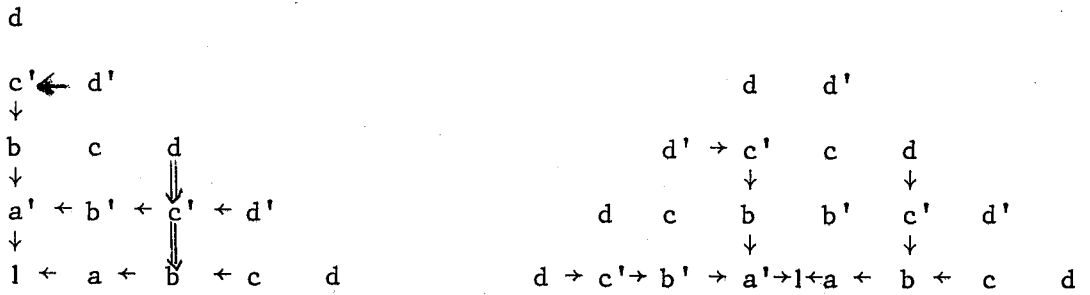
- 1 - passer à un réseau plus important  $\triangle$  ou  $\diamond$
- 2 - chercher à synthétiser sur deux réseaux
- 3 - considérer un réseau  $\triangle$  redondant.

VIII-1 - RESEAU  $\triangle$  OU  $\diamond$

Le nombre de fonctions synthétisables s'accroît avec ces réseaux puisque le nombre de chemins indépendants est accru. Il est donc licite de considérer un de ces réseaux lorsque le réseau  $\triangle$  est insuffisant.

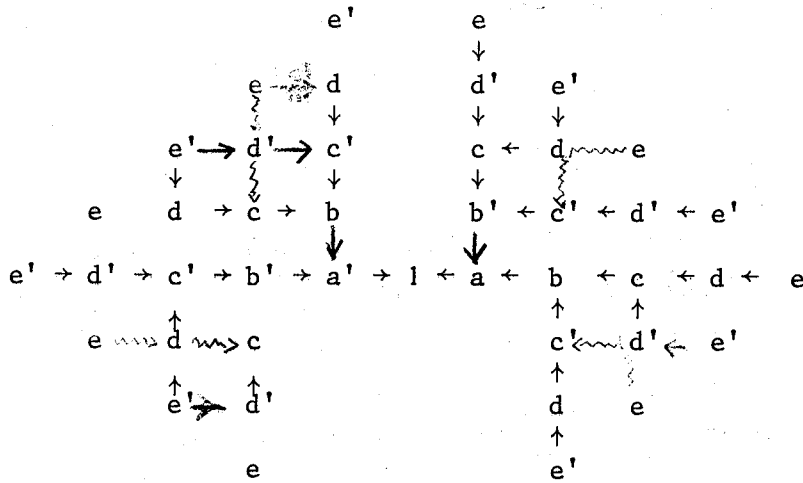
Les méthodes de synthèse à mettre en oeuvre sont identiques à celles précédemment exposées. On finira par s'y heurter aux mêmes difficultés que dans le réseau  $\triangle$  car toute fonction n'y sera pas synthétisable.

Exemple :  $F(a,b,c,d) = abc + a'b'c'd' + abc'd + a'bc'd'$  n'est pas synthétisable sur le réseau  $(1,a,b,c,d)$



mais le devient sur le réseau  $\Delta (1,a,b,c,d)$ .

Comme exemple de fonction non synthétisable sur aucun de ces réseaux prenons la clé de parité  $a \oplus b \oplus c \oplus d \oplus e$



VIII-2 - CAS DE DEUX RESEAUX  $\triangle$  INDEPENDANTS

Nous précisons réseaux "indépendants" car on pourrait considérer que deux réseaux  $\triangle$  forment un réseau  $\triangle$ . C'est vrai mais avec une contrainte :

l'ordre des variables dans les deux réseaux ainsi liés est fixé par l'ordre des variables dans un seul des deux réseaux.

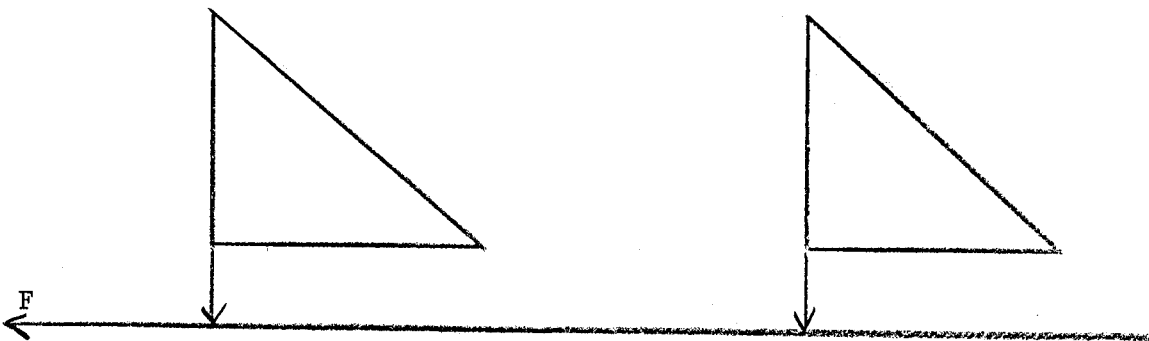
Au contraire, lorsque nous parlons de réseaux indépendants, l'ordre des variables dans un réseau ne dépend pas de l'ordre existant dans l'autre réseau.

Chacun des deux réseaux a donc son propre arbre des choix, celui du second vient se greffer sur chaque noeud du premier, et à priori il y a un nombre de choix extrêmement élevés :

$$(n!2^{n-2})^2.$$

Certains monômes peuvent être synthétisés sur chacun des deux réseaux s'ils sont nécessaires à la compatibilité. On a ainsi sur le second réseau une fonction incomplète.

Cette méthode mettant en jeu un nombre de choix considérable, il faut la concevoir d'une façon heuristique pour espérer l'exploiter raisonnablement.



VIII-3 - RESEAU  $\triangleleft$  REDONDANT

C'est cette possibilité que nous développerons dans le chapitre suivant, pour pallier aux insuffisances du réseau irrédondant.

Au réseau  $\triangleleft$  irrédondant on ajoute une couche redondante chaque fois qu'une impossibilité se manifeste.

PROPRIETE

*Ce réseau redondant est universel. Il permet la synthèse de toute fonction.*

Cette propriété est évidente, puisqu'en rajoutant suffisamment de colonnes, à la limite on obtient un réseau semblable à ceux des paragraphes I.2.3.2 qui représentent l'immersion dans un quadrillage d'un arbre dichotomique complet.



CHAPITRE IX

SYNTHESE SUR UN RESEAU  REDONDANT

A - CAS DES FONCTIONS COMPLETEES

IX-1 - LE PROBLEME

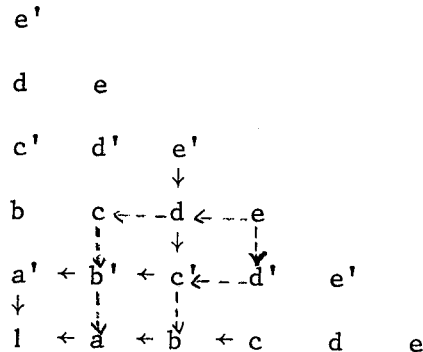
Il peut être envisagé de deux façons différentes.

- 1 - Tous les chemins représentant un monôme de F sont activés sur un réseau irrédondant. On introduit ainsi des monômes non couverts par la fonction et il va falloir "neutraliser" les chemins correspondants, c'est-à-dire corriger la fonction synthétisée  $F_s$  par une certaine fonction correctrice  $F_r$  telle que l'on ait

$$F = F_s \cdot F_r.$$

Exemple :  $F(a,b,c,d,e) = abc + a'b'c'de' + ab'cde + abc'd'e$ .

Nous synthétisons cette fonction, sans nous occuper des incompatibilités, sur le réseau irrédondant (1,a,b,c,d)



En fait nous obtenons ainsi une fonction de sortie

$$F_s(a,b,c,d,e) = b'c'd+abd+abe+ac'd+ac'e+abc.$$

1.1 - Il reste maintenant à déterminer la fonction correctrice  $F_r$  déterminée par l'équation

$$F(a,b,c,d,e) = F_s(a,b,c,d,e) \cdot F_r(a,b,c,d,e)$$

qui s'écrit encore

$$F \cdot (F'_s + F'_r) + F'_s \cdot F_r = 0$$

soit

$$F \cdot F'_s + F \cdot F'_r + F'_s \cdot F_r = 0.$$

Nous avons évidemment  $F_s \geq F$  puisque  $F_s$  couvre tous les monômes de  $F$  par construction du réseau. Ceci se traduit aussi par l'égalité équivalente :

$$F \cdot F'_s = 0.$$

L'équation se réduit alors à

$$F \cdot F'_r + F'_s \cdot F_r = 0$$

et est équivalente au système

$$\begin{cases} F \cdot F'_r = 0 \\ F'_s \cdot F_r = 0 \end{cases}$$

qui peut encore s'écrire

$$\boxed{F \leq F_r \leq F + F'_s}$$

Il y a une solution évidente  $F_r = F$  mais qui est à rejeter puisque  $F$  n'est pas synthétisable. Nous sommes ramenés à la synthèse d'une fonction incomplète avec

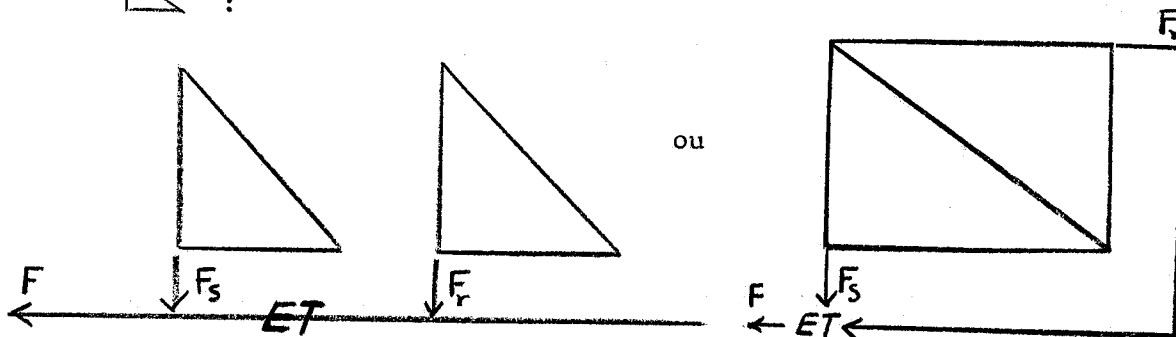
$$F_r = (F_{inf}, F'_{sup})$$

et

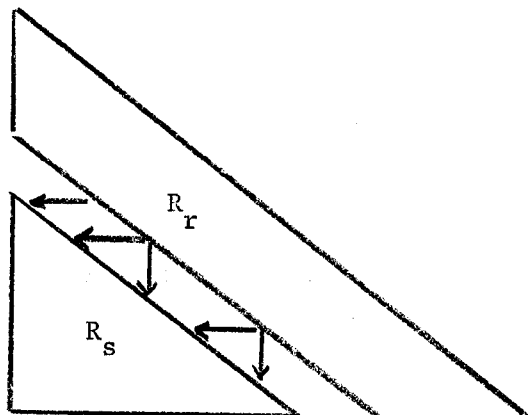
$$\begin{cases} F_{inf} = F \\ F_{sup} = F + F'_s \end{cases}$$

Mais cette fonction incomplète  $F_r$  peut très bien à son tour ne pas être synthétisable sur un réseau irrédondant et par suite nécessiter un réseau correcteur.

Une telle synthèse conduirait à un réseau comprenant au moins deux modules

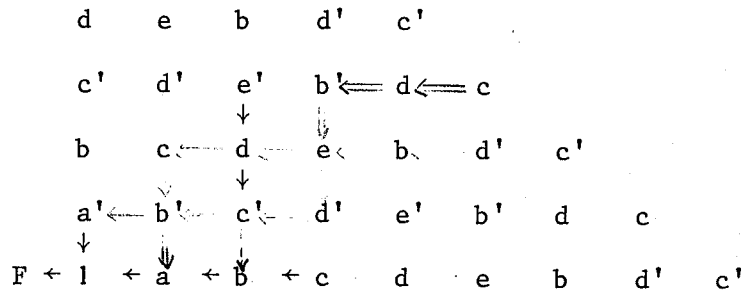


1.2 - Une autre façon de concevoir la correction du réseau irrédondant  $R_s$  synthétisant  $F_s$  est symbolisée par la figure ci-dessous :





Au réseau  $R_s$  nous ajoutons des couches redondantes qui corrigerons la fonction. Sur l'exemple précédent, nous obtenons le réseau redondant (1, a, b, c, d, e, b, d', c')



2 - Mais nous utiliserons une autre méthode. Celle que l'on vient d'exposer reporte les difficultés sur le circuit correcteur.

Ici nous ne séparerons plus les deux fonctions du réseau

- \* élaboration
- \* correction

les incompatibilités seront traitées au fur et à mesure de leur apparition. La synthèse se fait comme pour un réseau irrédondant tant qu'il n'y a pas d'incompatibilité. Si sur la couche  $i$  nous tombons sur une incompatibilité, nous ajouterons une couche redondante  $(i+1)$ .

#### IX-2 - MISE EN OEUVRE DE LA SECONDE METHODE

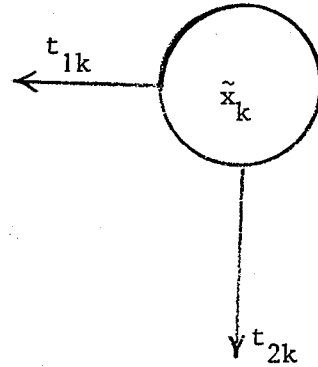
Un nouveau problème se pose : le choix de la variable affectée à la couche redondante qui doit permettre de dissocier les deux chemins de têtes  $t_{1k}$  et  $t_{2k}$  qui correspondent à des monômes incompatibles.

Il y a toujours une solution évidente :

$$x_{k+1} = \tilde{x}_{k-1}$$

car on a obligatoirement

$$\begin{cases} T_{1k} \cdot \tilde{x}_{k-1} = 0 \\ T_{2k} \cdot \tilde{x}_{k-1} = 0. \end{cases}$$



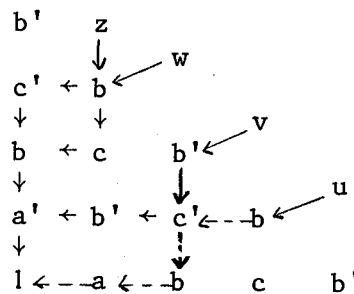
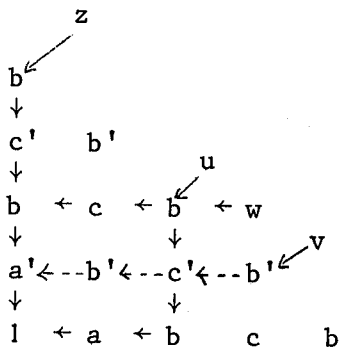
Mais la couche (k+1) redondante peut à son tour présenter des incompatibilités et nécessiter l'addition d'une couche (k+2) redondante.

Par la suite nous verrons qu'il se peut que nous revenions sur le choix de la variable affectée à la couche (k+1). Il faut donc que nous dressions la liste des variables affectables à la couche redondante (k+1).

Exemple :  $F_1(a,b,c,d,e,f) = abc'u + a'b'c'v + a'bcw$  avec  $u, v, w$  fonctions de  $d, e$  et  $f$ , différentes deux à deux.

Le réseau redondant  $(l,a,b,c,b)$  ne convient pas ( $u \neq w$ ).

Le réseau  $(l,a,b,c,b')$  conviendra



Mais pour  $F_2 = F_1 + a'bc'z$  avec  $z \notin \{u,v,w\}$  aucun des deux réseaux ne conviendra.

DEFINITION

Nous appellerons variables affectables à la couche redondante (i+1) les variables affectées aux couches j (j = 1, ..., i-1) telles que l'on ait pour tout les sommets  $\tilde{x}_k$  (k = 1, ..., i+1) où il y a incompatibilité

$$T_{1k} \cdot \tilde{x}_k = 0$$

$$T_{2k} \cdot \tilde{x}'_k = 0.$$

Exemple : pour  $F_1$  nous avons  $\tilde{x}_5 \in \{a, a', c, c'\}$  dans le réseau (1, a, b, c, b).

Remarque - Nous l'avons déjà dit précédemment, il est certain qu'en ajoutant suffisamment de couches on pourra synthétiser toute fonction.

Mais nous aurons bien vite un nombre de couches exagéré, l'ordre des variables ne jouant d'ailleurs aucun rôle, comme dans l'arbre complet. Aussi nous fixerons une contrainte : nous nous limiterons à l'addition de S couches redondantes entre deux couches redondantes.

IX-3 - L'ALGORITHME

Nous parcourons comme auparavant chaque sommet de chaque couche.

A la i<sup>ème</sup> couche pour le sommet  $\tilde{x}_k$  (k = 1, ..., i+1) on forme

\*  $\mu_g = \{\mu_{g_i} / i = 1, \dots, \alpha\}$  l'ensemble des monômes contenant les lettres  $\tilde{x}_{k-1}$  et  $\tilde{x}_k$ .

\*  $\mu_b = \{\mu_{b_j} / j = 1, \dots, \beta\}$  l'ensemble des monômes contenant les lettres de  $\tilde{x}'_{k-1}$  et  $\tilde{x}'_k$

qui passent par le sommet  $\tilde{x}_k$ .

- 1 - Si  $\alpha = \beta = 0$  on passe au sommet suivant.
- 2 - Si  $\alpha = 0$  et  $\beta \neq 0$  ou  $\beta = 0$  et  $\alpha \neq 0$  on inscrit les monômes sur le sommet traité et on passe au suivant.
- 3 - Si  $\alpha \neq 0$  et  $\beta \neq 0$ 
  - 3.1. - si  $\mu_g \vee \mu_b$  forme une constellation on inscrit comme dans le cas des fonctions incomplètes les monômes de  $\mu_b$  et  $\mu_g$  passant par le sommet et on fait  $k = k+1$ .

Nous sommes obligés de conserver les deux ensembles pour pouvoir déterminer le cas échéant la liste des variables affectables à une couche redondante ultérieure.

- 3.2. - si  $\mu_g \vee \mu_b$  ne forme pas une constellation il faut alors envisager l'adjonction d'une couche redondante.

3.2.1. - Si la couche  $i$  n'est pas redondante.

3.2.1.1. - Si c'est la première valeur de  $k$  pour laquelle on envisage une  $(i+1)^{\text{ème}}$  couche redondante, on dresse la liste  $L_{i+1}$  des variables affectables.

3.2.1.2. - Si ce n'est pas la première fois on ôte éventuellement de  $L_{i+1}$  les variables qui ne donnent pas satisfaction au sommet  $\tilde{x}_k$  considéré et on passe au sommet suivant.

— Si  $L_{i+1}$  n'est pas vide on passe à la couche  $(i+1)$ . Sinon il faut considérer le choix suivant pour la couche  $i$ .

3.2.2. - La couche  $i$  est redondante. On calcule le nombre de couches redondantes adjacentes deux à deux ajoutées depuis la dernière couche redondante.

3.2.2.1. - Ce nombre est inférieur à S et nous tombons dans le cas du paragraphe 3.2.1.

3.2.2.2. - Ce nombre est égal à S. Nous choisissons dans  $L_i$  le choix suivant parmi les variables affectables. Si  $L_i$  est vide on passe à la couche (i-1) etc...

La synthèse se termine lorsque (n-1) couches irrédondantes ont été traitées avec succès puisque ici comme auparavant la n-ième couche irrédondante ne peut provoquer aucune incompatibilité.

Sinon la synthèse est impossible avec la valeur de S. Il faut choisir une valeur supérieure.

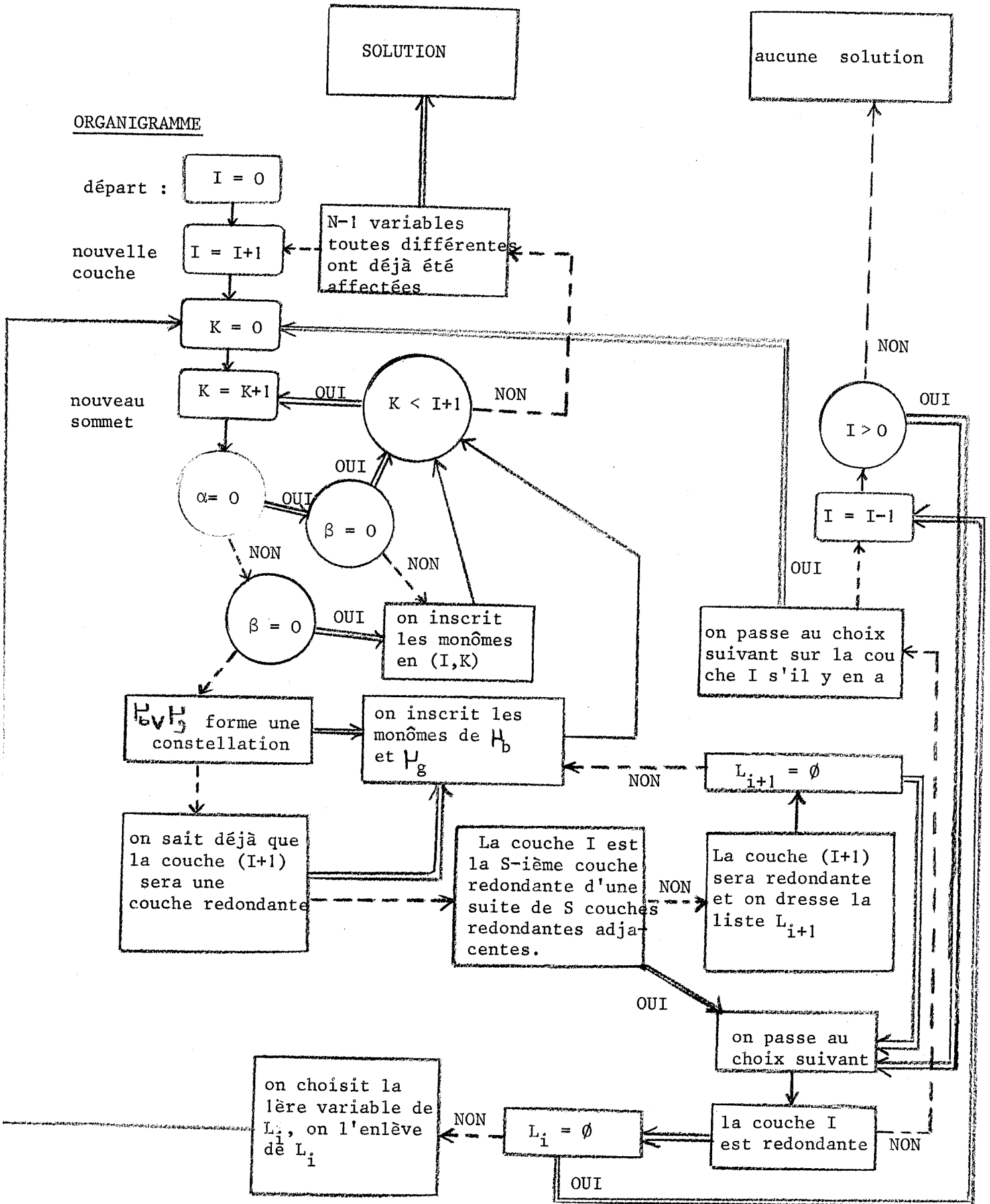
Remarque sur l'arbre des choix

L'arbre des choix contient celui relatif au réseau irrédondant, où entre certains couples de sommets adjacents viennent s'intercaler des sommets correspondants aux couches adjacentes.

Nous traiterons un exemple avec  $S = 1$  :

$$F = abcd'u+a'b'c'd'v+a'b'c'dw+a'bcdz = \sum_{i=1}^4 \mu_i$$

avec u,v,w,z fonctions de e et f et u,v,w,z différents deux à deux.



1 -  $x_1 = a$  : figure 1

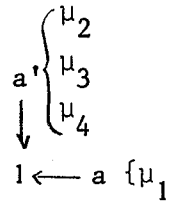


Figure 1

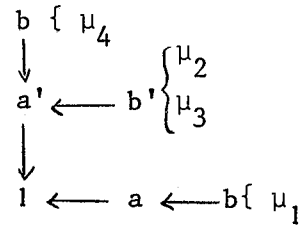


Figure 2

2 -  $x_2 = b$  : figure 2

3 -  $x_3 = c$  : figure 3

4.1.  $x_4 = d$  : figure 4

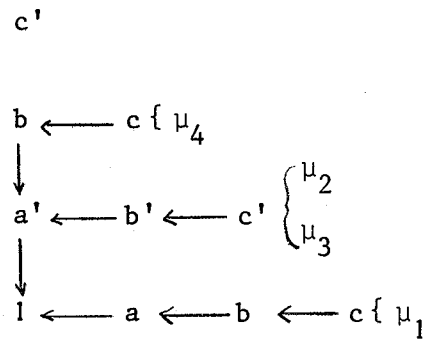


Figure 3

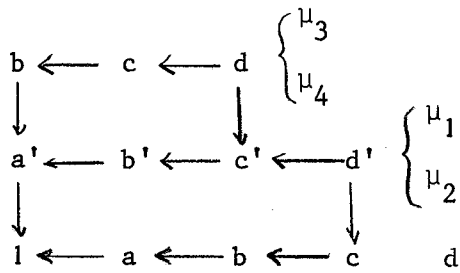
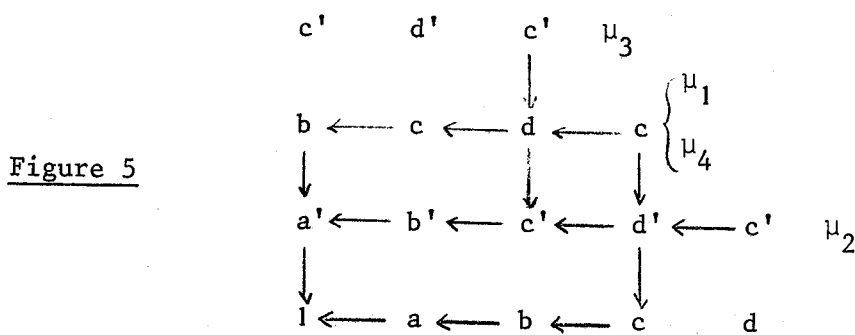


Figure 4

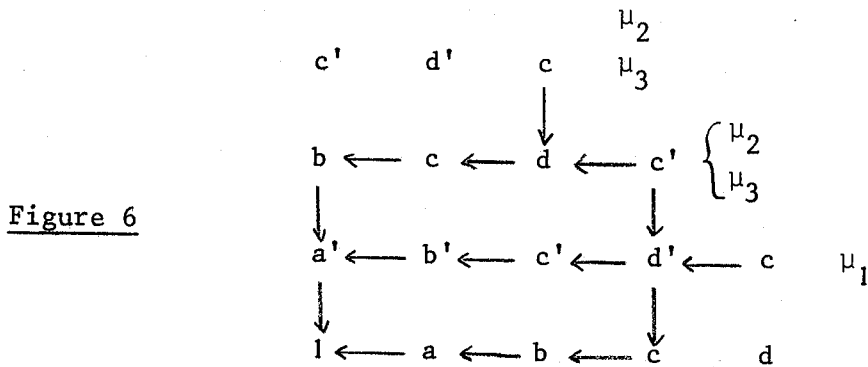
\* au sommet d' apparaissent les monômes incompatibles ( $u \neq v$ )  $\mu_1$  et  $\mu_2$   
 Nous aurons  $L_{5_1} = \{c, c', b, b', a, a'\}$  relativement au sommet d'.

\* au sommet d  $\mu_3$  et  $\mu_4$  incompatibles donnent  $L_{5_2} = \{c, c', b, b'\}$  d'où  
 $L_5 = L_{5_1} \wedge L_{5_2} = \{c, c', b, b'\}$

5.1.  $x_5 = c$  : figure 5 il y a incompatibilité entre  $\mu_1$  et  $\mu_4$



5.2.  $x_5 = c'$  : figure 6 cette fois il y a incompatibilité entre  $\mu_2$  et  $\mu_3$



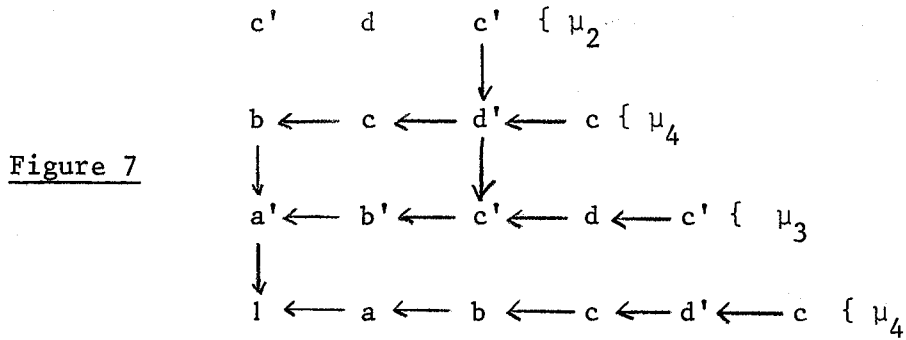
5.3.  $x_5 = b$ .  $\mu_1$  et  $\mu_4$  sont incompatibles.

5.4.  $x_5 = b'$   $\mu_2$  et  $\mu_3$  sont incompatibles.

$L_5$  étant alors épuisé il nous fait revenir sur la quatrième couche :



4.2.  $x_5 = c$  : figure 7 cette solution convient et la synthèse peut alors se poursuivre sur la 6ème couche



IV-4 - CONCLUSIONS

Nous avons programmé l'algorithme en PL/1 et traité des exemples avec succès pour  $S = 1$ .

Dans le cas général le réseau aura  $(n-1)(s+1)+2$  couches. Il serait intéressant d'évaluer la valeur minimale de  $S$  pour laquelle le réseau serait universel.

Avec  $S = 1$  nous avons obtenu le nombre de couches maximal pour les clés de parité :

$$\begin{array}{l}
 x_1 \oplus x_2 \oplus \dots \oplus x_n \\
 x_1 \oplus x_2 \oplus \dots \oplus x_n \oplus 1
 \end{array}$$

Le listing figure en appendice. Nous donnons quelques exemples à la fin de ce chapitre.

B - CAS DES FONCTIONS INCOMPLETES

La méthode peut être adaptée sans grandes difficultés aux fonctions incomplètes, mais nous n'avons pas programmé l'algorithme.

Exemple :

1 -  $F_1 = a \oplus b \oplus c \oplus d \oplus e$

solution :

abacadae

temps :

2 s.

2 -  $F_2 = F_1 + a'b'c'de'$

solution :

abacad'ae

temps :

1,40 s.

3 -  $F_3 = F_2 + a'b'cd'e'$

solution :

abacadae

temps :

0,89 s.

4 -  $F_4 = bce'(a \oplus d) + a'c'd'(b' \oplus e) + abc'de$

solution :

abdbca'e

temps :

1,10 s.



## CHAPITRE - XII

### SYNTHESE DE FONCTIONS CROISSANTES

#### XII-1 - THEOREME D'AMELIORATION EN SYNTHESE GENERALE

L'idée de M. Kuntzmann est d'utiliser le théorème d'amélioration en synthèse générale dans le cas des fonctions croissantes :

*Soit un réseau R admettant  $R_1$  comme partie queue et  $R_2$  comme amélioration de  $R_1$ . On peut construire à partir de  $R_2$  une amélioration de R.*

*On supprime de R la partie queue  $R_1$  et on met à sa place le réseau  $R_2$  en le connectant aux opérateurs de  $R - R_1$  de la manière suivante : si un opérateur de  $R - R_1$  avait pour entrée la sortie d'un opérateur  $N_1$  de  $R_1$ , on remplace celle-ci par la sortie de l'opérateur  $N_2$  de  $R_2$ .*

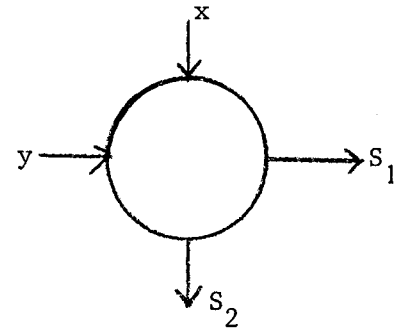
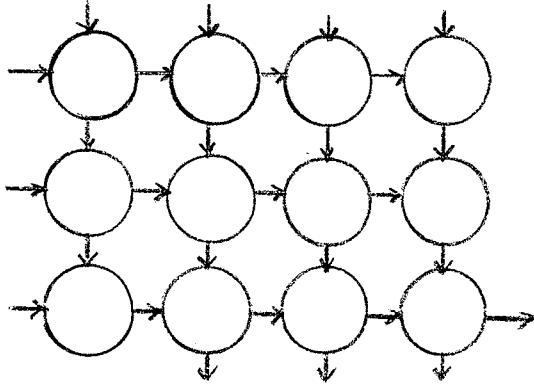
*Cette transformation*

- ne fait pas croître l'indicateur de coût*
- on démontre de proche en proche, pour tous les opérateurs de R, que leur sortie a été améliorée.*

#### XII-2 - RESEAU ET OPERATEURS UTILISES

Nous travaillons toujours dans le cadre des synthèses cellulaires, aussi le réseau utilisé est-il formé de cellules placées en lignes et colonnes.

Ces cellules sont reliées entre elles horizontalement et verticalement, chaque cellule étant liée à ses quatre voisines.



Chaque cellule comporte deux entrées (x,y) et deux sorties (S<sub>1</sub>,S<sub>2</sub>) pouvant produire toutes les fonctions croissantes de deux variables :

$$S_1, S_2 \in \{x, y, xy, x+y\}$$

Les entrées du réseau (les variables de la fonction à synthétiser) attaquent les portes situées sur le côté gauche et le bord supérieur du réseau.

### XII-3 - PRINCIPE DE LA METHODE

La fonction à synthétiser est élaborée par étapes et chaque stade on obtient une approximation optimale de la fonction en minimisant la distance algébrique entre la fonction à réaliser et l'approximation obtenue.

Une fonction  $F(x_1, \dots, x_n)$  est représentée par le couple  $(F_{\text{inf}}, F'_{\text{sup}})$  formé par sa borne inférieure et le complément de sa borne supérieure

$$F_{\text{inf}} = \sum_{i=1}^{M_{\text{inf}}} \mu_i$$

$$F_{\text{sup}} = \sum_{j=1}^{M_{\text{sup}}} \nu_j$$

avec

$$\mu_i = x_{i_1} x_{i_2} \dots x_{i_k}$$

$$\nu_j = x'_{j_1} x'_{j_2} \dots x'_{j_\ell}$$

et

$$i_m \in \{1, 2, \dots, n\} \quad \forall m = 1, \dots, k$$

$$k \leq n$$

$$j_m \in \{1, 2, \dots, m\} \quad \forall m = 1, \dots, \ell$$

$$\ell \leq n.$$

F sera représentée par un vecteur ligne formé de  $M = (M_{\text{inf}} + M_{\text{sup}})$  bits.  
Soit  $F_i$  le  $i^{\text{ème}}$  bit de F, nous aurons

$$F_i = 1 \quad \forall i = 1, \dots, M.$$

$\varphi$  étant une approximation de F nous définirons la distance algébrique de  $\varphi$  à F par

$$d(F, \varphi) = \sum_{i=1}^M |F_i - \varphi_i|$$

Comme  $F_i = 1$  quelque soit  $i$  nous aurons

$$d(F, \varphi) = M - \sum_{i=1}^{i=M} \varphi_i$$

Nous aurons :

$$\begin{aligned} \text{pour } i \leq M_{\text{inf}} & \quad \begin{cases} \varphi_i = 1 \\ \varphi_i = 0 \end{cases} \quad \begin{array}{l} \text{si } \varphi \geq \mu_i \\ \text{sinon} \end{array} \\ \\ \text{pour } i > M_{\text{inf}} & \quad \begin{cases} \varphi_i = 1 \\ \varphi_i = 0 \end{cases} \quad \begin{array}{l} \text{si } \varphi' \geq \nu_i \\ \text{sinon} \end{array} \end{aligned}$$

La fonction peut être donnée par un tableau de M colonnes et n lignes correspondant aux n variables.

Exemple :  $F(x,y,z) = xy+yzt$

$$\begin{aligned} F_{\text{inf}} &= xy+yzt & M_{\text{inf}} &= 2 \\ F'_{\text{sup}} &= y'+x'z'+x't' & M_{\text{sup}} &= 3 \end{aligned}$$

d'où le tableau ci-dessous :

x	1	0	0	1	1
y	1	1	1	0	0
z	0	1	0	1	0
t	0	1	0	0	1
xy	1	0	1	1	1
x+y	1	1	0	0	0

Une nouvelle ligne  $\Psi$  du tableau sera obtenue en faisant la somme ou le produit booléens de deux lignes  $\varphi$  et  $\theta$  :

\*  $\Psi = \varphi + \theta$  sera définie par

$$\Psi_i = \begin{cases} \varphi_i + \theta_i & \text{si } i \leq M_{\text{inf}} \\ \varphi_i \cdot \theta_i & \text{si } i > M_{\text{inf}} \end{cases}$$

\*  $\Psi = \varphi \cdot \theta$  sera définie par

$$\Psi_i = \begin{cases} \varphi_i \cdot \theta_i & \text{si } i \leq M_{\text{inf}} \\ \varphi_i + \theta_i & \text{si } i > M_{\text{inf}} \end{cases}$$

#### XII-4 - REMARQUES SUR LES PROPRIETES DES FONCTIONS CROISSANTES

Nous rappellerons ici quelques propriétés classiques :

- 1 - les monômes premiers de la borne inférieure sont tous obligatoires. Ils couvrent ce que l'on appelle les sommets caractéristiques de 1ère espèce.

Nous n'avons donc aucun problème de recherche de base première irrédondante.

- 2 - de même les monômes premiers de complément de la borne supérieure sont tous obligatoires. Ils couvrent les sommets caractéristiques de 2ème espèce.

$F'_{\text{sup}}$  ne nécessite donc pas une recherche compliquée de base première.



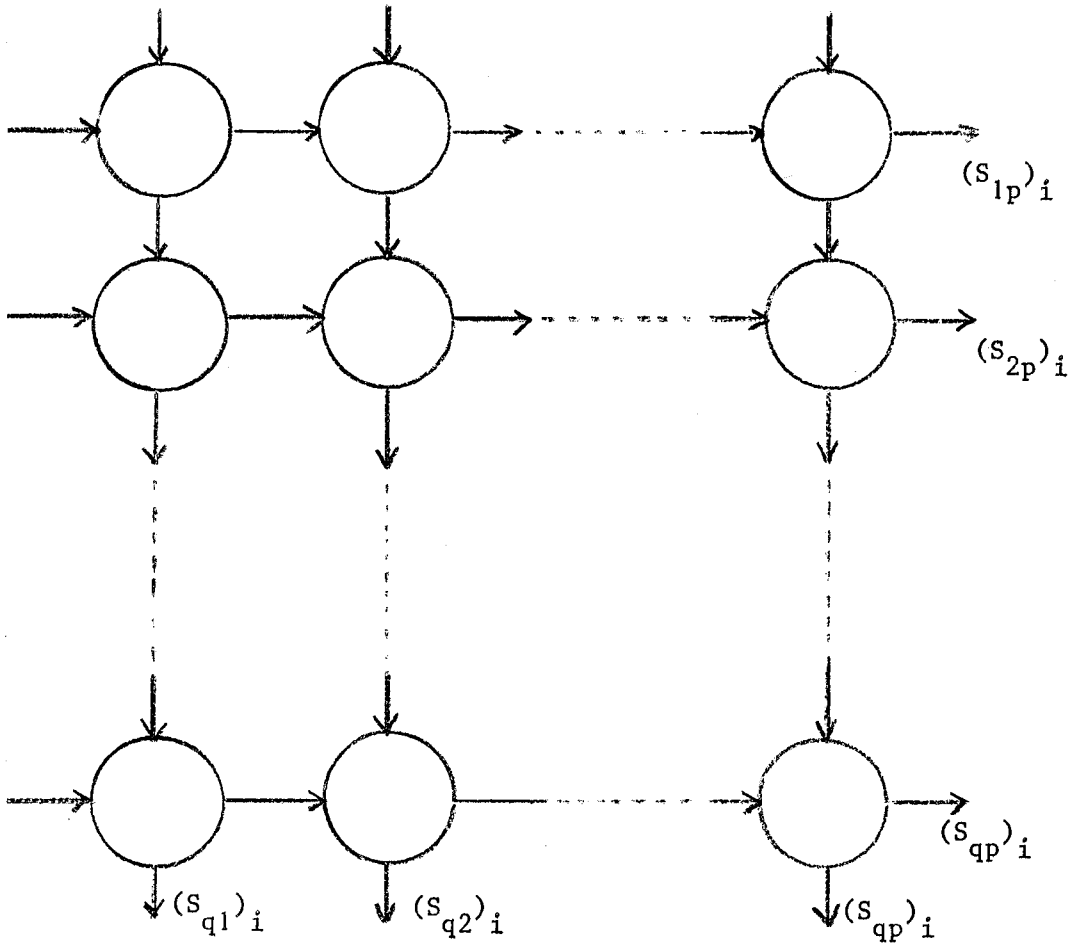
La fonction  $\varphi$  qui sera synthétisée par le réseau devra être compatible avec  $F = (F_{\text{inf}}, F'_{\text{sup}})$ , il faut et il suffit pour cela qu'elle prenne la valeur 1 sur les sommets caractéristiques de première espèce et 0 sur ceux de 2ème espèce.

C'est ce que nous exprimons par :

$$d(\varphi, F) = 0.$$

### XII-5 - ETUDE EXHAUSTIVE

Si l'on cherche la meilleure solution, celle qui nécessite le nombre minimal de cellules, on ne pourra éliminer une partie queue  $R_1$  au profit d'une partie queue  $R_2$  que si toutes les sorties de  $R_2$  sont meilleures que les sorties correspondantes de  $R_1$ .



Partie queue  $R_i$

$R_2$  sera meilleure que  $R_1$  si l'on a

$$(S_{ij})_2 > (S_{ij})_1 \quad \forall i = 1, \dots, q \text{ et } \forall j = 1, \dots, p.$$

Nous rappelons qu'un vecteur ligne  $\varphi$  sera supérieur à un vecteur ligne  $\Psi$  si l'on a :

$$\varphi > \Psi \Leftrightarrow \varphi_i \geq \Psi_i \quad \forall i = 1, \dots, M.$$

Nous serons ainsi amenés à conserver à chaque stade un grand nombre de parties queues concurrentes, ce qui est vite prohibitif en temps calcul et en occupation mémoire.

#### XII-6 - ETUDE HEURISTIQUE

Pour éviter ces inconvénients nous chercherons une "bonne" solution, sans vouloir atteindre absolument la meilleure.

Nous choisissons donc une méthode heuristique en nous bornant à la condition :

$$(S_{qp})_2 > (S_{qp})_1$$

pour décider d'une partie queue  $R_2$  meilleure que  $R_1$ .

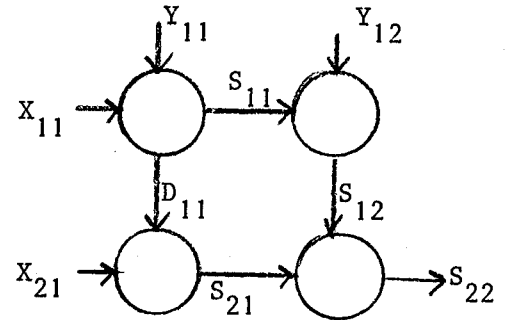
La synthèse consistera ainsi à chercher à chaque étape le minimum de

$$d(F, S_{pq}).$$

XII-7 - PREMIERE ETAPE : OBTENTION DE  $S_{22}$

Nous cherchons tout d'abord la sortie  $S_{22}$  la plus proche de  $F(x_1, \dots, x_n)$ , obtenue par un réseau de 4 opérateurs.

$S_{22}$  peut être fonction de 4, 5 ou 6 variables simples, suivant la configuration des entrées/sorties des quatre cellules.



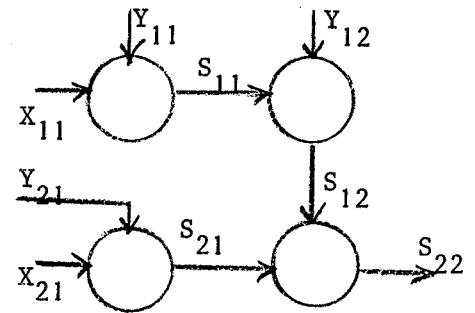
Configuration 3

XII-7-1 - Cas des configurations 1 et 2

Nous obtenons pour  $S_{22}$  toutes les fonctions croissantes :

$$(X_{12} \dot{+} Y_{12}) \dot{+} (X_{21} \dot{+} Y_{21})$$

$$[(X_{11} \dot{+} Y_{11}) \dot{+} Y_{12}] \dot{+} (X_{21} \dot{+} Y_{21})$$



Configuration 2

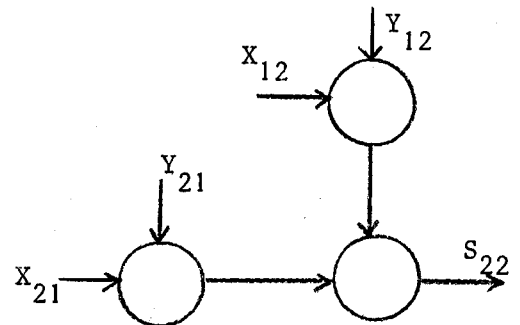
XII-7-2 - Configuration 3

C'est la seule configuration dans laquelle les deux sorties de l'opérateur (1,1) sont utilisées ensemble.

Si nous avons  $S_{11} = X_{11} \dot{+} Y_{11}$  nous aurons la sortie duale  $D_{11} = X_{11} \dot{+} Y_{11}$ . Seules les sorties de la forme

$$S_{22} = (S_{11} \dot{+} Y_{12}) \dot{+} (D_{11} \dot{+} X_{21})$$

sont à étudier, car les fonctions



$$(S_{11} + Y_{12}) + (S_{11} + X_{21}) = \begin{cases} S_{11} \cdot Y_{12} \cdot X_{21} \\ S_{11} \cdot Y_{12} \\ S_{11} \cdot (Y_{12} + X_{21}) \\ S_{11} + Y_{21} \end{cases}$$

sont toutes synthétisables avec au plus trois opérateurs.

XII-7-3 - L'algorithme

Nous cherchons donc à minimiser  $d(F, S_{22})$  en établissant le tableau suivant

	1	.....	M <sub>inf</sub>	.....	M	
$X_1$						n colonnes } T <sub>1</sub>
⋮						
$X_i$						
⋮						
$X_n$						
$P_{ij} = X_i + X_j$						n(n+1) colonnes au plus
$P_{ij} + X_k$						2n(n <sup>2</sup> -n+2) colonnes } T <sub>3</sub> au plus

Tableau 1 : il est formé par la donnée de la fonction

Tableau 2 : ses colonnes représentent toutes les quantités  $(X_i + X_j) / (X_i \cdot X_j)$  maximales c'est-à-dire non inférieures aux colonnes  $X_i$  où à d'autres colonnes  $(X_k + X_l)$

Tableau 3 : on y trouve toutes les quantités  $(X_i \in T_1) + [(X_j + X_k) \in T_2]$  maximales, c'est-à-dire non inférieures aux colonnes des tableaux T<sub>1</sub> et T<sub>2</sub> où à d'autres colonnes de T<sub>3</sub>.

Le choix de  $S_{22}$  correspondra à la colonne ayant le plus de "1", colonne appartenant à l'un des trois ensembles :

1 - (colonne  $\in T_2$ )  $\dagger$  (colonne  $\in T_2$ )  $\rightarrow$  conf. 1

2 - (colonne  $\in T_2$ )  $\dagger$  (colonne  $\in T_3$ )  $\rightarrow$  conf. 2

3 - (colonne  $\in T_3$ )  $\dagger$  (colonne  $\in T_3$ )  $\rightarrow$  conf. 3

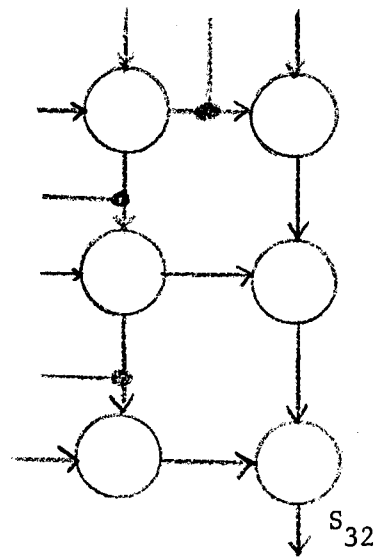
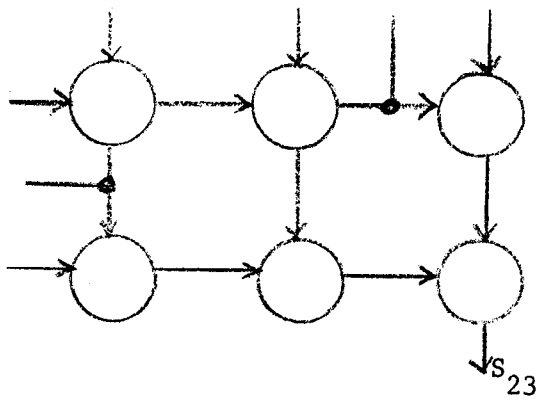
soit  $[(X_i \cdot X_j) \dagger X_k] \dagger [(X_i + X_j) \dagger X_\ell]$

avec  $\ell \neq k$ .

### XII-8 - DEUXIEME ETAPE

XII-8-1 - Bien entendu cette étape n'a lieu que si la précédente n'a pas donné satisfaction, c'est-à-dire, si la colonne de  $S_{22}$  comprend au moins un zéro.

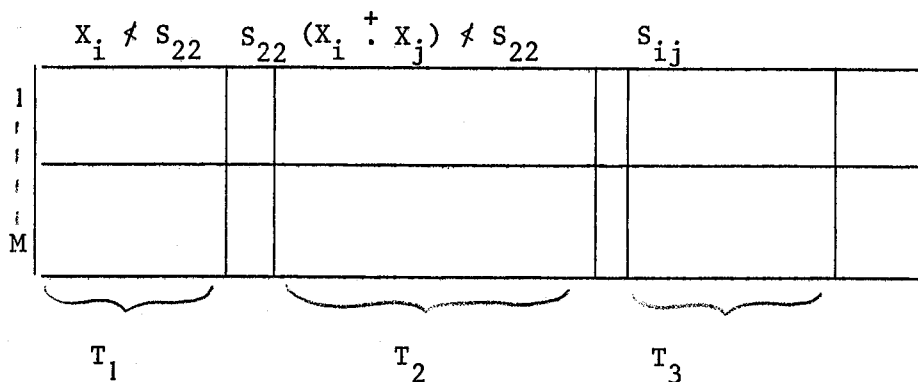
Nous optimisons cette fois la sortie  $S_{32}$  ou  $S_{23}$  suivant le cas :



Notre tableau précédent va subir quelques aménagements :

- 1) dans  $T_1$  nous ne gardons que les variables  $X_i$  non couvertes par  $S_{22}$   
 $X_i \not\prec S_{22}$ .
- 2) idem pour  $T_2$ , ne sont conservés que les  $X_i + X_j$  telles que  
 $X_i + X_j \not\prec S_{22}$ .

Le tableau est donc constitué de ces deux parties et de  $S_{22}$ . Nous allons lui adjoindre une nouvelle troisième partie représentant les fonctions dont nous pouvons disposer sur les bords du réseau après la première étape.



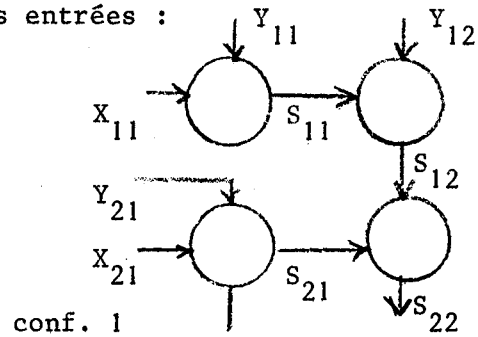
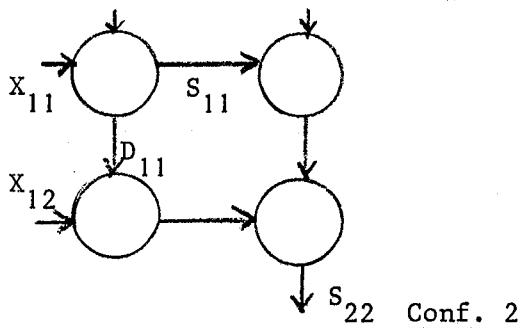
Ces fonctions seront :

$$S_{11} = f(X_{11}, Y_{11})$$

$$S_{12} = f(S_{11}, Y_{12})$$

$$S_{21} = f(S_{11}, X_{21}) \text{ ou } f(X_{21}, Y_{21})$$

Nous pouvons disposer encore, puisque les opérateurs utilisés peuvent produire les 4 fonctions croissantes de leurs entrées :



$$D_{12} = f^*(S_{11}, Y_{12}) \quad \text{la duale de } S_{12}$$

$$D_{21} = f^*(D_{11}, X_{21}) \quad \text{où } f^*(X_{21}, Y_{21}) \text{ duale de } S_{21}$$

$D_{11}$  dans le cas de conf. 1.

Dans  $T_3$  ne seront conservées bien sûr que les colonnes non couvertes par  $S_{22}$ .

XII-8-2 - L'algorithme

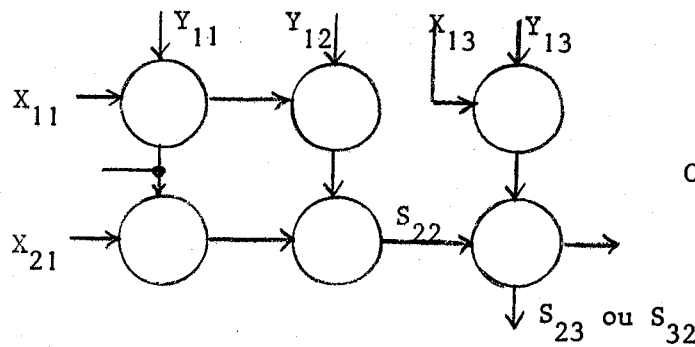
Nous formerons

1) toutes les fonctions  $S_{22} + [(X_i + X_j) \in T_2]$

2) ainsi que toutes les quantités  $S_{22} + [(X_i \in T_1) + (S_{ij} \in T_3)]$

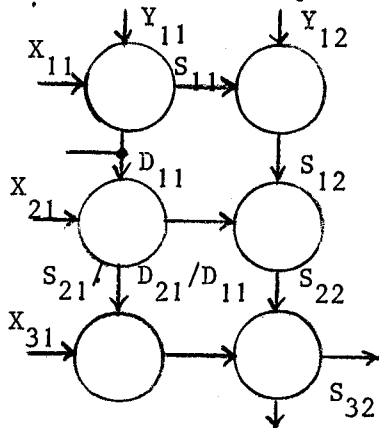
Nous conserverons la meilleure amélioration, c'est-à-dire la fonction ayant un nombre maximal de '1'.

Les fonctions 1) correspondent au schéma suivant :

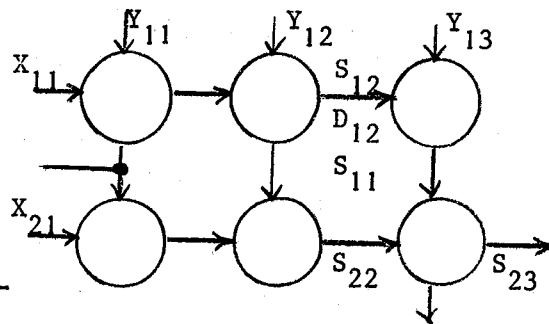


Conf. 3-1 - Conf. 2.1

alors que les fonctions 2) correspondent aux deux possibilités suivantes :



conf. 3.2 -  
conf. 2.2



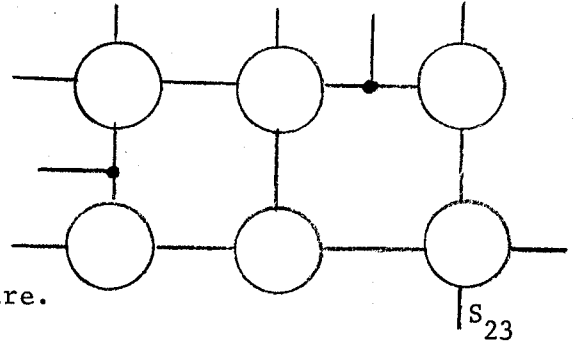
conf. 3.3 -  
conf. 2.3

A l'issue de cette seconde étape, le réseau se présente toujours sous la forme de la figure suivante, en permutant, si nécessaire, les couples d'opérateur

(1,2) et (2,1)

(3,1) et (1,3)

(3,2) et (2,3)



cf. l'exemple n° 1 en fin de chapitre.

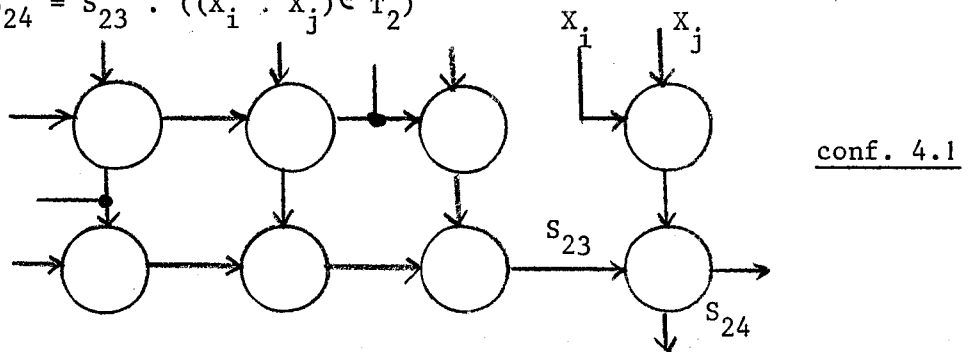
XII-9 - TROISIEME ETAPE

Nous conservons dans  $T_1$  les  $X_i \neq S_{23}$

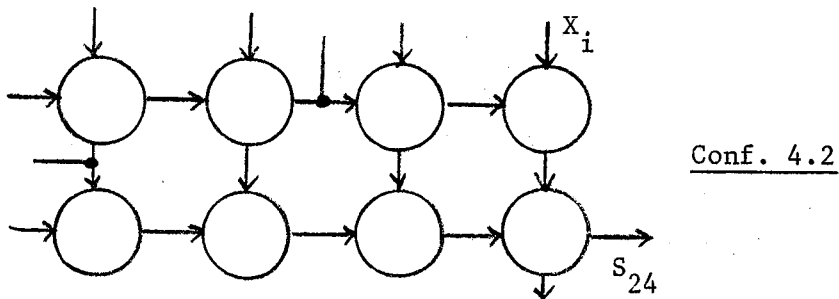
$T_2$  les  $(X_i + X_j) \neq S_{23}$  et nom inférieurs à d'autres colonnes de  $T_2$ .

Nous formerons successivement, pour retenir la meilleure sortie possible :

1 - les fonctions  $S_{24} = S_{23} + ((X_i + X_j) \in T_2)$



2 - les fonctions  $S_{24} = S_{23} + S_{ij}$  avec  $S_{ij} = X_{13} + (X_i \in T_1)$  dans le cas où une des sorties de (1,2) est utilisée comme entrée de (1,3).





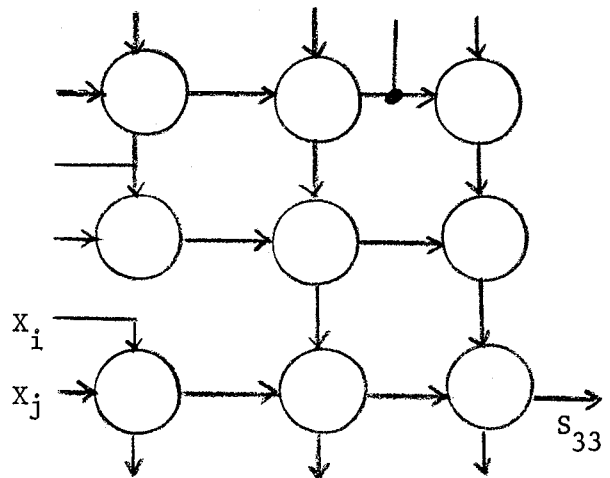
$$S_{ij} = \begin{pmatrix} S_{13} \\ \text{ou} \\ D_{13} \end{pmatrix} \cdot (X_i \in T_1)$$

3 - les fonctions  $S_{33} = \{ [(X_i \cdot X_j) \in T_2] \cdot S_{ij} \} \cdot S_{23}$

avec

$$S_{ij} = \begin{cases} S_{12} \\ S_{21} \\ S_{22} \\ D_{22} \end{cases}$$

Conf. 4.3

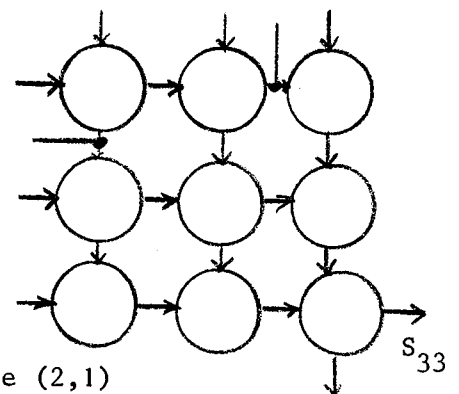


4 - les fonctions  $S_{33} = [(S_{ij})_1 \cdot S_{ij}] \cdot S_{23}$

avec

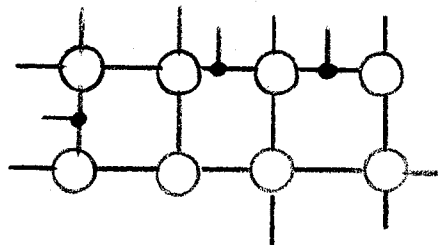
$$(S_{ij})_1 = \begin{cases} S_{21} \\ D_{21} \\ D_{11} \\ \text{(ou)} \cdot (X_i \in T_1) \\ S_{11} \end{cases}$$

si une sortie de (1,1) est utilisée comme entrée de (2,1)

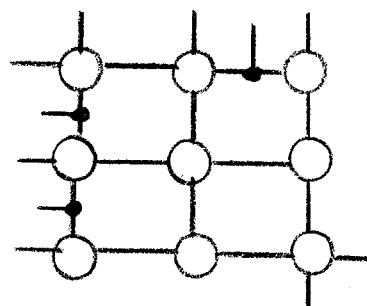


Conf. 4.4

Nous prendrons comme fonction  $S_{33}$  la meilleure approximation de  $F(x_1, \dots, x_I)$ .  
Après cette amélioration le réseau peut avoir deux aspects généraux :



Configuration 2 x 4

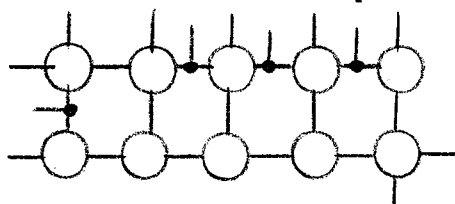


Configuration 3 x 3

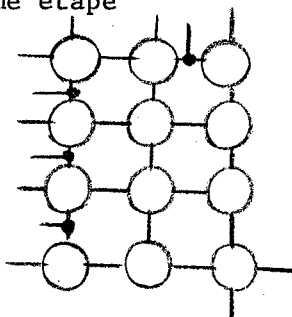
XII-10 - QUATRIEME ETAPE

Nous préciserons encore ce stade : suivant la configuration  $2 \times 4$  ou  $3 \times 3$  par l'étape précédente, il y aura lieu de faire une recherche particulière.

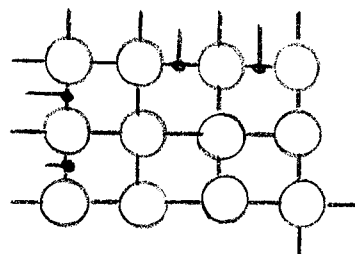
- 1 - Si nous avons à l'entrée la configuration  $2 \times 4$  nous aurons à l'issue de cette quatrième étape



Conf. 2 x 5



Conf. 4 x 3



Conf. 3 x 4

- 2 - Si par contre nous avons hérité d'une configuration  $3 \times 3$ , nous aurons à la sortie une des configurations  $3 \times 4$  ou  $4 \times 3$ .

A l'issue de cette étape, nous ne conservons que deux configurations :  $2 \times 5$  et  $3 \times 4$  en ramenant à ce cas, la configuration  $4 \times 3$  par échange des opérateurs :

$$(i,j) \rightarrow (j,i) \quad \forall i = 1,2,3,4$$

$$\quad \quad \quad \quad \quad \forall j = 1,2,3.$$

XII-11 - ETAPES SUIVANTES

Les améliorations successives se poursuivent suivant le même mécanisme.  
Nous aurons aux stades successifs de calcul :

- $6 \times 2, 5 \times 3, 4 \times 4$
  - $7 \times 2, 6 \times 3, 4 \times 5$
- etc...

XII-12 - CAS GENERAL

Bien que réservée aux fonctions croissantes, cette méthode convient à une fonction quelconque. On peut en effet exprimer toute fonction sous forme croissante en augmentant le nombre de ses variables.

Chaque variable  $x_i$  qui apparaît sous ses deux formes (directe et complé-  
mentée) dans l'expression de la fonction, se dédouble en deux variables :

$$x_i = x_i$$

$$X_i = x_i'$$

$$F(x_1, x_2, \dots, x_n) = F_c(x_1, X_1, x_2, X_2, \dots, x_n, X_n).$$

Exemple :  $F = abc + a'bd + bd'e' + a'e'f + f'g'$  s'écrira sous forme croissante :

$$F_c = abc + Abd + bDE + Aef + FG.$$

XII-13 - RESULTATS

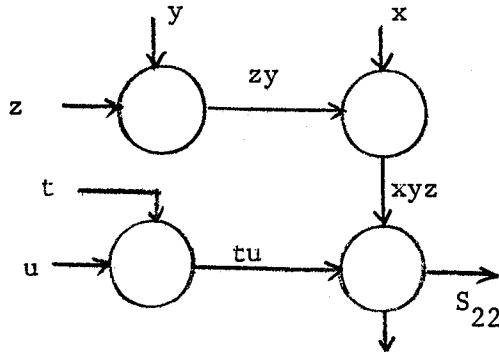
L'algorithme a été programmé en assembleur 360. Le listing en fin de chapitre.

Nous n'avons ici programmé que les quatre premières étapes.

Exemple n° 1 :  $F = F_{inf} = xyztu + ztuv$

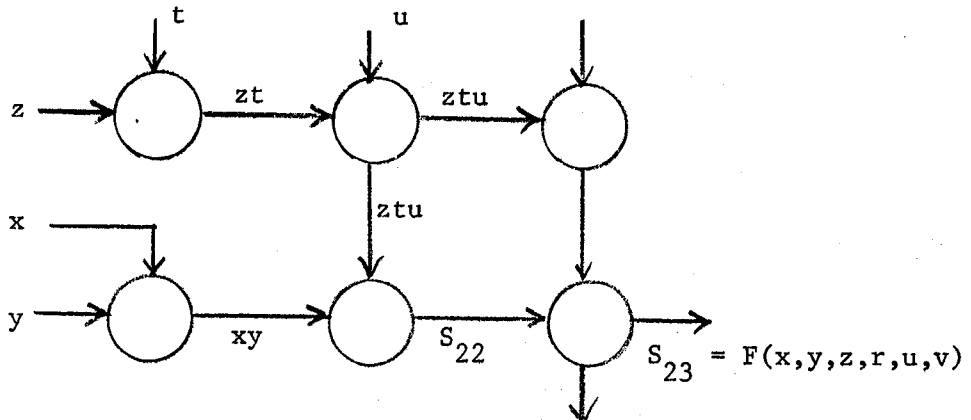
$$F'_{sup} = z't'+u'+x'v'+y'v'$$

Nous trouvons (cf. tableau 1)  $S_{22} = yzxtu$  ce qui correspond au réseau ci-dessous :



Comme sortie  $S_{23}$  nous trouvons  $S_{22} + v \cdot S_{12}$  mais la sortie  $S_{12}$  considérée dans  $S_{23}$  n'est pas celle obtenue à la fin de la première étape où l'on avait  $S_{12} = xyz$ .

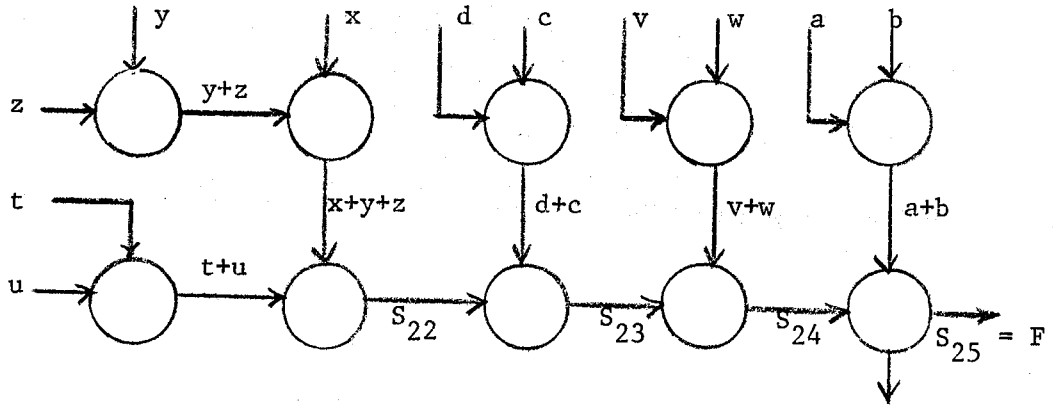
L'algorithme testant le rôle identique des cinq variables dans  $S_{22}$  a transformé le réseau ci-dessus :



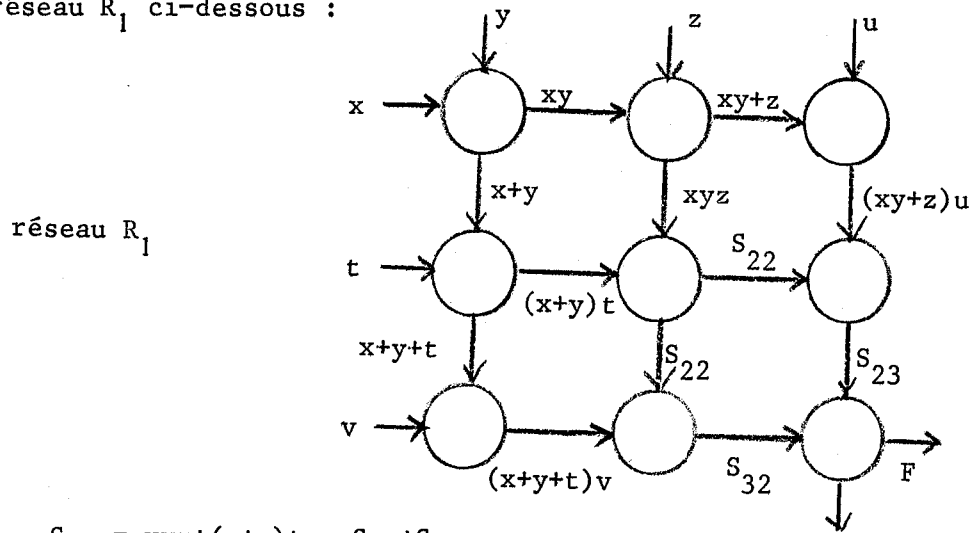
Exemple n° 2  $F = F_{\text{inf}} = x+y+z+t+u+v+w+a+b+c+d$

$$F'_{\text{sup}} = x'y'z't'u'v'w'a'b'c'd'$$

La synthèse donne le résultat ci-dessous (cf. tableau 2)



Exemple n° 3  $F(x,y,z,t,u,v) = xy(z+u)+zu+(x+y)(t+v)+tv$  est synthétisée par le réseau  $R_1$  ci-dessous :



$$S_{22} = xyz+(x+y)t = S_{12}+S_{21}$$

$$S_{23} = S_{22}+(xy+z)u = S_{22}+S_{13}$$

$$S_{33} = S_{23}+(x+y+t)v+S_{22}$$

La première étape du programme donnera

$$S_{22} = v + yt + x(y+t)$$

(cf. tableau 3).

Cette sortie comportant 13 "1" sa distance à la fonction est :

$$d(S_{22}, F) = 4.$$

La sortie  $S_{22}$  fournie par le réseau ci-dessus comporte 12 "1"

xyz	10000000	111111111
(x+y)t	01000010	111111111
xyz+(x+y)t	11000010	111111111

et nous avons  $d(xyz+(x+y)t, f) = 5$  qui forme donc une sortie inférieure à  $S_{22}$  trouvée par le programme.

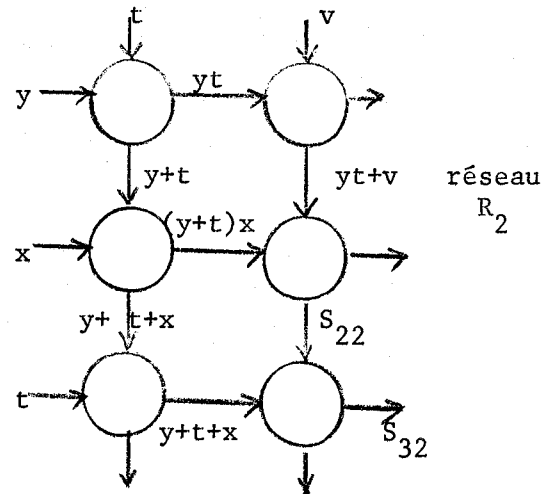
Dans la seconde étape, nous aurons

$$\begin{aligned} S_{23} &= (x+y+t) \cdot S_{22} \\ &= xy + xt + yt + xv + yv + vt \end{aligned}$$

et

$$d(S_{32}, f) = 2$$

(cf. tableau 4).



Alors que sur le réseau  $R_1$  de neuf portes nous avons :

$$d(S_{32}, f) = 3.$$

La sortie élaborée par le programme est donc là encore plus près de  $f$  que celle donnée par le réseau  $R_1$ . Cependant, il est déjà évident que le programme ne trouvera pas une solution comprenant neuf opérateurs :

- il manque encore les monômes  $xyz$   
 $xyu$   
 $zu$
- $S_{32}$  comporte en trop le monôme  $xy$ .

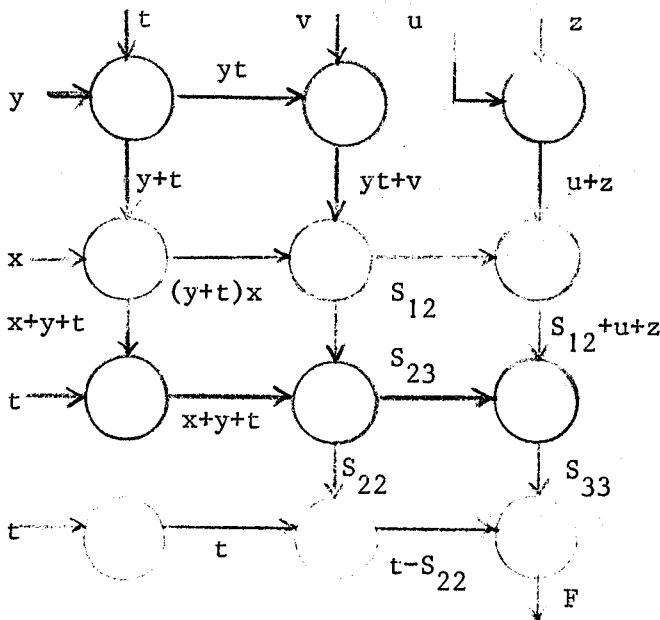
Par l'adjonction d'une troisième colonne au réseau  $R_2$ . On ne peut former que  $(zu)$  ou  $(z+u)$

- $zu$  donnant le monôme manquant  $zu$
- $(z+u)$  donnant les monômes  $xyz$   
 $xyu$  avec l'aide de  $xy \in S_{32}$

mais on ne peut avoir les deux ensemble.

Exemple 4 :  $F(x,y,z,t,u,v) = xy(z+u) + (x+y+v)t + (x+y)v + \emptyset(zu)$ .

Cette fonction est celle précédemment traitée mais dans laquelle le sous-cube  $zu$  est indéterminé. Le programme donne la solution suivante :



(cf. tableau 7)

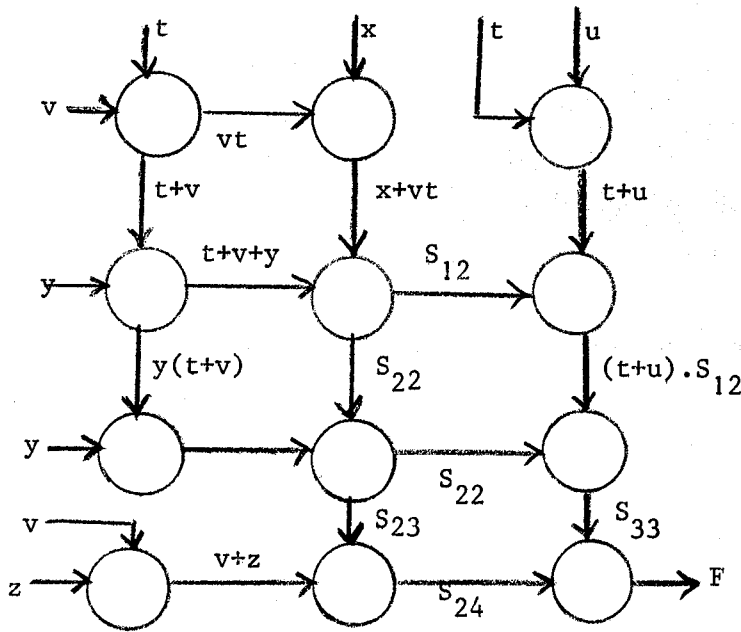
$$\begin{cases} S_{22} = yt+v+(y+t)x \\ S_{23} = S_{22} \cdot (D_{21}+t) \\ S_{33} = (y+z+S_{12}) \cdot S_{23} \\ S_{34} = t \cdot S_{22} + S_{33} \end{cases}$$

La fonction devient ici synthétisable alors qu'elle ne l'était pas dans l'exemple 3.

Exemple 5 :  $F = xy(z+u)+(x+y+t)v+xt+\phi(yt+zu)$ .

C'est la fonction précédente avec en plus indétermination sur les sommets couverts par  $yt$ .

Nous obtenons (cf. tableau 8).

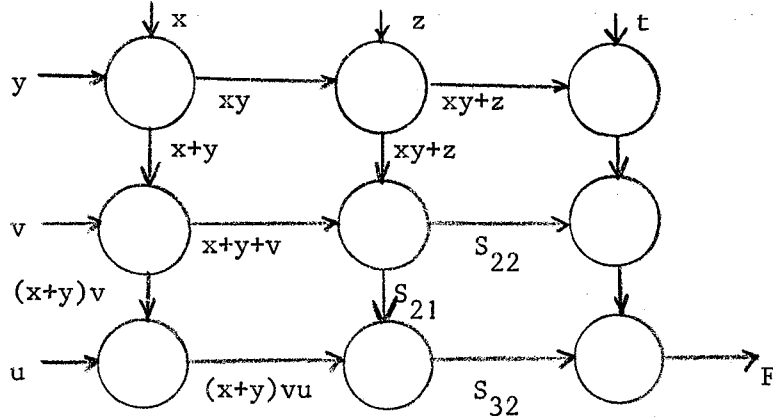


$$\begin{cases} S_{22} = (x+tv)(t+v+y) \\ S_{23} = S_{22} + D_{21} \cdot y \\ S_{24} = (v+z)S_{23} \\ S_{34} = [(t+u) \cdot S_{12} \cdot S_{22}] + S_{24} \end{cases}$$



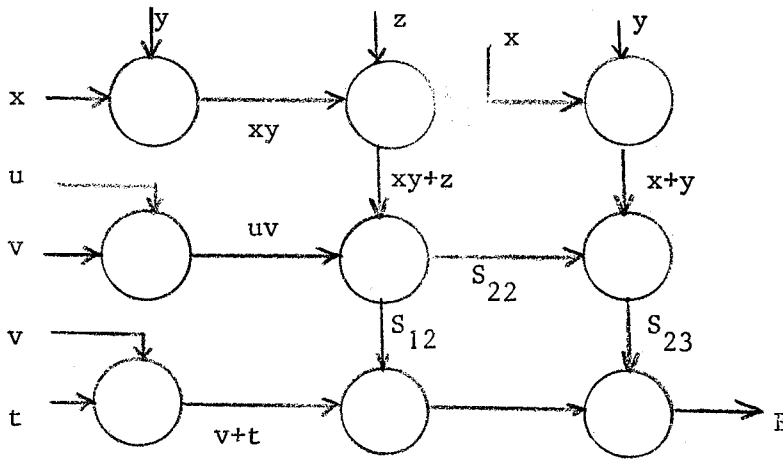
Exemple 6 :  $F(x,y,z,t,u,v) = (x+y)(uv+z)+xy+(v+t)z$

synthétisée à priori par le réseau ci-dessous.



$$\begin{cases} S_{22} = (xy+z)(x+y+v) \\ S_{23} = S_{22} + (xy+z)t \\ S_{33} = D_{21} \cdot u + S_{21} + S_{23} \end{cases}$$

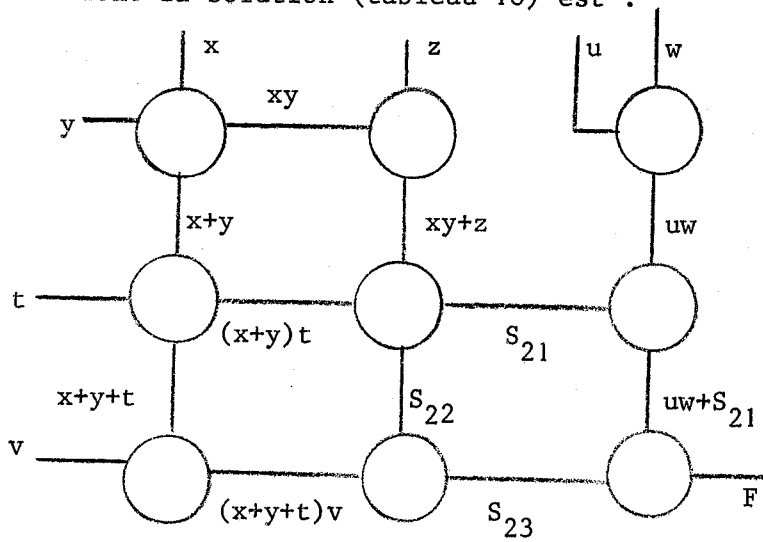
Le programme donne la solution représentée ci-dessous (tableau 9).



$$\begin{cases} S_{22} = xy+z+uv \\ S_{23} = S_{22}(x+y) \\ S_{33} = (v+t) \cdot S_{12} + S_{23} \end{cases}$$

Exemple 7 :  $F = zt(x+y)+xy(t+zuvw)$

dont la solution (tableau 10) est :



$$\begin{cases} S_{22} = (xy+z) \cdot (x+y) \cdot t \\ S_{23} = D_{12} \cdot v + S_{22} \\ S_{33} = (u \cdot w + S_{21}) \cdot S_{23} \end{cases}$$



CHAPITRE 12 : EXEMPLES 1

```

start
EXECUTION BEGINS...
1000010 x
1000001 y
1110000z
1101000 t
1100100 u
0100011 v
1011111 SORTIE S22 : ((y.z).x).(t.u)
0100011 v
1100100 u
1110000 z
1101000 t
1011111 S22
0100111 v.u
0110011 v.z
0101011 v.t
1110100 u.z
1101100 u.t
1111000 z.t
1110100 S11
1111100 S12
1110000 D12
1101100 S21
1100000 D21
1111111 (S12.v)+S22=S23

```

TABLEAU 1 (exemple 1)

```

start
EXECUTION BEGINS...
11110000111010011 x
10101010110100111 y
10000001101101010 z
01000110111111100 t
00100001010011101 u
00011100001111111 v
11111100011101111 SORTIE S22 : ((y.t)+v)+((y+t).x)

```

TABLEAU 3 (exemple 3)

TABLEAU 2

(exemple 2)

```

start
EXECUTION BEGINS...
10000000001 x
01000000001 y
00100000001 z
00010000001 t
00001000001 u
00000100001 v
00000010001 w
00000001001 a
00000000101 b
00000000011 c
00000000001 d
11111000001 SORTIE S22 : ((y+z)+x)+(t+u)
00000000001 d
000000000101 c
0000000001001 b
000000010001 a
000000100001 w
000001000001 v
11111000001 S22
000000000111 d+c
000000001011 d+b
000000010011 d+a
000000100011 d+w
000001000011 d+v
000000001101 c+b
000000010101 c+a
000000100101 c+w
000001000101 c+v
000000011001 b+a
000000101001 b+w
000001001001 b+v
000000110001 a+w
000001010001 a+v
000001100001 w+v
11000000001 S11
11100000001 S12
00000000001 D12
00011000001 S21
00000000001 D21
111110000111 (d+c)+S22=S23
000001000001 v
000000100001 w
00000001001 b
000000010001 a
111110000111 S23
000001100001 v+w
000001001001 v+b
000001010001 v+a
000000101001 w+b
000000110001 w+a
000000011001 b+a
11111100111 (v+w)+S23=S24
000000010001 a
000000001001 b
11111100111 S24
000000011000 a+b
11111111111 (a+b)+S24=S25

```

```

11110000111010011 x
10101010110100111 y
10000001101101010 z
01000110111111100 t
00100001010011101 u
00011100001111111 v
11111100011101111 S22
10100000111110111 x.y
11111010110000011 x+y
10000000111111011 x.z
11110001101000010 x+z
01000000111111111 x.t
11110110111010000 x+t
00100000111011111 x.u
11110001010010001 x+u
00010000111111111 x.v
10000000111101111 y.z
1010101100100010 y+z
00000010111111111 y.t
11101101101001000 y+t
00100000110111111 y.u
10101011010000101 y+u
00001000111111111 y.v
11000111101101000 z+t
00000001111111111 z.u
10100001000001000 z+u
10011101001101010 z+v
01100111010011100 t+u
00000100111111111 t.v
01011110001111100 t+v
00111101000011101 u+v
00000010111111111 S11
11101110110100100 D11
00011110001111111 S12
11110010111010011 D12
11100000111101111 S21
11111101100000000 D21
11111101111101111 (D21+t).S22=S23

```

TABLEAU 4 (exemple 3)

=====

```

00011100001111111 v
00100001010011101 u
10000001101101010 z
01000110111111100 t
11111101111101111 S23
10011101001101010 v+z
00111101000011101 v+u
00000100111111111 v.t
01011110001111100 v+t
00000001111111111 u.z
10100001000001000 u+z
01100111010011100 u+t
11000111101101000 z+t
11111111111101111 (u.z)+S23=S24

```

TABLEAU 5 (exemple 3)

-----

```

00011100001111111 v
00100001010011101 u
10000001101101010 z
01000110111111100 t
11111111111110111 S24
10011101001101010 v+z
00111101000011101 v+u
00000100111111111 v.t
01011110001111100 v+t
00000001111111111 u.z
10100001000001000 u+z
01100111010011100 u+t
11000111101101000 z+t
10111011111111111 (D14+v).S24=S25
00011100001111111 v
00100001010011101 u
10000001101101010 z
01000110111111100 t

```

TABLEAU 6 (exemple 3)

-----

```

start
EXECUTION BEGINS...
1111000111010011 x
1010101110100111 y
1000000101101010 z
0100011111111100 t
0010000010011101 u
0001110001111111 v
1111110011101111 SORTIE S22 : ((y.t)+v)+((y+t).x)
1111000111010011 x
1010101110100111 y
1000000101101010 z
0100011111111100 t
0010000010011101 u
0001110001111111 v
1111110011101111 S22
1010000111110111 x.y
1111101110000011 x+y
1000000111111011 x.z
0100000111111111 x.t
1111011111010000 x+t
0010000111011111 x.u
0001000111111111 x.v
1000000111101111 y.z
0000001111111111 y.t
1110111110100100 y+t
0010000110111111 y.u
0000100111111111 y.v
1100011101101000 z+t
1010000000001000 z+u
1001110001101010 z+v
0110011010011100 t+u
0000010111111111 t.v
0101110011111100 t+v
0011110000011101 u+v
0000001111111111 S11
1110111110100100 D11
0001111001111111 S12
1111001111010011 D12
1110000111110111 S21
1111111111000000 D21
1111111111110111 (D21+t).S22=S23
0001110001111111 v
0010000010011101 u
1000000101101010 z
0100011111111100 t
1111111111110111 S23
1001110001101010 v+z
0011110000011101 v+u
0000010111111111 v.t
0101110011111100 v+t
101000000001000 u+z
0110010100111100 u+t
110001101101000 z+t
1001111111111111 ((u+z)+S12).S23=S33
0100011111111100 t
1011111111111111 S33
1111111111111111 ((t ).S22)+S33=S34
    
```

TABLEAU 7 (exemple 4)

```

start
EXECUTION BEGINS...
111100111010011 x
1010101110100111 y
100000101101010 z
0100011111111100 t
001000010011101 u
0001110011111111 v
1111011111101111 SORTIE S22 : ((t.v)+x).
0001110011111111 v ((t+v)+y)
1010101110100111 y
10000101101010 z
0100011111111100 t
001000010011101 u
1111011111101111 S22
0000101111111111 v.y
1011110001001111 v+y
100111001101010 v+z
0000011111111111 v.t
0101110011111100 v+t.
001111000011101 v+u
1000001111011111 y.z#
111011110100100 y+t
0010001101111111 y.u
110001101101000 z+t
101000000001000 z+u
0110010100111100 t+u
0000011111111111 S11
0101110011111100 D11
1111011110100111 S12
0000001111111111 D12
111111000100100 S21
0000101111111111 D21
1111111111101111 (D21.y)+S22=S23
0001110011111111 v
001000010011101 u
100000101101010 z
0100011111111100 t
1111111111101111 S23
100111001101010 v+z
001111000011101 v+u
0000011111111111 v.t
0101110011111100 v+t
101000000001000 u+z
0110010100111100 u+t
110001101101000 z+t
1001111111111111 (v+z).S23=S24
0100011111111100 t
001000010011101 u
1001111111111111 S24
0110010100111100 t+u
1111111111111111 (((t+u).S12).S22)+S24=
=S34
    
```

TABLEAU 8 (exemple 5)

```

11101111111111111111 SORTIE S22 : ((x.y)+z).((x+y).t)
11011111000000 x
1011010100100 y
0111001010100 z
0001000000010 w
0001000001000 u
0001000000001 v
1110111111111111 S22
1001111100100 x.y
1111010000000 x+y
0101111010100 x.z
1111001000000 x+z
0001111000010 x.w
0001111001000 x.u
0001111000001 x.v
0011011110100 y.z
1111000000100 y+z
0001010100110 y.w
0001010101100 y.u
0001010100101 y.v
0001001010110 z.w
0001001011100 z.u
0001001010101 z.v
0001000001010 w.u
0001000000011 w.v
0001000001001 u.v
1111001000100 S12
1001111100100 D11
1001111100100 S11
000111110100 D12
111111110101 (D12.v)+S22=S23
0001000001000 u
0001000000010 w
111111110101 S23
0001000001010 u.w
1111111111111 ((u.w)+S21).S23=S33

```

TABLEAU 10 (exemple 7)

```

11111110011111111111 SORTIE S22 : ((x.y)+z)+(u.v)
10010101111100 x
1010001110011 y
0110110101111 z
0100000010000 t
0001101010110 S22 v 1111111001111 S22
1000000111111 x.y
1011011110000 x+y
0000010111111 x.z
1111110101100 x+z
1101010010000 x+t
0001000111110 x.v
1001111010100 x+v
0010000111111 y.z
1110111100011 y+z
1110001010000 y+t
0000001110111 y.v
1011101010010 y+v
0100000111111 z.t
0000100111111 z.v
0101101010000 t+v
1110110101111 S12
0001001011111 S21
1000000111111 S11
0000000111111 D12
1011011111111 (x+y).S22=S23
0001101010110 v
0100000010000 t
0110110101111 z
1011011111111 S23
0101101010000 v+t
0000100111111 v.z
0111111000110 v+z
0100000111111 t.z
1111111111111 ((v+t).S12)+S23=S33

```

TABLEAU 9 (exemple 6)

CAS GENERAL : SYNTHESE DE FONCTIONS COMPLETES  
 SUR RESEAUX IRREDONDANTS  
 PROBLEME DE COUVERTURE

```

JAM73: PROC OPTIONS(MAIN) ;
/* PROGRAMME DE COLORIAGE D'UN GRAPHE DE N SOMMETS
   NISMAX : BORNE SUPERIEURE DU NOMBRE D'ENSEMBLES
           INTERIEUREMENT STABLES MAXIMAUX.
   IX : BORNE SUPERIEURE DU NOMBRE DE COUVERTURES*/
DCL(N,IX,NISMAX) FIXED BIN,
NSOM FIXED BIN EXTERNAL;
GET LIST(N,NISMAX,IX);
BEGIN ;
  DCL SOM( NISMAX) BIT(N);
  CALL JAM71(SOM,N,NISMAX);
  PUT SKIP DATA(NSOM);
  CALL JAM72(SOM,N,NSOM,IX,NISMAX);
  END ; END JAM73 ;
* PROCESS('COMP,SIZE=95232,ST,ATR,XREF');
(SUBRG):JAM71:PROC(SOM,N,NISMAX);
/* RECHERCHE DES ENSEMBLES INTERIEUREMENT STABLES
   MAXIMAUX D'UN GRAPHE
   LES DONNEES SONT RENTREES DANS L'ORDRE SUIVANT
   -LE NOMBRE DE SOMMETS N DU GRAPHE
   -LES COUPLES X,Y AVEC X<Y SI X ET Y SONT DEUX
     SOMMETS ADJACENTS
   LES RESULTATS SONT DONNES SOUS FORME DE CHAINES DE BITS
   -SI LE J-EME BIT DE LA I-EME CHAINE EST EGAL A 1
     LE SOMMET J APPARTIENT AU I-EME ENSEMBLE INTER. ST.
   MAXIMAL */
DCL NSOM FIXED BIN EXTERNAL,
(VAR(N),(SOM,CLASS2,CLASS3,CLASS3B,CLASS4)(NISMAX),USOM,U4,UA,UM,
UASOM,UMSOM) BIT(N),
(DISTAN(N,N),DEG(N),X,Y,N,NISMAX) FIXED BIN;
VAR='0'B;
DISTAN=0;ON ENDFILE (SYSIN) GOTO SUITE;
LIRE:GET LIST(X,Y);DISTAN(X,Y),DISTAN(Y,X)=1;GOTO LIRE;
SUITE:DO I=1 TO N; DEG(I)=SUM(DISTAN(*,I));END;
PUT EDIT('MATRICE DE CONNEXION DU GRAPHE DE ',N,' SOMMETS')
(A(35),F(3),A);PUT SKIP(4);
DO I=1B TO N;DO J=1B TO N;PUT EDIT(DISTAN(I,J)) (F(1));
END;PUT SKIP;END;
DO I=1 TO N;DO J=I+1 TO N;IF DISTAN(I,J)=1 THEN DO;
IF DEG(I)>=DEG(J) THEN SUBSTR(VAR(I),J,1)='1'B;
ELSE SUBSTR(VAR(J),I,1)='1'B; END;END;END;
J=0B;RE:J=J+1B;IF J>N THEN DO;
PUT SKIP LIST('GRAPHE SANS ARETE.SOLUTION EVIDENTE');STOP;END;
IF VAR(J)='0'B THEN GOTO RE;
NSOM=2;SOM(1)='0'B;SUBSTR(SOM(1),J,1)='1'B;
SOM(2)=VAR(J);
PUT EDIT('MONOMES ASSOCIES AUX ',N,' SOMMETS')
(PAGE,A(21),F(3),A);PUT SKIP(4);
DO I=1B TO N;PUT LIST(VAR(I),DEG(I)) SKIP;END;
DO J=J+1B TO N;UM=VAR(J);IF UM='0'B THEN GOTO FINJ;

```



```

IF USOM=(UASOM|UM) THEN DO;NS=NS+1;SOM(NS)=USOM;GOTO FINI;END;
IF USOM=UASOM THEN DO;N2=N2+1B;NS=NS+1B;SOM(NS),CLASS2(N2)=USOM;GOTO
FINI;END;
IF USOM=UMSOM THEN DO;N3=N3+1B;NS=NS+1B;SOM(NS),CLASS3(N3)=USOM;GOTO
FINI;END;
N4=N4+1;CLASS4(N4)=USOM; FINI;END;
DO I=1B TO N4;U4=(CLASS4(I)|UA);
DO K=1B TO N2;IF U4=(U4|CLASS2(K)) THEN GOTO FINI42;END;
NS=NS+1B;SOM(NS)=U4;
FINI42:END;
N3B=0;
DO I=1B TO N4;U4=(CLASS4(I)|UM);
DO K=1B TO N3;IF U4=(U4|CLASS3(K)) THEN GO TO FINI43;END;
DO K=1B TO N3B;UMSOM=CLASS3B(K);UASOM=(UMSOM|U4);IF U4=UASOM
THEN GO TO FINI43;
IF UMSOM=UASOM THEN CLASS3B(K)='0'B;END;
N3BB=0;
DO K=1B TO N3B;IF CLASS3B(K)='0'B THEN DO;
N3BB=N3BB+1;CLASS3B(N3BB)=CLASS3B(K);END;END;
N3B=N3BB+1;CLASS3B(N3B)=U4;
FINI43:END;
DO I=1B TO N3B;SOM(NS+1)=CLASS3B(I);END;
NSOM=NS+N3B;
FINI:END;
PUT EDIT((55)'*') (PAGE,A);
PUT EDIT('* LES ',NSOM,
' ENSEMBLES INTERIEUREMENT STABLES MAXIMAUX *')
(SKIP,A(7),F(3),A);
PUT EDIT((55)'*') (SKIP,A);PUT SKIP(5);
DO I=1 TO NSOM;SOM(I)='SOM(I)';PUT SKIP LIST(SOM(I));END;
END JAM71;
* PROCESS('COMP,SIZE=95232,ST,ATR,XREF');
(SUBRG):JAM72: PROC(SOM,N,NSOM,IX,NISMAX) ;
/* RECHERCHE DE LA COUVERTURE MINIMALE D'UN GRAPHE DONT
ON CONNAIT LES ENSEMBLES INTERIEUREMENT STABLES MAXIMAUX
PAR LE TABLEAU SOM(NSOM) */
DCL((G,TZERO)(10*IX),T(NSOM,IX),V,COUV,M,U,UI) BIT(NSOM),
(NT,MDUAL)(NSOM),SOM(NISMAX) BIT(N),(N,IX,NSOM,NISMAX)
FIXED BIN;
ON SUBRG GO TO IMP;
G='0'B ;N1=0B;PUT PAGE;
DO I=1B TO NSOM ;
IF SUBSTR(SOM(I),1,1)='1'B THEN DO ;
N1=N1+1B ; SUBSTR(G(N1),1,1)='1'B ; END ; END ;
DO J=10B TO N ; N2=0B ; M='0'B ;PUT SKIP DATA(J);
DO I=1B TO NSOM ;
IF SUBSTR(SOM(I),J,1)='1'B THEN DO ;
N2=N2+1B ; SUBSTR(M,I,1)='1'B ; MDUAL(N2)=I ; END ; END ;
NS,NZERO=0B ; NT=0B ;
DO I=1B TO N1 ; U=G(I) ; U=(UI & M) ;
IDEX1=INDEX(U,'1'B) ; IF IDEX1=0B THEN DO ;
NZERO=NZERO+1B ; TZERO(NZERO)=UI ; GOTO FINI ; END ;
NS=NS+1B ; G(NS)=UI ;
SUBSTR(U,IDEX1,1)='0'B;IDEX2=INDEX(U,'1'B);
IF IDEX2=0B THEN GOTO FINI ;
NT(IDEX1)=NT(IDEX1)+1B;T(IDEX1,NT(IDEX1))=UI;

```

```

FINI: END ;
DO I=1B TO N2 ; DO K=1B TO NZERO ;
U=TZERO(K) ; NDUAL=MDUAL(I) ; SUBSTR(U,NDUAL,1)='1'B ;
DO L=1B TO NT(NDUAL);
IF U=(U|T(NDUAL,L)) THEN GOTO FINK ; END ;
NS=NS+1B ; G(NS)=U ; FINK: END ; END ;
N1=NS ; END ;
PUT PAGE DATA(N1,NS,NC,I,J,K,L,NZERO,N2, IDEX1, IDEX2);
NC=10000 ;
DO I=1B TO N1 ; NTEST=TEST1(G(I)) ;
IF NTEST < NC THEN DO ; NC=NTEST ; COUV=G(I) ; END ;
END ;
PUT EDIT('LA COUVERTURE MINIMALE EST COMPOSEE DE ',NC,' ENSEMBLES')
(PAGE,A(39),F(3),A);PUT EDIT((52)'*') (SKIP,A);PUT SKIP(3) LIST(COUV);
PUT SKIP(3) DATA(COUV) ;
DCL TEST1 ENTRY RETURNS(FIXED BIN);
TEST1:PROCEDURE(UU) FIXED BIN;
DCL UU BIT(NSOM);
NU=0B;
DO IU=1B TO NSOM;IF SUBSTR(UU,IU,1)='1'B THEN
NU=NU+1B;END;RETURN(NU);END TEST1;
IMP:PUT PAGE DATA(N1,NS,NC,I,J,K,L,NZERO,N2, IDEX1, IDEX2);
PUT SKIP LIST(NT);
END JAM72 ;
EOF:

```



## SYNTHESE DE FONCTIONS INCOMPLETES SUR RESEAUX IRREDONDANTS

```

(SUBRG):
JAM91005 : PROCEDURE OPTIONS(MAIN);
GET EDIT(N,M,MS) (F(2),X(2),F(2),X(2),F(2));
GET SKIP;
BEGIN;
DCL(MONOMS(MS),MONOMX) CHARACTER(2*N);
DCL(MONOM(2**N),RESTE(2**N),REST(2**N)) CHAR(2*N);
DCL (VAR(2*N),PRIM(2*N)) CHARACTER(2);
DCL (PREZ(2*N),VOI(N,N)) BIT(N*M);
DCL UN BIT(M) VARYING;
DO I=1 TO M;
GET LIST(MONOM(I));
END;
GET LIST(VAR,MONOMS);
PREZ='0'B;
DO I=1 BY 2 TO (2*N);
DO J=1 TO M;
IF SUBSTR(MONOM(J),I,2)=VAR(I) THEN UN='1'B;
ELSE UN='0'B;

SUBSTR(PREZ(I),J,1)=UN;
SUBSTR(PREZ(I+1),J,1)=(~UN);
END;
END;
PUT EDIT('LES ',N,' VARIABLES DE LA FONCTION')(X(20),A(4),F(2),A);
PUT SKIP(3);
PUT LIST(VAR);
PUT EDIT('LES ',M,' MONOMES DE LA BORNE INFERIEURE')
(SKIP(10),X(20),A(4),F(2),A);

PUT SKIP(3);
DO I=1 TO M;
PUT LIST(MONOM(I));
END;
PUT EDIT('LES ',MS,' MONOMES DU COMPLEMENT DE LA BORNE SUPERIEURE')
(SKIP(10),X(10),A(4),F(2),A);

PUT SKIP(3);
PUT LIST(MONOMS);
PUT PAGE;
DO I=2 BY 2 TO (2*N);
PRIM(I)=VAR(I-1); PRIM(I-1)=VAR(I);
END;
UN='1'B;
DO I=2 TO M;
UN=UN||'1'B;
END;
VOI(1,1)=UN;
DCL POINTE CHARACTER(4*(N-1)) VARYING;
DCL INTRO(2:N);

```

```

/* */
/* */
/* */
/* */

```

INCV01 :

```

    PROCEDURE(IV,JV,RV);
    DCL RV LABEL;
    PUT EDIT('POINTE : ',POINTE) (SKIP,A(9),A);
    INTROU=INTROD;
    DO KV=1 TO (IV-1);
    IF VOI(IV-1,KV)='0'B THEN GO TO FINKV;
    IF MOD(KV,2)=0 THEN J=-1;
    ELSE J=1;
    IF MOD(JV,2)=0 THEN I=-1;
    ELSE I=1;
    MV=JV+I*(J+1)/2;
    LV=JV+I*(1-J)/2;
    CALL VOIE(KV,LV);
    CALL VOIE(KV+1,MV);
    FINKV : END;

```

/\* \*/  
/\* \*/  
/\* \*/

```

VOIE :
    PROCEDURE(KW,LW);
    DCL UW BIT(M*N);
    UW='0'B;
    DCL RESTES CHAR(M*N) VARYING;
    DCL (CORPS,CORP) CHARACTER(M*N) VARYING;
    PUT SKIP(2);
    DO I=1 TO IV;
    DO J=1 TO IV;
    PUT EDIT (VOI(I,J)) (X(2),B(INTROD));
    END;
    PUT SKIP;
    END;
    PUT SKIP(2);
    UW=BOOL(VOI(IV-1,KV),PREZ(LW),'0001'B);
    PUT EDIT('UW : ',UW) (SKIP,A(5),B);
    IF UW='0'B THEN GO TO VOIEND;
    IF VOI(IV,KW)='0'B THEN VOI(IV,KW)=UW;
    ELSE
    DO;
    CALL INREST(RESTE,VOI(IV,KW),CORPS);
    CALL INREST(REST,UW,CORP);
    PUT EDIT('RESTE') (SKIP,A);
    DO I=1 TO INTROD;
    PUT LIST(RESTE(I));
    END;
    PUT EDIT('REST') (SKIP,A);
    DO I=1 TO INTROD;
    PUT LIST(REST(I));
    END;
    PUT EDIT('CORPS : ',CORPS) (SKIP,A(8),A);
    PUT EDIT('CORP : ',CORP) (SKIP,A(7),A);
    RESTES='';
    DO I=1 TO INTROD;
    IF RESTE(I)='0' THEN
    DO;
    IF INDEX(RESTES,RESTE(I))=0 THEN GO TO FINI;
    RESTES=RESTES||RESTE(I);

```

```

NMS=0;
DO J=1 TO INTROD;
IF RESTE(I)=REST(J) THEN
DO;
REST(J)='0' ;
SUBSTR(VOL(IV,KW),J,1)='1'B;
NMS=NMS+1;
END;
END;
IF NMS>0 THEN GO TO FINI;
DO K=1 BY 2*(IV-1) TO LENGTH(CORP);
MONOMX=RESTE(I);
NJK=0;
DO J=1 BY 2 TO (2*N);
IF SUBSTR(MONOMX ,J,2)='00' THEN
DO;
SUBSTR(MONOMX,J,2)=SUBSTR(CORP,K+NJK,2);
NJK=NJK+2;
END;
END;
DO J=1 TO MS;
IF MONOMX=MONOMS(J) THEN
DO;
INTRO(IV)=0;
GO TO RV;
END;
END;
INTRO(IV)=INTRO(IV)+1;
INTROU=INTROU+1;
MONOM(INTROU)=MONOMX;
SUBSTR(VOL(IV,KW),INTROU,1)='1'B;
DO J=1 BY 2 TO 2*N;
IF SUBSTR(MONOM(INTROU),J,2)=VAR(J) THEN UN='1'B;
ELSE UN='0'B;
SUBSTR(PREZ(J),INTROU,1)=UN;
SUBSTR(PREZ(J+1),INTROU,1)=(UN);
END;
PUT EDIT('INTROU = ',INTROU,MONOM(INTROU))
(SKIP,A(9),F(2),X(5),A);
PUT EDIT('PREZ') (SKIP(2),A);
PUT SKIP;
PUT LIST(PREZ);
PUT SKIP(2);
END;
END;
FINI : END;
RESTES='';
DO I=1 TO INTROD;
IF REST(I)='0' THEN
DO;
IF INDEX(RESTES,REST(I))=0 THEN
DO;
SUBSTR(VOL(IV,KW),I,1)='1'B;
GO TO FINII;
END;
END;
RESTES=RESTES||REST(I);

```

```

DO K=1 BY 2*(IV-1) TO LENGTH(CORPS);
MONOMX=REST(1);
NJK=0;
DO J=1 BY 2 TO (2*N);
IF SUBSTR(REST(1),J,2)='00' THEN
DO;
SUBSTR(MONOMX,J,2)=SUBSTR(CORPS,K+NJK,2);
NJK=NJK+2;
END;
END;
DO J=1 TO MS;
IF MONOMX=MONOMS(J) THEN
DO;
INTRO(IV)=0;
GO TO RV;
END;
END;
SUBSTR(VOI(IV,KW),1,1)='1'B;
INTRO(IV)=INTRO(IV)+1;
INTROU=INTROU+1;
MONOM(INTROU)=MONOMX;
SUBSTR(VOI(IV,KW),INTROU,1)='1'B;
DO J=1 BY 2 TO 2*N;
IF SUBSTR(MONOM(INTROU),J,2)=VAR(J) THEN UN='1'B;
ELSE UN='0'B;
SUBSTR(PREZ(J),INTROU,1)=UN;
SUBSTR(PREZ(J+1),INTROU,1)=(^UN);
END;
PUT EDIT('INTROU = ',INTROU,MONOM(INTROU))
(SKIP,A(9),F(2),X(5),A);
PUT EDIT('PREZ') (SKIP(2),A);
PUT SKIP;
PUT LIST(PREZ);
PUT SKIP(2);
END;
END;
FINII : END;
END;

```

```

/* */
/* */
/* */
/* */

```

```

INREST :
PROCEDURE(AR, BR, CR);
DCL AR(M*N) CHAR(2*N);
DCL BR BIT(M*N);
DCL CR CHAR(M*N) VARYING;
DCL DR CHAR(2*N) VARYING;
CR='';
DO I=1 TO INTROU;
DR='';
IF SUBSTR(BR,I,1)='0'B THEN AR(I)='0' ;
ELSE
DO;
AR(I)=MONOM(I);
DO J=1 BY 2 TO (2*N);

```

```

IF INDEX(POINTE,VAR(J))=0 THEN
DO;
DR=DR||SUBSTR(AR(1),J,2);
SUBSTR(AR(1),J,2)='00';
END;
END;
END;
IF INDEX(CR,DR)=0 THEN CR=CR||DR;
END;
END INREST;
VOIEND : END;
INTROD=INTROU;
PUT EDIT('INTROD : ',INTROD) (SKIP,A(9),F(2));
END INCVOI;
DCL INCVOI ENTRY(FIXED BINARY, , );

```

```

/* */
/* */
/* */
/* */

```

```

VERIFI : PROCEDURE(IVE,JVE,KVE,RVE,SVE);
DCL (RVE,SVE) LABEL ;
POINTE=SUBSTR(POINTE,1,4*(IVE-2));
JVE=JVE+KVE;
IF JVE>(2*N) THEN GO TO SVE;
IF INDEX(POINTE,VAR(JVE))=0 THEN GO TO RVE;
POINTE=POINTE||VAR(JVE)||PRIM(JVE);
DO J=1 TO IVE;
VOI(IVE ,J)='0'B;
END;
INTROD=INTROD-INTRO(IVE);
INTRO(IVE)=0;
END VERIFI;
DCL VERIFI ENTRY(FIXED BINARY,,FIXED BINARY,,);

```

```

/* */
/* */
/* */
/* */

```

```

INTROD=1;
DO I=2 TO N;
INTRO(I)=0;
END;
I1=-1;
IUN :
CALL VERIFI(2,I1,2,IUN,IMPOSS);
CALL INCVOI(2,I1,IUN);
I2=0;
IDEU :
CALL VERIFI(3,I2,1,IDEU,IUN);
CALL INCVOI(3,I2,IDEU);
I3=0;
ITROI :
CALL VERIFI(4,I3,1,ITROI,IDEU);
CALL INCVOI(4,I3,ITROI);
I4=0;
IKATR :
CALL VERIFI(5,I4,1,IKATR,ITROI);

```



```
CALL INCVOI(5,14,IKATR);
I5=0;
ICINQ :
CALL VERIFI(6,15,1,ICINQ,IKATR);
CALL INCVOI(6,15,ICINQ);
PUT PAGE;
PUT EDIT('SOLUTION : ',POINTE) (PAGE,A(11),A);
PUT EDIT('LES ',INTROD,' MONOMES DE LA BORNE INFERIEURE')
      (SKIP(10),X(20),A(4),F(2),A);
PUT SKIP(5);
DO I=1 TO INTROD;
PUT EDIT(MONOM(I)) (SKIP,X(5),A);
END;
GO TO FINI;
IMPOSS :
PUT SKIP;
PUT EDIT('AUCUNE SOLUTION')(A);
FINI : END;
      END
END JAM91005 ;
EOF:
```

## SYNTHESES DE FONCTIONS COMPLETES SUR RESEAUX REDONDANTS AVEC S=1

```

(SUBRG):
JAM96 : PROCEDURE OPTIONS (MAIN) ;
DCL SYSPRINT FILE STREAM PRINT ENVIRONMENT (F(120)),
SYSIN FILE STREAM ENVIRONMENT (F(80)),
D96 FILE STREAM EXTERNAL ENVIRONMENT (F(80)),
(N,M) FIXED DEC;
GET FILE (D96) LIST (N,M);
BEGIN ;
DCL MONOM(M) CHARACTER(2*N),
L(3) LABEL,
(IX(2*(N+1)),IV1,KA(N)) FIXED BIN,
((VAR,PRIM)(2*N),C2,CC2) CHAR(2),
LVAR CHAR(4*N),
(VOI(2*(N+1),2*(N+1)),PREZ(2*N),APREZ,AVOI,AUV,AB,BA,(A,B)(N),UV)
BIT(M),
(UN BIT(M) ,(POINTE,VARED,VAREDI) CHAR(4*N),
QUEUES CHARACTER(2*M*N) ,
QUEUE CHAR(2*N)) VARYING;
GET FILE (D96) LIST (MONOM,VAR,LVAR);
PUT EDIT('LES ',N,' VARIABLES DE LA FONCTION'
(X(20),A(4),F(2),A(25)));
PUT LIST(VAR) SKIP(3);
PUT EDIT('LES ',M,' MONOMES DE LA FONCTION'
(SKIP(10),X(20),A(4),F(2),A(23)));
PUT SKIP(3);
DO I=1 TO M;
PUT EDIT(MONOM(I)) (SKIP,A);
END;
DO I=2 BY 2 TO (2*N);
PRIM(I)=VAR(I-1); PRIM(I-1)=VAR(I);
END;
UN='1'B;DO I=2 TO M;UN=UN||'1'B;END;
VOI(1,1)=UN;
DO I=1 BY 2 TO (2*N);
DO J=1 TO M;
IF SUBSTR(MONOM(J),I,2)=VAR(I) THEN
DO;
SUBSTR(PREZ(I),J,1)='1'B;
SUBSTR(PREZ(I+1),J,1)='0'B;
END;
ELSE
DO ;
SUBSTR(PREZ(I),J,1)='0'B;
SUBSTR(PREZ(I+1),J,1)='1'B;
END;
END;
END;
END;
IX=0;
/*
/*
INCVOI :

```

UNIVERSITE DE GRENOBLE  
 BIBLIOTHEQUE CENTRALE  
 30000 ST MARTIN D'HERPES  
 TEL 04 77 15 71 11  
 FAX 04 77 15 71 12  
 www.univ-grenoble.fr  
 CEDEX 40 53  
 38000 - GRENOBLE - GARE

```

PROCEDURE(IV,JV);
DISPLAY('INCVOI');
NL=0; IV1=IV; JV1=JV;
LPOINTE=LENGTH(POINTE)/4 -1; IRED=0;
IDEX=INDEX(POINTE,VAR(JV)); IF IDEX $\neq$ 0 THEN IF IDEX $\leq$ 4*LPOINTE
THEN IRED=1;
IF MOD(JV,2)=0 THEN IJV=-1; ELSE IJV=1;
DO KV=1 TO IV-1;
IF VOI(IV-1,KV)='0'B THEN GO TO FINKV;
IVOI =TEST1(VOI(IV-1,KV));
IF IRED=0 THEN IF IVOI=2**(N-LPOINTE) THEN GO TO FINKV;
IF MOD(KV,2)=0 THEN IKV=0; ELSE IKV=1;
CALL VOIE(KV,JV+IJV*(1-IKV));
CALL VOIE(KV+1,JV+IKV*IJV);
FINKV : END;
/*
/*
DCL TEST1 ENTRY RETURNS(FIXED BINARY);
TEST1 : PROCEDURE(UT) FIXED BINARY;
DCL UT BIT(M);
LT=0;
DO I=1 TO M;
IF SUBSTR(UT,I,1)='1'B THEN LT=LT+1;
END;
RETURN(LT);
END TEST1;
/*
/*
VOIE :
PROCEDURE(KW,LW);
DISPLAY('VOIE');
PUT SKIP DATA(IV1,KV,IJV,IKV,JV1);
PUT SKIP(5); DO J=1 TO IV1; DO I=J TO IV1;
PUT EDIT(VOI(I,J)) (X(2),B(1)); END; PUT SKIP; END;
UW=BOOL(VOI(IV-1,KV),PREZ(LW),'0001'B);
IF UW='0'B THEN GO TO VOIEND;
IF VOI(IV,KW)='0'B THEN VOI(IV,KW)=UW;
ELSE DO;
IF TEST1(UW) $\neq$ TEST1(VOI(IV,KW)) THEN DO;
DISPLAY('TEST1');
UW:NL=NL+1; A(NL)=VOI(IV,KW); B(NL)=UW; KA(NL)=KW;
GO TO VOIEND; END;
QUEUES='';
DO MW=1 TO M;
IF SUBSTR(VOI(IV,KW),MW,1)='0'B THEN GO TO FINMW;
DO NW=1 BY 2 TO (2*M);
IF INDEX(POINTE,VAR(NW)) =0 THEN
QUEUES=QUEUES||SUBSTR(MONOM(MW),NW,2);
END;
FINMW : END;
DO MW=1 TO M;
IF SUBSTR(UW,MW,1)='0'B THEN GO TO FIMW;
QUEUE='';
DO NW=1 BY 2 TO (2*M);
IF INDEX(POINTE,VAR(NW)) =0 THEN
QUEUE=QUEUE||SUBSTR(MONOM(MW),NW,2);

```

```

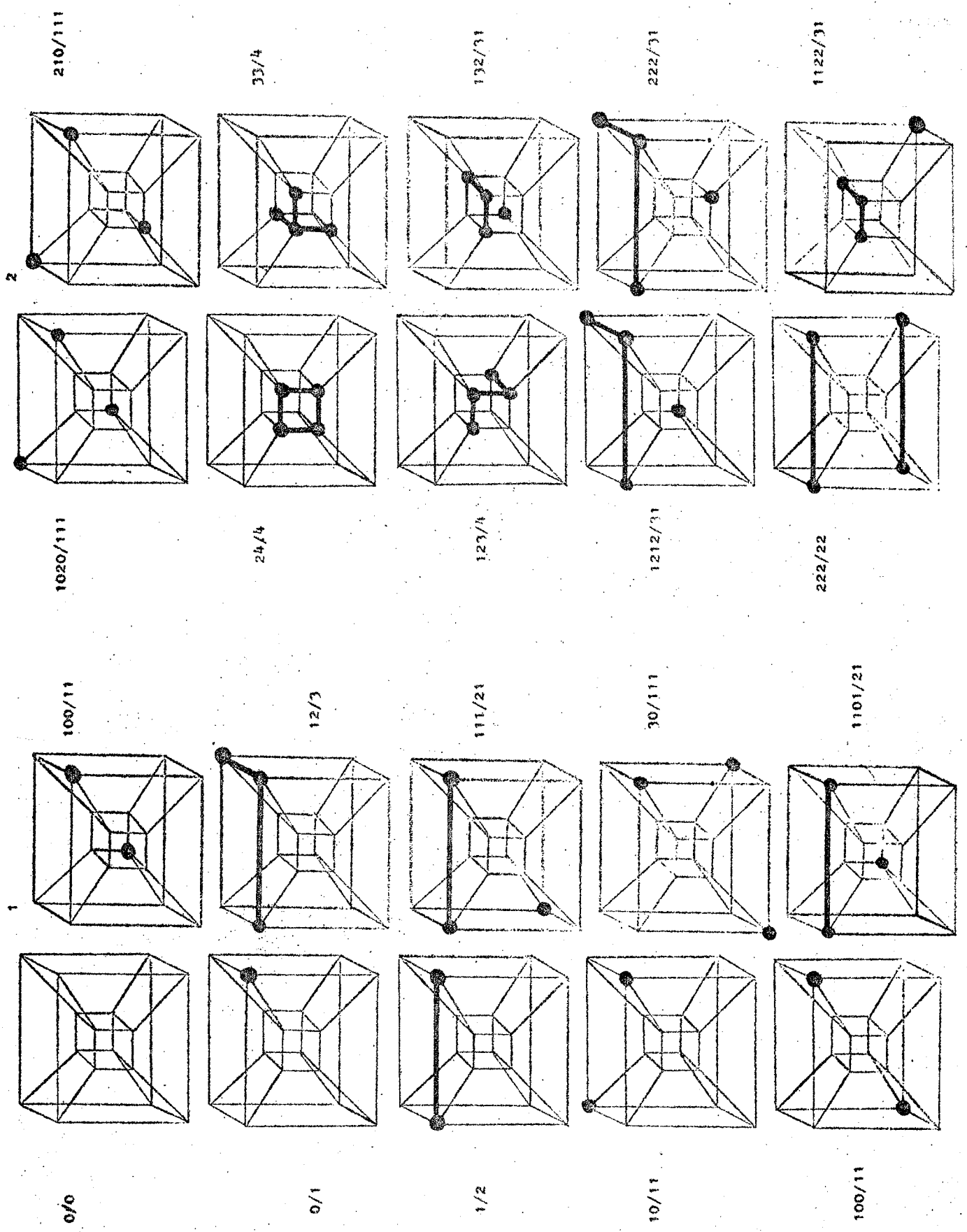
END;
IF INDEX(QUEUES,QUEUE)=0 THEN DO;DISPLAY('QUEUES'); GO TO UNM;END;
FIMW : END;
END;
VOIEND :
PUT SKIP(5);DO J=1 TO IV1;DO I=J TO IV1;
PUT EDIT(VOI(I,J)) (X(2),B(M));END;PUT SKIP;END;
END VOIE;
/*
/*
END INCVOI;
DCL INCVOI ENTRY(FIXED BINARY, );
/*
/*
PUT PAGE;
IX(2)=-1;
ADR2 : POINTE='';
IX(2)=IX(2)+2;
IF IX(2)>2*N THEN GO TO IMPOSS;
POINTE=VAR(IX(2))||PRIM(IX(2));
PUT SKIP DATA(POINTE);
VOI(2,1),VOI(2,2)='0'B;
CALL INCVOI(2,IX(2));
CALL COUCHE(3,L(1));
COUCHE : PROCEDURE(I,L1) RECURSIVE;
DCL LI LABEL;
L(1)=DEB;L(2)=ADR;L(3)=ETIQ;
DISPLAY('COUCHE');
GO TO LI;
DEB : IX(1)=0;
ADR : IX(1)=IX(1)+1;
IF IX(1)>2*N THEN IF I=3 THEN GO TO ADR2;
ELSE CALL COUCHE(I-1,L(2));
IF INDEX(POINTE,VAR(IX(1)))=0 THEN GO TO ADR;
POINTE=POINTE||VAR(IX(1))||PRIM(IX(1));
PUT SKIP DATA(POINTE);
ETIQ : DO J=1 TO I;VOI(I,J)='0'B;END;
CALL INCVOI(I,IX(1));
PUT SKIP DATA(NL,IV1,IX(IV1));
IF NL>0 THEN IF IRED=1 THEN GO TO INDIK1;ELSE GO TO INDIK2;
IF LENGTH(POINTE)>=4*(N-1) THEN GO TO SOLUT1;
CALL COUCHE(I+1,L(1));
END COUCHE;
DCL COUCHE ENTRY(FIXED BIN,LABEL);
INDIK1 :
IF SUBSTR(VARED,1,4)='0000' THEN VARED=SUBSTR(VARED,5);
DISPLAY('INDIK1');
IF VARED='' THEN DO;POINTE=SUBSTR(POINTE,1,4*LPOINTE);
CALL COUCHE(IV1-1,L(2));END;
C2=SUBSTR(VARED,1,2);
PUT SKIP DATA(C2);
IX(IV1)=(INDEX(LVAR,C2)+1)/2;
PUT SKIP DATA(IX(IV1));
VARED=SUBSTR(VARED,3);
CALL COUCHE(IV1,L(3));
INDIK2 :

```

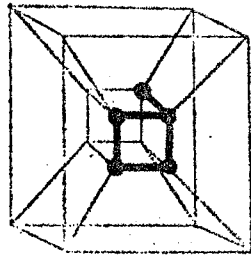
```

DISPLAY('INDIK2');
VARED='';
AB=A(1);BA=B(1);KAB=KA(1);
PUT SKIP DATA(AB,BA,KAB);
DO J=1B BY 100B TO 4*LPOINTE+100B;CC2=SUBSTR(POINTE,J,2);
IIX=(INDEX(LVAR,CC2)+1)/2;APREZ=PREZ(IIX);
AVOI=BOOL(APREZ,AB,'0001'B);
  AUW=BOOL(APREZ,BA,'0001'B);
  III=AVOI='0'B;JJJ=AUW='0'B;
  IF III=JJJ THEN VARED=VARED||VAR(IIX)||PRIM(IIX);
END;
VOI(IV1,KAB)=BOOL(AB,BA,'0111'B);
DO I=10B TO NL;
AB=A(I);BA=B(I);KAB=KA(I);
PUT SKIP DATA(BA,AB,KAB);
DO J=1B BY 100B TO LENGTH(VARED);CC2=SUBSTR(VARED,J,2);
IF CC2='00' THEN GO TO FINJ;IIX=(INDEX(LVAR,CC2)+1)/2;APREZ=PREZ(IIX);
AVOI=BOOL(APREZ,AB,'0001'B);
  AUW=BOOL(APREZ,BA,'0001'B);
  III=AVOI='0'B;JJJ=AUW='0'B;
  IF III=JJJ THEN SUBSTR(VARED,J,4)='0000';
FINJ : END;
VOI(IV1,KAB)=BOOL(AB,BA,'0111'B);
END;
PUT SKIP DATA(VARED);
IV1=IV1+1;GO TO INDIK1;
SOLUTI :
PUT EDIT('SOLUTION : ',POINTE)(PAGE,A(11),A);
PUT SKIP(3);
DO J=IV1 BY (-1) TO 1;
DO I=J TO IV1;
PUT EDIT(VOI(I,J)) (X(2),B(M));
END;
PUT SKIP;
END;
GO TO FINI;
IMPOSS :
PUT EDIT('AUCUNE SOLUTION')(PAGE,A);
FINI :
PUT LIST(IX) SKIP(5);
PUT SKIP DATA(IV1,IX(IV1),VARED,LVAR);
END ;
END JAM96;
E OF:

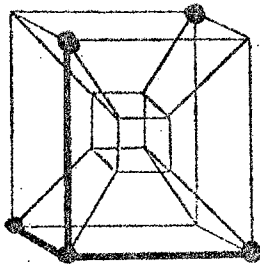
```



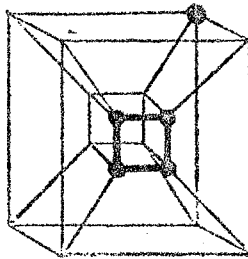
4



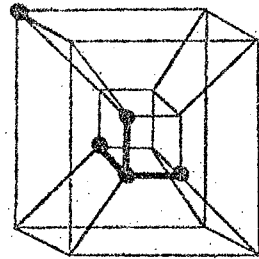
127/5



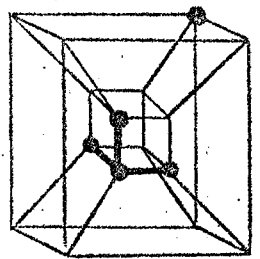
163/41



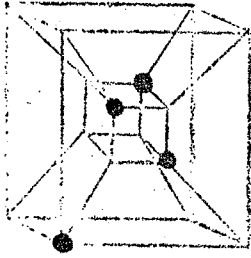
1234/41



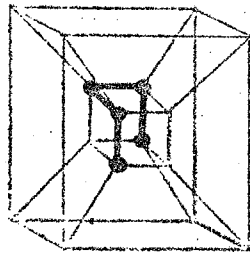
1153/41



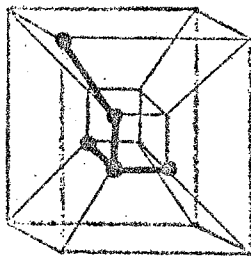
1333/41



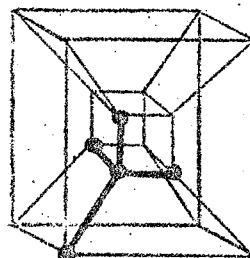
1050/1111



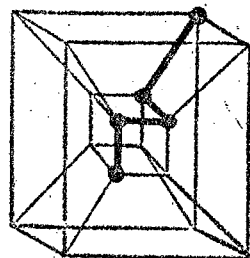
244/5



244/5

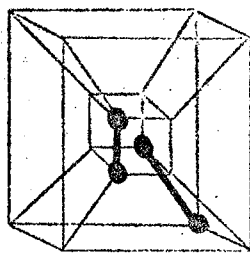


64/5

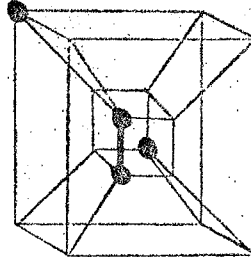


1234/5

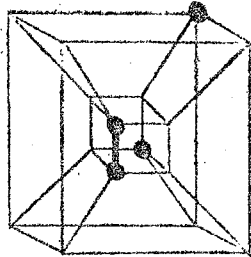
3



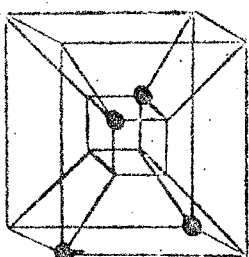
1212/22



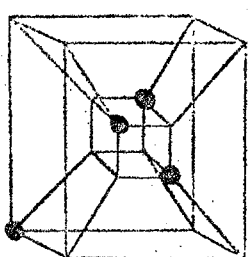
321/211



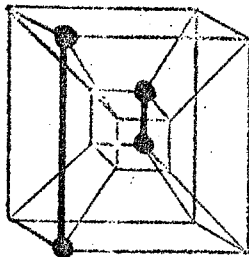
1221/211



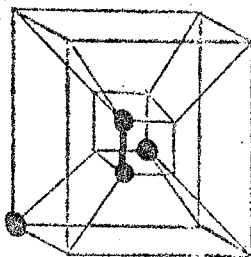
2040/1111



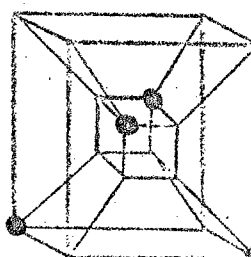
130/1111



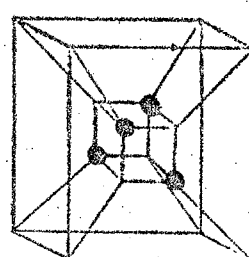
2202/22



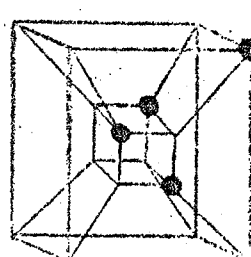
231/211



420/1111



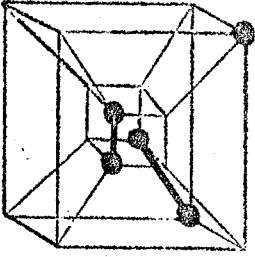
60/1111



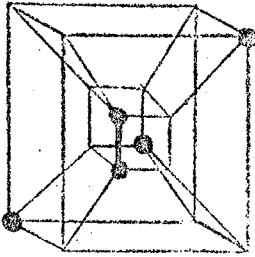
60/1111

6

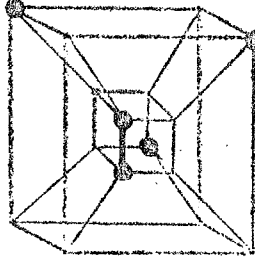
1432/221



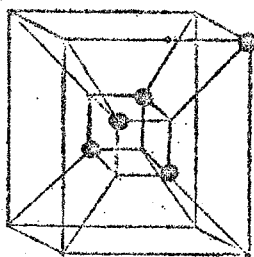
541/2111



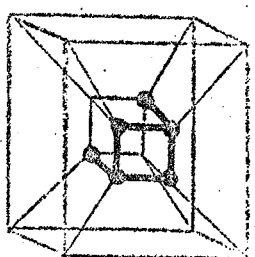
541/2111



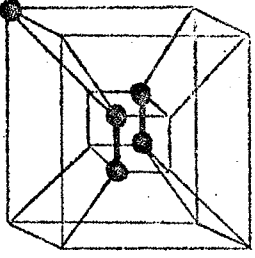
1090/11111



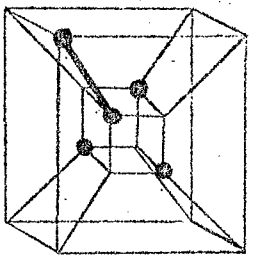
276/6



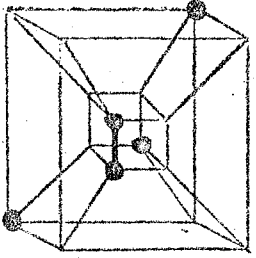
442/221



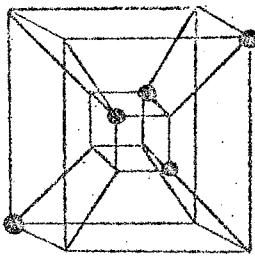
361/2111



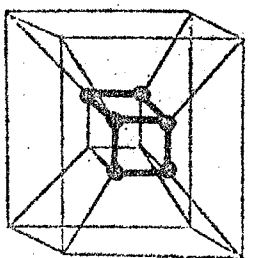
1351/2111



420/11111

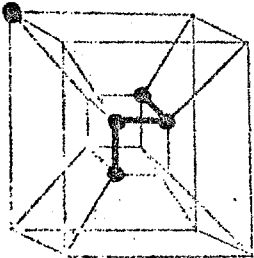


267/6

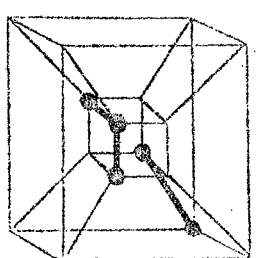


5

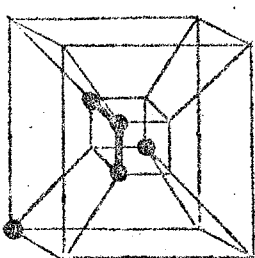
343/41



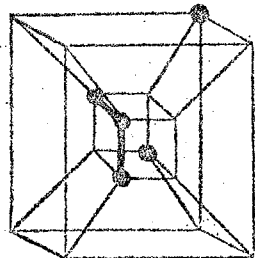
1333/32



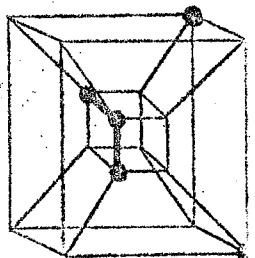
262/311



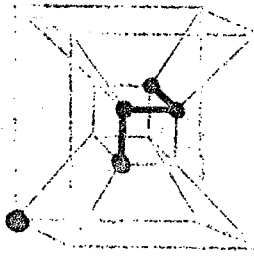
1252/311



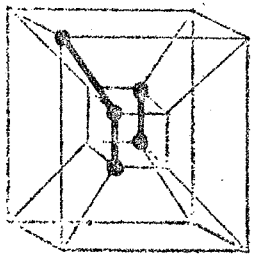
2242/311



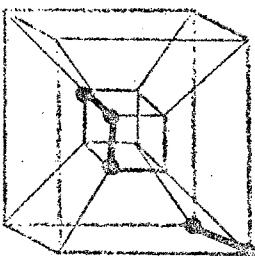
1333/41



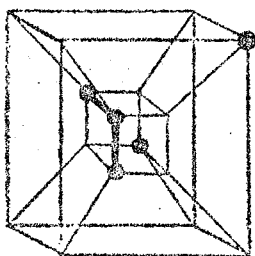
1333/32



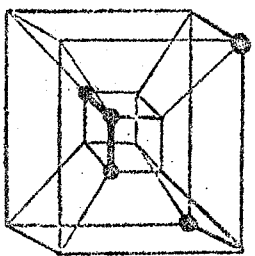
2323/32



442/311

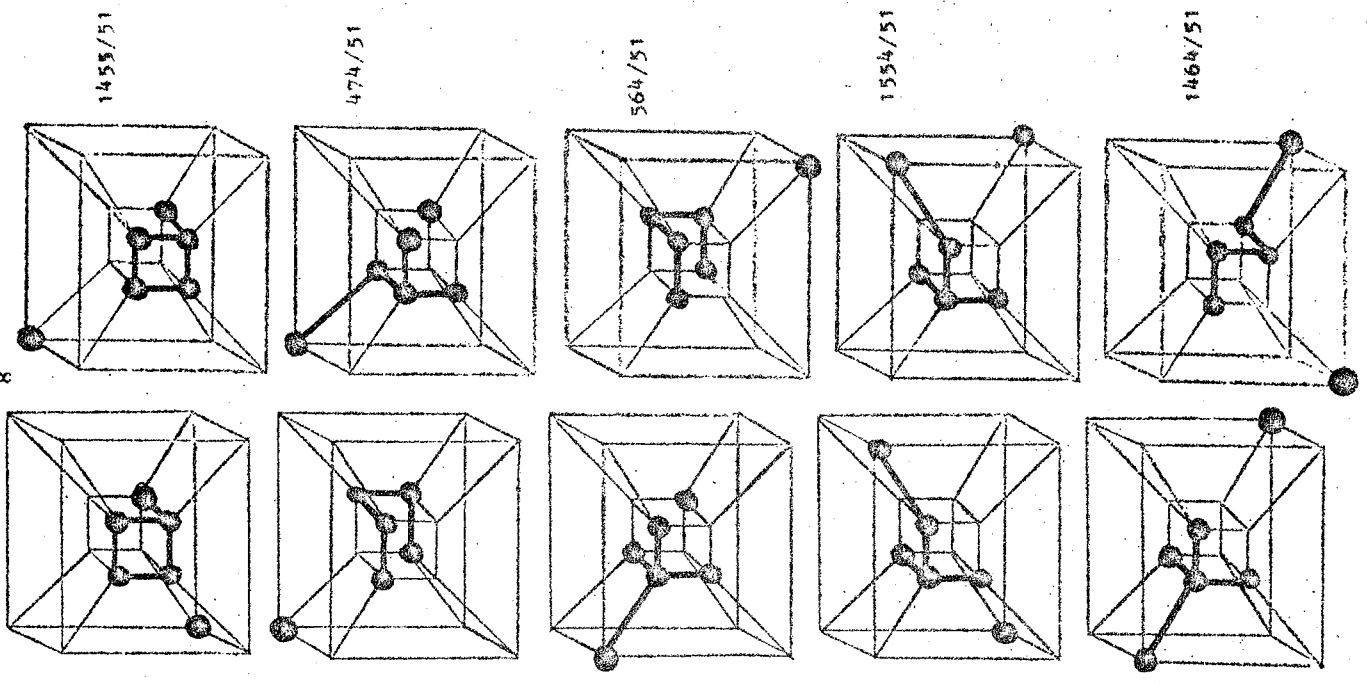


1432/311



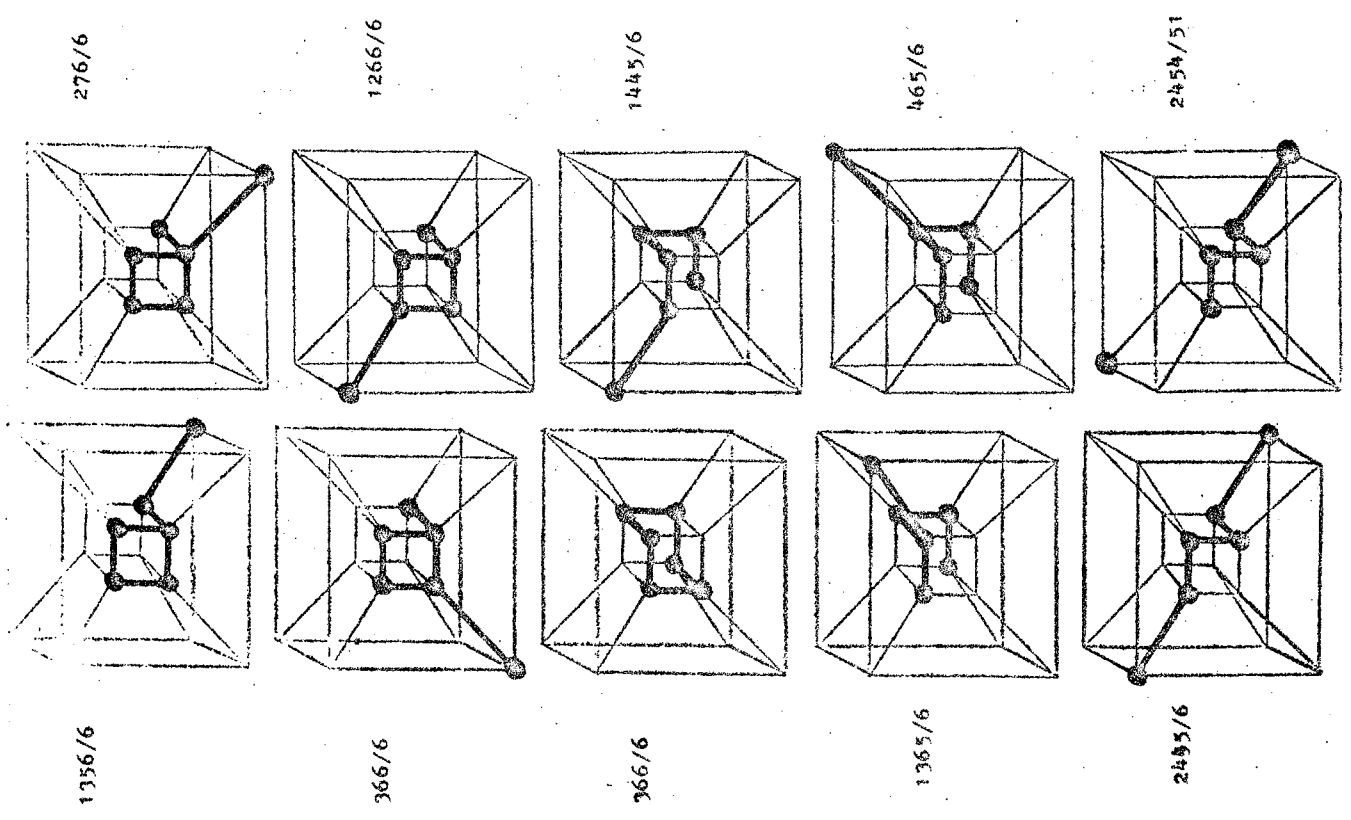


8



1365/51	474/51	1194/51	1464/51	1464/51
---------	--------	---------	---------	---------

7

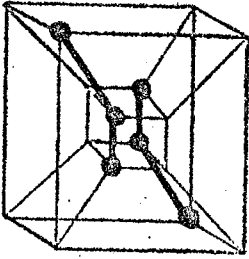


1336/6	366/6	366/6	1365/6	2455/6
--------	-------	-------	--------	--------

10

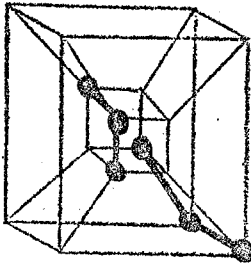
2544/33

bda



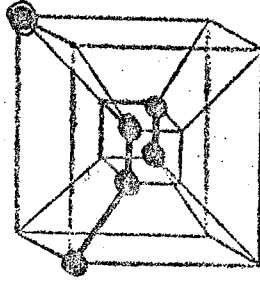
2554/33

ac'd'



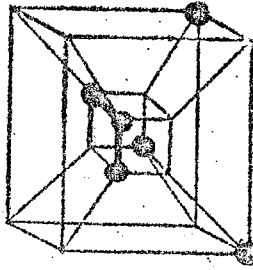
1563/321

adb'



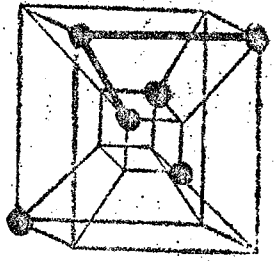
2382/311

ad'b



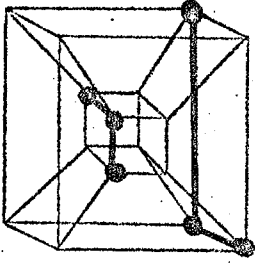
502/311

aucune solut.



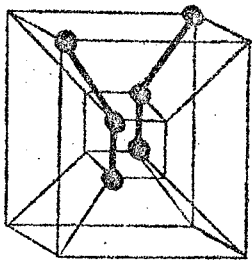
3444/33

acb



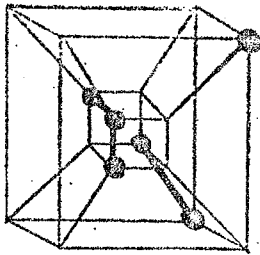
2454/33

acb



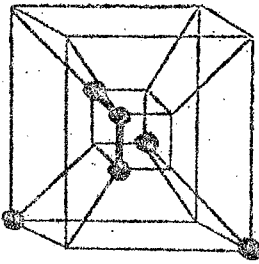
1653/321

aucune solut.



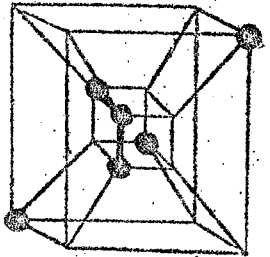
1392/311

ac'b



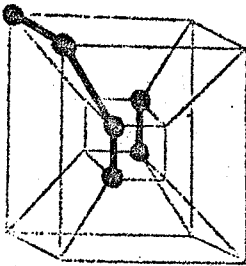
672/311

aucune solut.

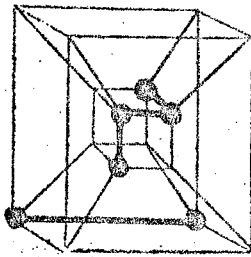


9

1554/42

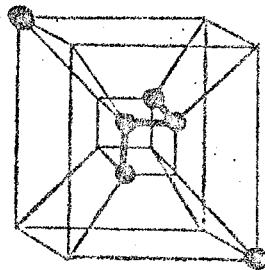


2544/42

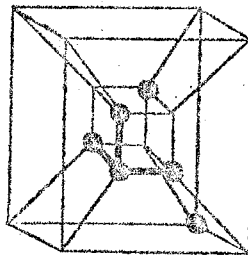


663/411

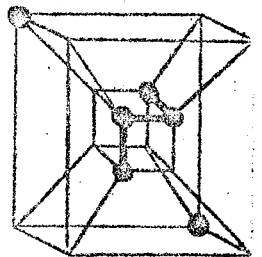
aucune solut.



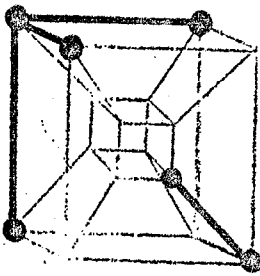
1293/411



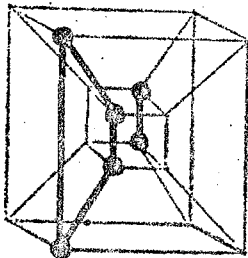
1563/411



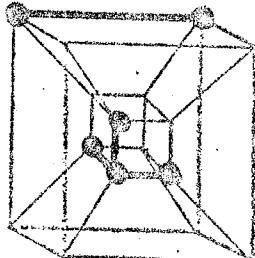
1464/42



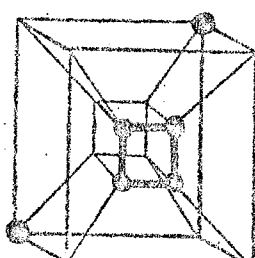
2445/42



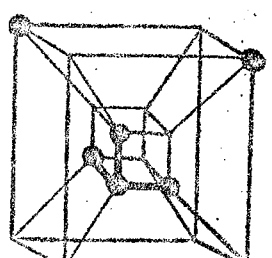
2454/42



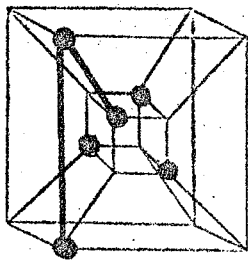
2554/411



2283/411

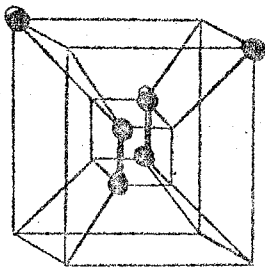


AA-



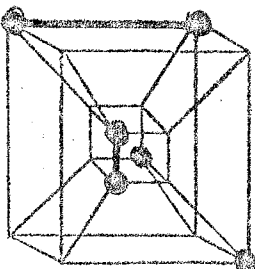
1392/3111

ba'c'



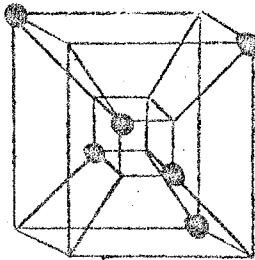
672/2211

aucune solut.

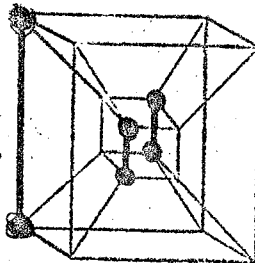


1662/2211

aucune solut.

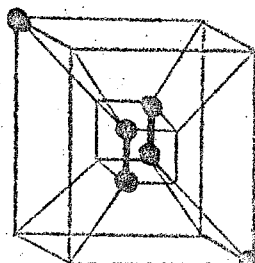


3050/11111



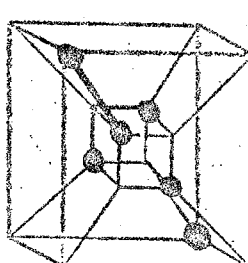
663/222

ab'd



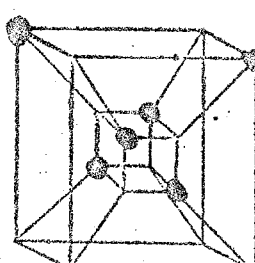
672/2211

cab'



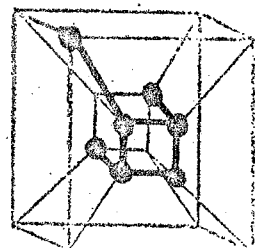
1491/21111

aucune solut.

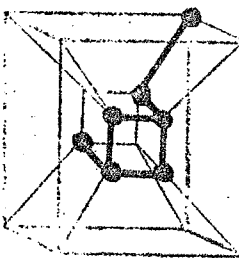


2050/111111

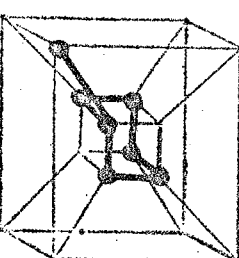
597/7



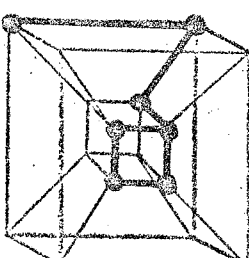
1347/7



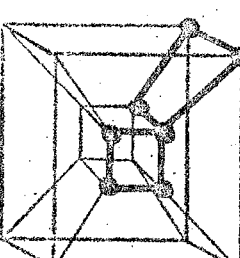
1587/7



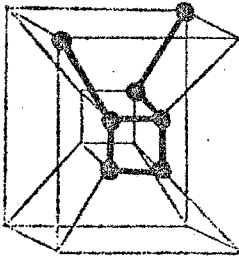
1587/7



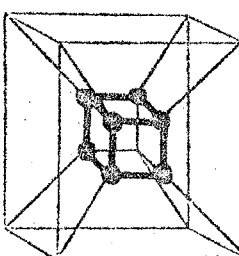
2577/7



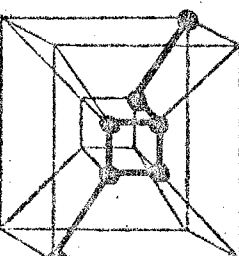
14887/7



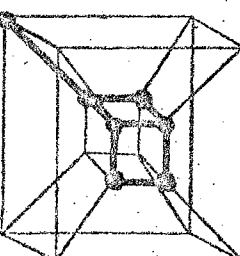
1587/7



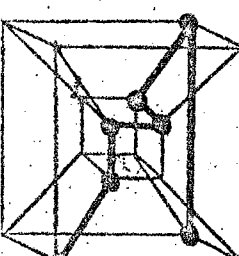
399/7



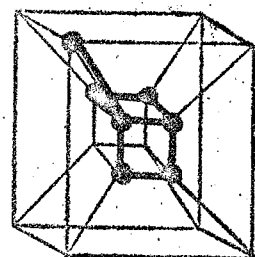
2577/7



14887/7

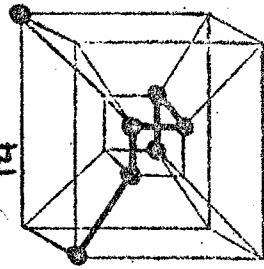


3666/7

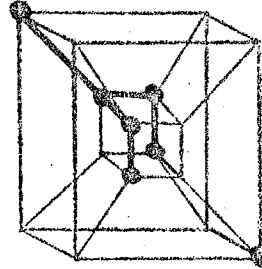


498/7

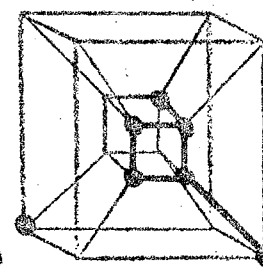
14



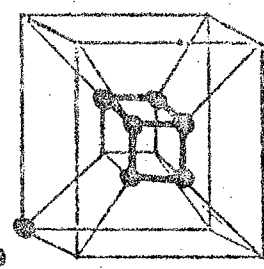
1785/61



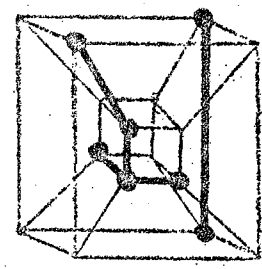
1785/61



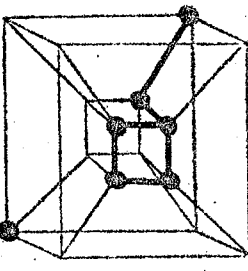
1686/61



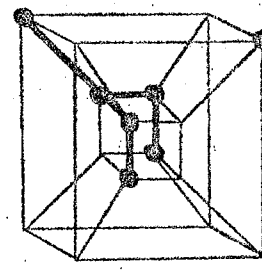
1587/61



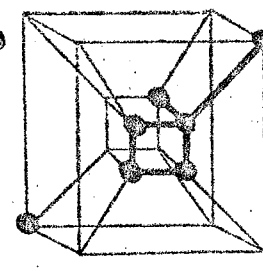
2775/52



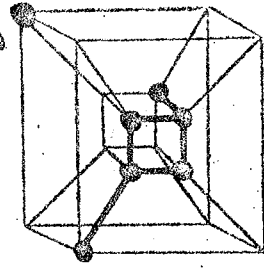
2676/61



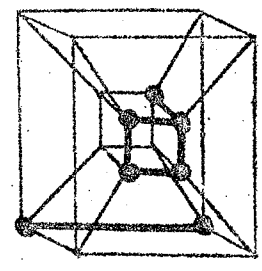
795/61



1686/61

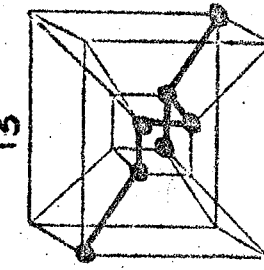


2496/61

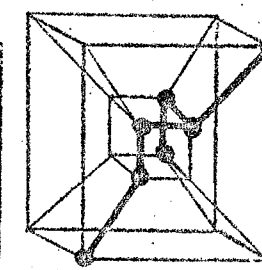


2676/52

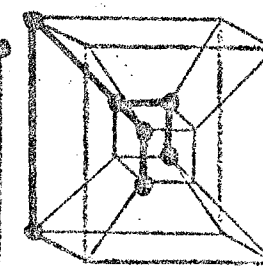
13



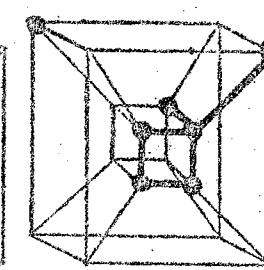
2676/7



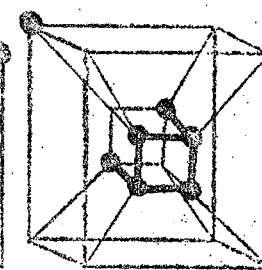
1686/7



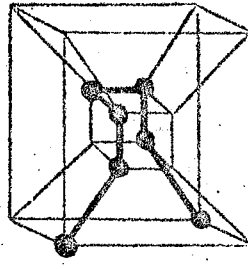
696/7



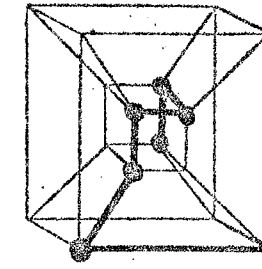
1445/61



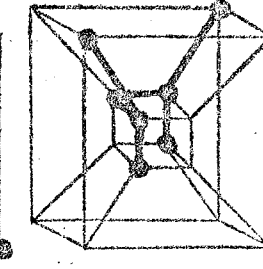
1446/61



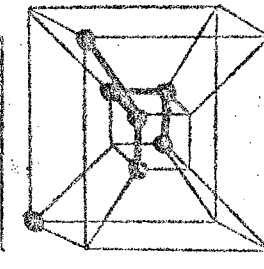
2676/7



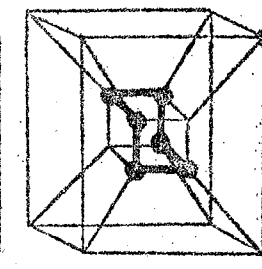
1686/7



2496/7



1545/61



696/61

16

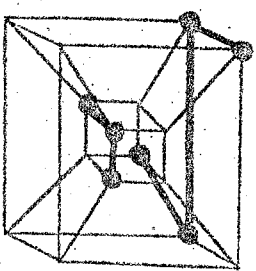
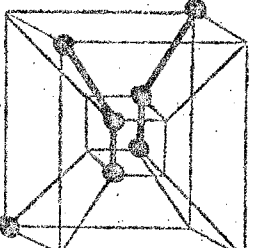
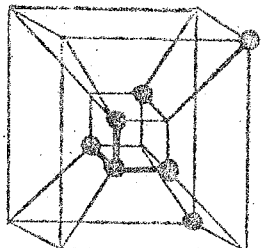
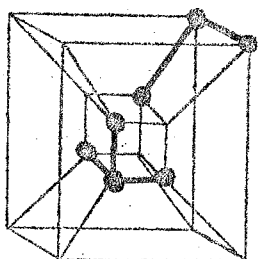
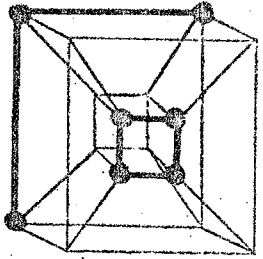
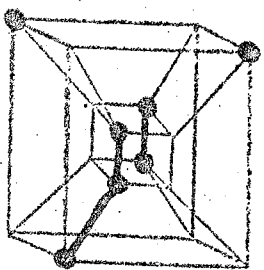
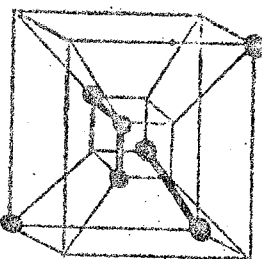
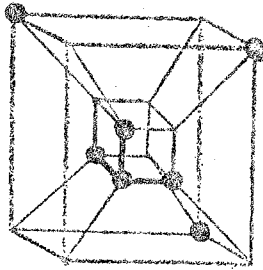
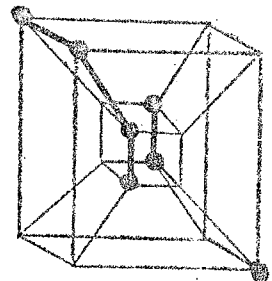
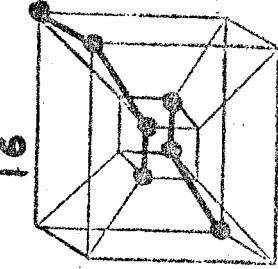
2775/43

1884/421

3309/4111

1884/331

17A3/3211



1785/52

3666/43

2694/511

2595/43

2204/511

2303/4111

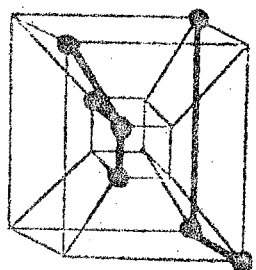
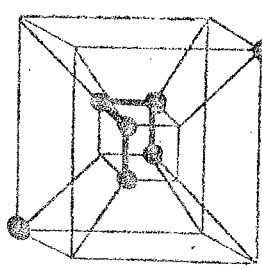
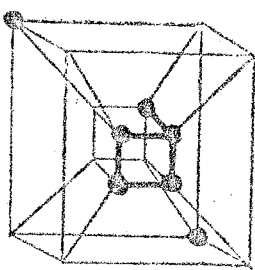
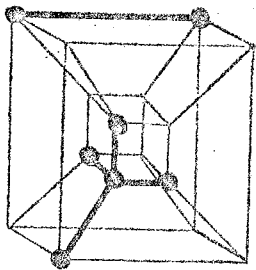
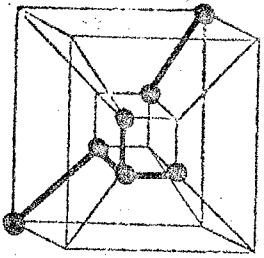
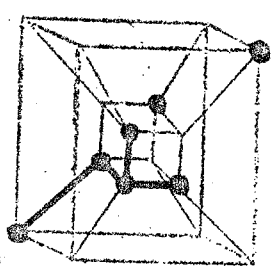
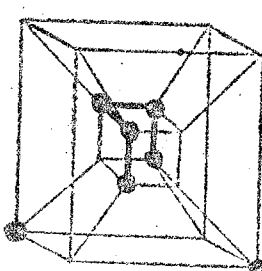
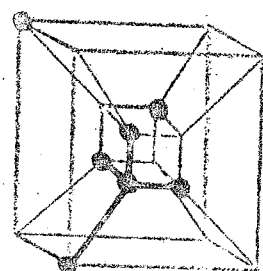
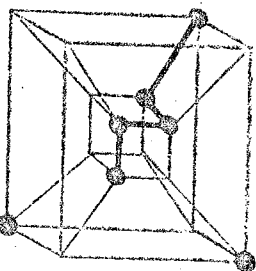
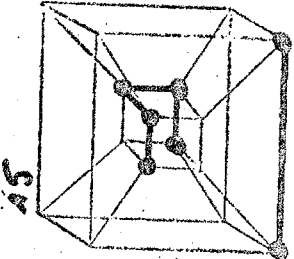
16A4/511

2694/331

16A4/511

2874/43

45



1785/52

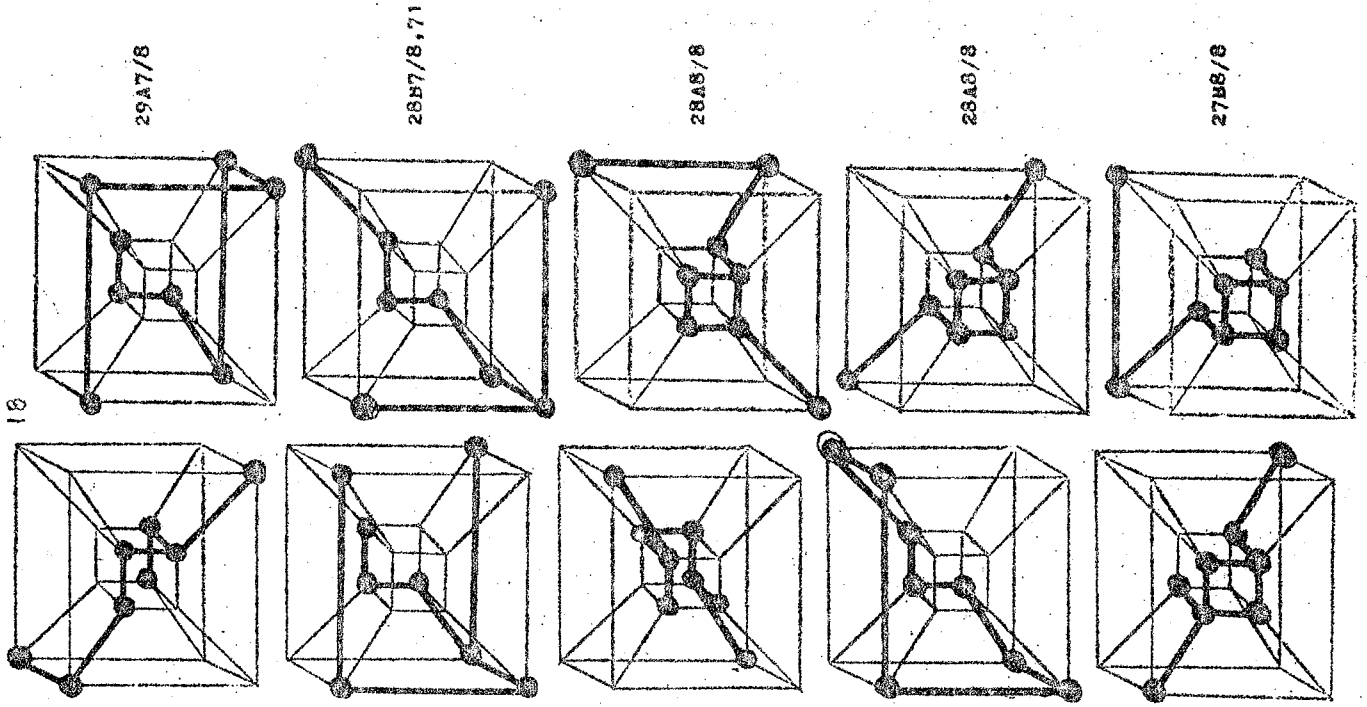
2595/52

2595/511

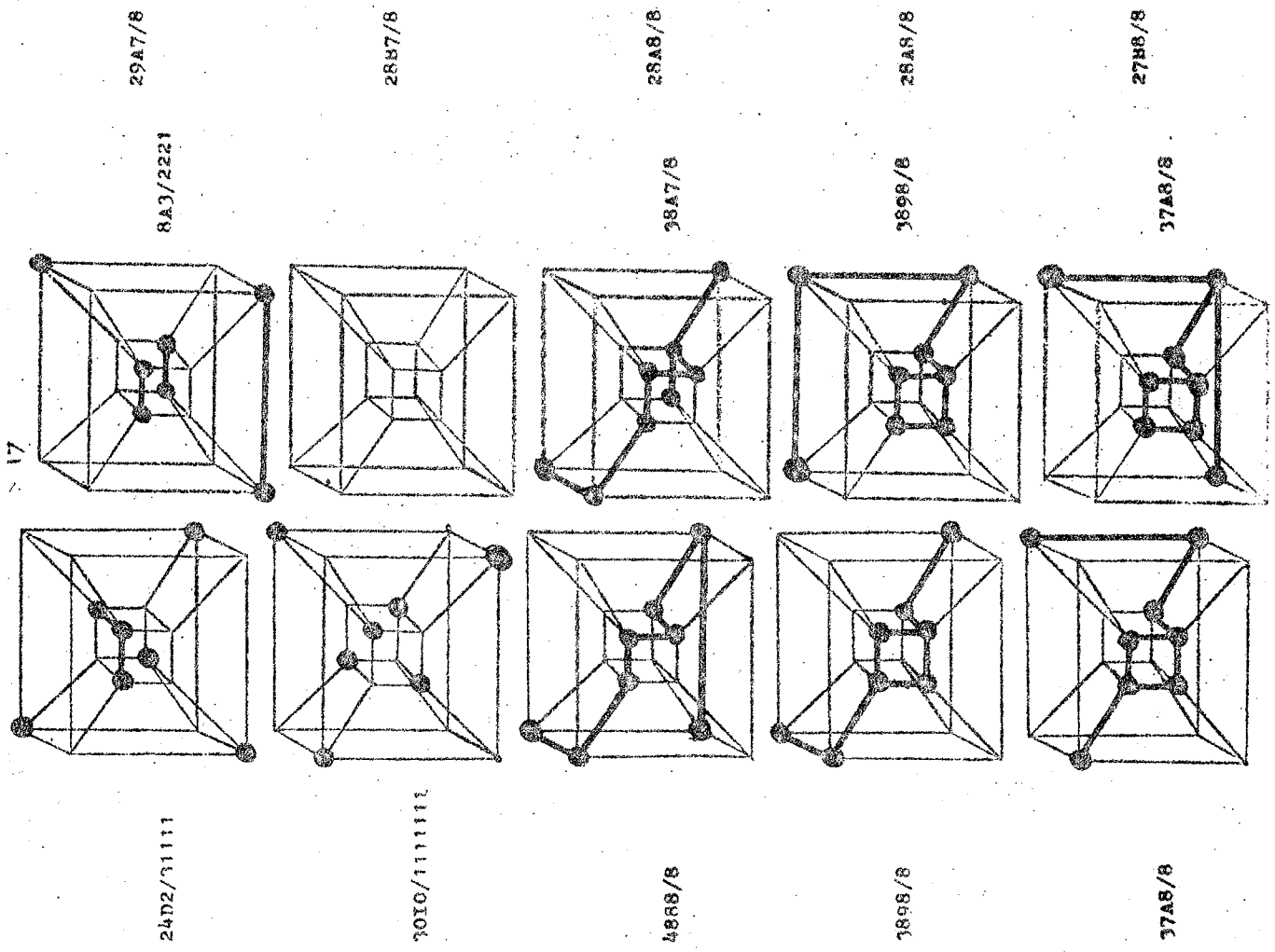
894/511

3585/43

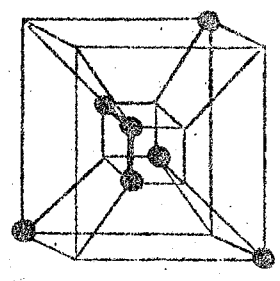
18



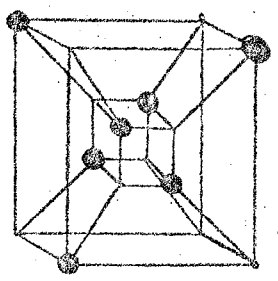
17



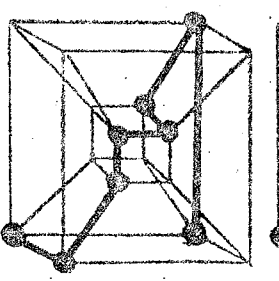
24D2/11111



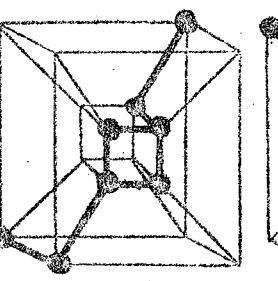
3010/1111112



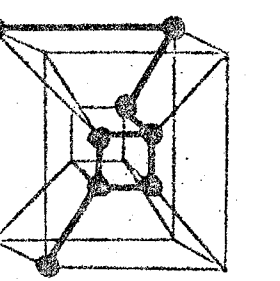
4868/8



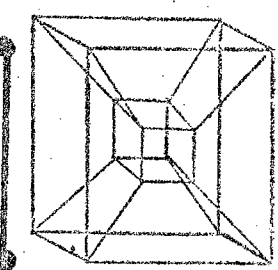
3898/8



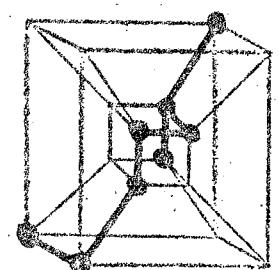
37A8/8



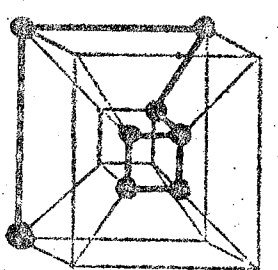
8A3/2221



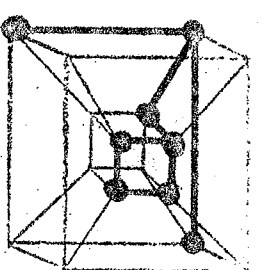
38A7/8



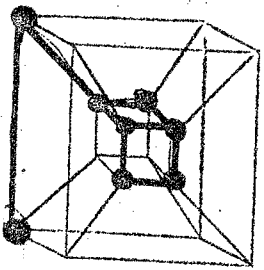
3898/8



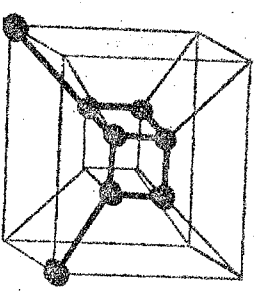
37A8/8



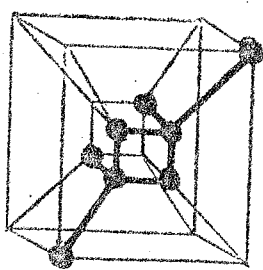
19



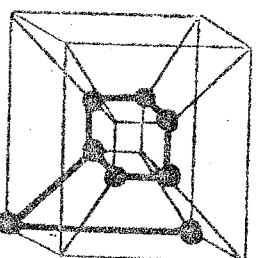
27A9/8



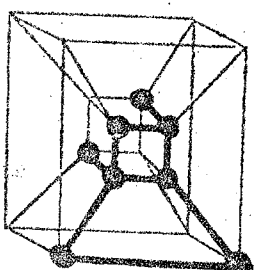
26B9/8



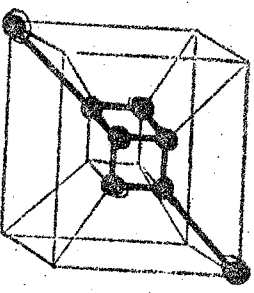
24B8/8



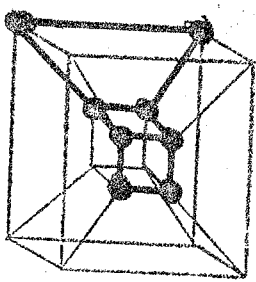
18B8/8



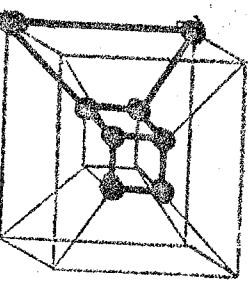
16C9/8



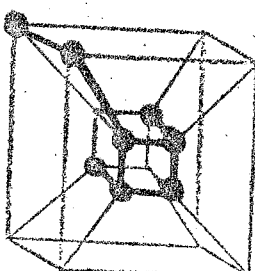
16BA/8



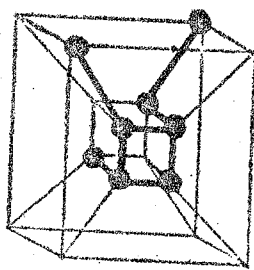
27A9/8



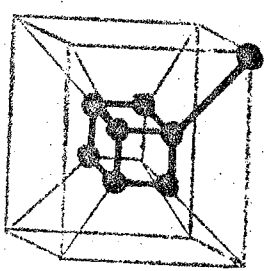
26AA/8



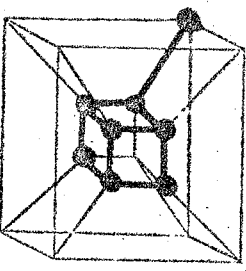
17C8/8



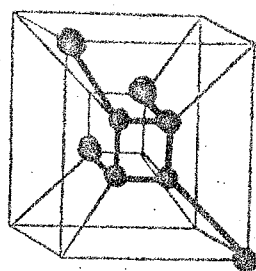
18B8/8



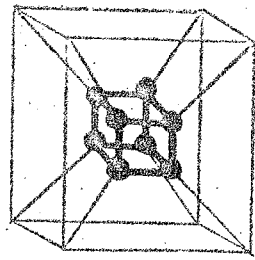
15CA/8



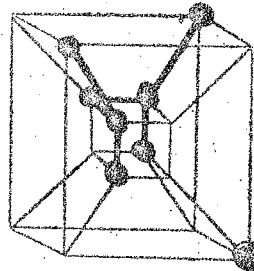
16BA/8



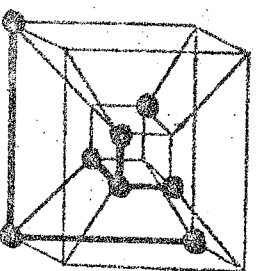
8C8/8



4CC/8

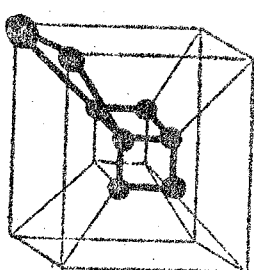


36D6/71,53

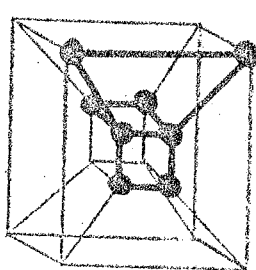


26B6/71,611

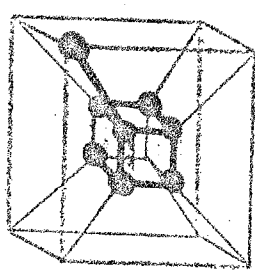
20



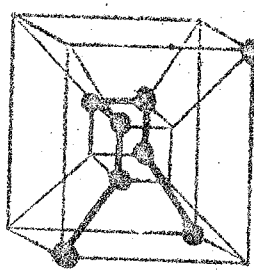
16BA/8



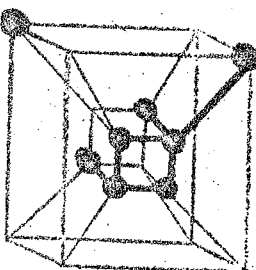
6CA/8



6CA/8



29B6/71,53



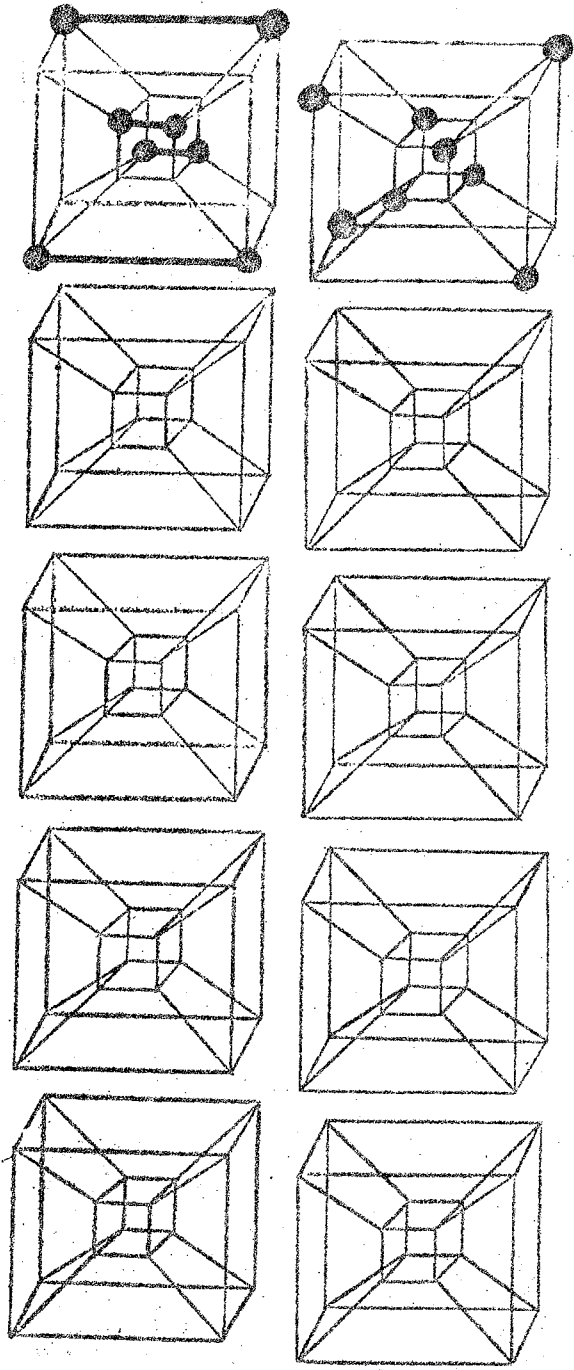
25E7/71





BD4/2222

4090/1111111



CP&TUSERID MONTAGND 11/19/70 09.22.48 \*\*\*\*\*

ZONE DSECT F:0  
 NCM&RE1 DC 2F  
 N DC F:6  
 MINF DC F:8  
 MSUP DC F:9  
 DEC4N DS F  
 DEC&MBIS DS F  
 DEC4NI DS F  
 N2 DC F:10000  
 N3 DC F  
 BORNA DS F  
 AVAR DS A(VAR)  
 ZERO DC F:0  
 AVARN DS F  
 M DS F  
 AVARV2 DS F  
 AVARV3 DS F  
 AMELICR2 DS F  
 SAUVR DS F  
 BORNLINF DS 4F  
 ACRPRODI DC V(PRODUIT1)  
 ACRPROB2 DC V(PRODUIT2)  
 ACRSPG DC V(SPG)  
 ADELIM DC V(ELIMINI)  
 ACRCHER DC V(CHERSIJ)  
 ACRCHERX DC V(CHERSIJX)  
 ACRCHI DC V(CHERSIJ)  
 ACRCHI DC V(CHERSIJ)  
 ACRCHI DC V(CHERSX1)  
 ACRCHI DC V(XIXJSZ2)  
 ADRPG DC V(PG)  
 ADRPG1 DC V(PG1)  
 BIT53 DS F  
 BIT52 DS F  
 BITINE DC XL4\*FF00000  
 BITSUP DC XL4\*00FF8000  
 BITUN DC XL4\*FFFF8000  
 AMELIOR DS 100F  
 SORTS22 DC CL13\*SORTIE S22 :  
 SORT53 DC CL7\*S22=S23  
 CHAINE DC CL70  
 CHAINEB DS CL20  
 CHAINEM DS F  
 BLANC DC CL40  
 PARENT2 DC CL15  
 SIGNE DS CL1  
 SIGNES22 DS CL1  
 SIJ DS CL3  
 SAS DS F:4  
 BS11 DC CL4\*S11 ,F:0  
 BJ11 DC CL4\*D11 ,F:0  
 BS12 DC CL4\*S12 ,F:0  
 BJ12 DC CL4\*D12 ,F:0  
 BS21 DC CL4\*S21 ,F:0  
 BJ21 DC CL4\*D21 ,F:0  
 BS22 DC CL4\*S22 ,F:0  
 BJ22 DC CL4\*D22 ,F:0  
 BS13 DC CL4\*S13 ,F:0  
 BJ13 DC CL4\*D13 ,F:0  
 BS23 DC CL4\*S23 ,F:0

B023 DC CL4\*D23 ,F:0  
 BS31 DC CL4\*S31 ,F:0  
 BJ31 DC CL4\*D31 ,F:0  
 BS32 DC CL4\*S32 ,F:0  
 BJ32 DC CL4\*D32 ,F:0  
 BS33 DC CL4\*S33 ,F:0  
 BJ33 DC CL4\*D33 ,F:0  
 BS14 DC CL4\*S14 ,F:0  
 BJ14 DC CL4\*D14 ,F:0  
 BS24 DC CL4\*S24 ,F:0  
 BJ24 DC CL4\*D24 ,F:0  
 BS34 DC CL4\*S34 ,F:0  
 BJ34 DC CL4\*D34 ,F:0  
 BS15 DC CL4\*S15 ,F:0  
 BJ15 DC CL4\*D15 ,F:0  
 BS25 DC CL4\*S25 ,F:0  
 BJ25 DC CL4\*D25 ,F:0  
 BS35 DC CL4\*S35 ,F:0  
 BJ35 DC CL4\*D35 ,F:0  
 PARENTEZ DC CL5  
 TRIPLET DS CL5  
 SORTIE DS CL5  
 OF DC CL8\*TYPLIN  
 DC AL111  
 DC AL3(CHAINI)  
 DC C\*K  
 DC AL3(70)  
 DG OD  
 DS CL8\*CONVAIT  
 DC CL4\*CONI  
 DC 800F:0  
 DC CL48\*Y Y Z T U V W A B C D E  
 DC CL48\*F G H I J K L M N O P Q  
 CL2000  
 CSECT  
 BALR R15,0  
 USING \*,R15  
 L USING R11,ADONNEES  
 USING R11,ADONNEES  
 USING ZONE,R11  
 ENTRY PRODUIT1,PRODUIT2,ELIMINI,CHERSIJ,SPG,XIXJSZ2  
 ENTRY CHERSI,CHERSX1,CHERSIJX,CASSI  
 EXTRN DONNEES  
 \* SOUS-PG DE COMPARAISON AVEC LES CHAINES DEJA EXISTANTES  
 LA R4,0  
 SR R6,R6  
 LR R7,R3  
 O R7,TAB(R6)  
 C R7,TAB(R6)  
 BE COMPD CAS D'ELIMINATION  
 A R6,=F:4  
 C R6,BORNA  
 BNE COMPA  
 CR R6,R5  
 BE COMPE  
 LR R7,R3  
 O R7,TAB(R6)  
 C R7,TAB(R6)  
 BE COMPD CAS D' ELIMINATION  
 CR R7,R3  
 BNE COMPC ON N'A PAS A ELIMINER UNE CHAINE PRECEDENMENT  
 S R5,=F:4 AJOUTEE

CR R6,R5  
 BE COMPE ON ELIMINE LA DERNIERE AJOUTEE  
 LA R8,TAB DANS R8 ADRESSE DU DERNIER AJOUTE  
 AR R8,R5  
 LA R9,TAB  
 AR R9,R6 DANS R9 ADRESSE OU CN VA L'INSERER : A  
 MVC 014,R9,01R8) L'ADRESSE DE L'ELIMINE  
 LR R9,R6  
 C R5,NZ  
 BL CAS1  
 S R9,N2  
 SRA R9,2  
 M R8,=F,5'  
 A R9,AVARV2  
 S R10,=F,5'  
 MVC 015,R9,01R10)  
 B COMPB  
 L R8,AVAR  
 AR R8,R5  
 A R9,AVAR  
 MVC 013,R9,01R8)  
 B COMPB  
 A R6,=F,4'  
 CR R6,R5  
 BE COMPE  
 B COMPB  
 LA R4,1  
 BR 14  
 L R6,M  
 SR R5,R5  
 LR R7,R3  
 SRL R7,1  
 SLA R7,1  
 BNO SUITEST  
 A R5,=F,1'  
 BCT R6,RETEST  
 BR 14  
 \* FERMATION DE X-Y  
 ETI L R3,BITS22  
 B ET2  
 ET L R3,TAB(R1)  
 ET2 LR R4,R3  
 N R3,TAB(R2)  
 O R4,TAB(R2)  
 N R4,BISUP  
 OR R3,R4  
 BR 14  
 ST R3,SAUVRET  
 L R4,DEC4N  
 SR R2,R2  
 L R3,TAB(R2)  
 OR R3,R1  
 CR R3,R1  
 BNE NONELIMI  
 S R4,=F,4'  
 CR R2,R4  
 BE FINELIMI  
 L R5,TAB(R4)  
 LR R5,R4  
 A R5,AVAR

LR R6,R2  
 A R6,AVAR  
 MVC 011,R6,01R5)  
 B ELIMBISI  
 NONELIMI A R2,=F,4'  
 CR R2,R4  
 BNE ELIMBISI  
 ST R4,DEC4N NOUVELLE VALEUR DE DEC4N  
 S R4,=F,4'  
 ST R4,DEC4N1  
 A R4,=F,8'  
 ST R4,DEC4NBIS  
 L R4,DEC4N  
 A R4,AVAR  
 ST R4,AVARN  
 B FINSPG  
 ST R2,SAUVRET  
 SR R1,R1  
 LR R2,R1  
 A R2,=F,4'  
 BAL 14,ET  
 BAL 14,COMP  
 C R4,=F,1'  
 BE UNDU  
 MVI SIGNE,C,1'  
 BAL 14,CHARGE1  
 BAL 14,OU  
 BAL 14,COMP  
 C R4,=F,1'  
 BE NONET  
 MVI SIGNE,C,1'  
 BAL 14,CHARGE1  
 A R2,=F,4'  
 C R2,DEC4N  
 BNE UNET  
 A R1,=F,4'  
 C R1,DEC4N1  
 BL UNET1  
 ST R5,NZ  
 A R5,AVARV2  
 B FINSPG  
 ST R2,SAUVRET  
 L R1,BORNIINF  
 L R2,BORNIINF+8  
 CLC ADPRG(4),ADPRG1  
 BL VARET2  
 LA R3,TAB-1  
 AR R3,R2  
 CLI 01R3),C,N'  
 BE VARETS  
 BAL 14,ET  
 BAL 14,COMP  
 C R4,=F,1'  
 BE VAROU  
 MVI SIGNE,C,1'  
 BAL 14,CHARGE2  
 BAL 14,OU  
 BAL 14,COMP  
 C R4,=F,1'  
 BE VARETS

CHARGE1	ST	R3,TAB(R5)
LR	LR	R7,R1
A	A	R7,AVAR
L	L	R10,AVAR
AR	AR	R10,R5
MVC	MVC	O(1,R10),O(R7)
MVC	MVC	I(1,R10),SIGNE
LR	LR	R7,R2
A	A	R7,AVAR
MVC	MVC	2(1,R10),O(R7)
A	A	R5,=F,4,
BR	BR	14
ST	ST	R3,TAB(R5)
CHARGE2	ST	R5,=F,4,
ST	ST	ADP(4),ADP(4)
A	CLC	ADP(4),ADP(4)
CLC	CLC	CHARGE2
BL	BL	R4,TAB-4
LA	LA	R4,R2
AR	AR	O(3,R10),O(R4)
MVC	MVC	CHARGE2B
B	B	R3,R2
CHARGE2A	LR	R3,AVAR
LR	A	O(3,R10),O(R3)
A	MVC	3(1,R10),SIGNE
MVC	MVC	R3,R1
LR	LR	R3,AVAR
A	A	4(1,R10),O(R3)
MVC	MVC	R10,=F,5,
A	A	14
BR	BR	SIJ+1(1),CHAINED+2
MVC	MVC	SPEICHER(4),ADP(4)
MVC	MVC	R15,ADP(4)
ST	ST	TRIPLET(1),CHAINED+7
MVC	MVC	R10,S11C CORRESPOND A X-Z/X+Z
LA	LA	R7,BITS3
LA	LA	R8,CHERSIJ
BAL	BAL	R3,BITS3
L	L	14,TESTI
BAL	BAL	R5,BORNINF
C	C	S11D
BL	BL	R3,BS11+4
ST	ST	BORNINF+4(3),TRIPLET
MVC	MVC	R5,BORNINF
ST	ST	SIJ(1),SIJ+1
MVC	MVC	R10
BR	BR	TRIPLET+2(1),CHAINED+2
MVC	MVC	SIJ+1(1),CHAINED+4
MVC	MVC	R10,S11B
BAL	BAL	CHAINED+3(1),CHAINED+12
CLC	CLC	ZURRUCK
BNE	BNE	TRIPLET+2(1),CHAINED+11
MVC	MVC	R10,S11B
BAL	BAL	TRIPLET(1),CHAINED+2
MVC	MVC	R10,S11B
BAL	BAL	TRIPLET(1),CHAINED+4
MVC	MVC	R10,S11B
BAL	BAL	TRIPLET+2(1),CHAINED+13
MVC	MVC	R10,S11B
BAL	BAL	TRIPLET(1),CHAINED+2
MVC	MVC	R10,S11B
BAL	BAL	TRIPLET(1),CHAINED+7
MVC	MVC	

VARETS	MVI	SIGNE,C**
	BAL	I4,CHARGE2
	A	R2,PAS
	C	R2,BORNINF+12
	BL	VARET
	A	R1,=F,4,
	C	R1,BORNINF+4
	BL	VARET1
	ST	R5,N3
	ST	R10,AVAR3
	B	FINSPG
XIXJS22	ST	R3,SAUVRET
XIXJS22A	BAL	14,ETI
	LR	R4,R3
	O	R4,BITS22
	C	R4,BITS22
	BE	ETS22
	LR	R4,R3
	O	R4,BS22+4
	C	R4,BS22+4
	BE	ETS22
	BAL	14,TESTI
	C	R5,NOMBRE1
	BNH	ETS22
	MVC	PARENT2+3(3),PARENT2
	ST	R5,NOMBRE1
	MVI	R2,AMELIOR
	ST	AMELIOR+3,C**
	BAL	R3,TAB(R10)
ETS22	LR	14,OU1
	O	R4,R3
	C	R4,BITS22
	BE	OUS22
	LR	R4,R3
	O	R4,BS22+4
	C	R4,BS22+4
	BE	OUS22
	BAL	14,TESTI
	C	R5,NOMBRE1
	BNH	OUS22
	MVC	PARENT2+3(3),PARENT2
	ST	R5,NOMBRE1
	STH	R2,AMELIOR
	MVI	AMELIOR+3,C**
	ST	R3,TAB(R10)
	A	R2,=F,4,
OUS22	C	R2,BORNA
	BNE	XIXJS22A
	B	FINSPG
	L	FORMATION DE X+Y
	B	R3,BITS22
OU1	L	OU2
OU	L	R3,TAB(R1)
OU2	LR	R4,R3
	N	R3,TAB(R2)
	O	R4,TAB(R2)
	N	R4,BITINF
	OR	R3,R4
	BR	14
		MEMBRISATION DE X.Y/X+Y

BAL	R10, S11B
L	R3, BS12+4
BAL	14, TEST1
ST	R5, BORNLINE
MVC	TRIPLET(3), BS11
MVC	TRIPLET(3), CHAINEB+3
LA	R10, XYZTU
CLI	TRIPLET+3, C, .
BE	XYZTU3
NVI	TRIPLET+3, C, .
B	XYZTU
NVI	TRIPLET+3, C, .
LA	R7, BITS3
BAL	R8, CHERSX1
L	R3, BITS3
BAL	14, TEST1
C	R5, BORNLINE
BL	XYZTU2
ST	R3, BS12+4
ST	R5, BORNLINE
BR	R10
MVC	TRIPLET+4(1), CHAINEB+2
BAL	R10, XYZTU
MVC	TRIPLET+4(1), CHAINEB+7
BAL	R10, XYZTU
MVC	TRIPLET+4(1), CHAINEB+11
BAL	R10, XYZTU
MVC	TRIPLET+4(1), CHAINEB+13
BAL	R10, XYZTU
MVC	SAUVRET(4), BORNLINE+8
MVC	ADRRP(4), SPEICHER
B	FINSPG
ST	R8, SAUVRET
L	R9, AVARN
CLI	TRIPLET+1, C, .
BE	CHER11
MVI	TRIPLET+1, C, .
B	CHER1
MVI	TRIPLET+1, C, .
CLC	TRIPLET(3), O(1R9)
BNE	CHER1
S	R9, AVAR
LA	R8, TAB
AR	R8, R9
MVC	O(4, R7), O(1R8)
B	FINSPG
A	R9, F, 4, .
C	R9, AVARV2
BL	CHER1
B	FINSPG
ST	R8, SAUVRET
L	R9, AVARN
CLC	TRIPLET(3), O(1R9)
BNE	CHER1
S	R9, AVAR
LA	R8, TAB
AR	R8, R9
MVC	O(4, R7), O(1R8)
B	FINSPG
A	R9, F, 4, .
C	R9, AVARV2
BL	CHER1
B	FINSPG
ST	R8, SAUVRET
L	R9, AVARN
CLC	TRIPLET(3), O(1R9)
BNE	CHER1
S	R9, AVAR
LA	R8, TAB
AR	R8, R9
MVC	O(4, R7), O(1R8)
B	FINSPG
A	R9, F, 4, .
C	R9, AVARV2

BL	CHERB
B	FINSPG
CHERSIJK	R8, SAUVRET
ST	R9, AVAR
L	TRIPLET(1), O(1R5)
CLC	CHARKAI
BNE	R8, AVAR
L	TRIPLET+2(1), O(1R8)
CLC	CHARSXE1
BE	R8, F, 4, .
A	CHARSXF1
B	R9, AVAR
S	R1, R9
LR	R8, AVAR
S	R2, R8
LR	14, SAUVR
ST	TRIPLET+1, C, .
CLI	CHARXCI
BE	14, ET
BAL	CHARXDI
B	14, OU
BAL	R4, R3
LR	14, SAUVR
L	R3, O(1R7)
ST	FINSPG
B	R9, F, 4, .
A	R9, AVARN
C	CHARXBI
BL	FINSPG
B	R8, SAUVRET
ST	R9, AVAR
L	TRIPLET+4(1), O(1R9)
CLC	CHERXAI
BNE	CHERXAI
LA	R8, BS11
CLC	TRIPLET(3), O(1R8)
BE	CHERSXE1
A	R8, F, 4, .
B	CHERSXF1
S	R9, AVAR
LR	R1, R9
LR	R2, R8
A	R2, F, 4, .
LA	R3, TAB
SR	R2, R3
ST	14, SAUVR
CLI	TRIPLET+3, C, .
BE	CHERXCI
BAL	14, ET
B	CHERXDI
BAL	14, OU
LR	R4, R3
L	14, SAUVR
ST	R3, O(1R7)
B	FINSPG
A	R9, F, 4, .
C	R9, AVARN
BL	CHERXBI
L	R4, SAUVRET
L	R15, ADRPG
BR	R4

SAUVRET DS F  
 SPEICHER DS F  
 ADDRESS DC A (DONNEES)  
 R1 EQU 12  
 R2 EQU 2  
 R3 EQU 3  
 R4 EQU 4  
 R5 EQU 5  
 R6 EQU 6  
 R7 EQU 7  
 R8 EQU 8  
 R9 EQU 9  
 R10 EQU 10  
 R11 EQU 11  
 R15 EQU 13  
 END SPGA

CP67USERIO MONTAGND  
 ZONE DSECT F:0'  
 NOMBREI DS 2F  
 S22 DS F:8'  
 N DC F:5'  
 MINF DC F:11'  
 MSUP DC F  
 DEC4N DS F  
 DEC4NBIS DS F  
 DEC4NI DS F  
 N2 DC F:10000'  
 N3 DC F  
 BORNA DS F  
 AVAR DC A (VAR)  
 ZERO DC F:0'  
 AVARN DS F  
 M DS F  
 AVARV2 DS F  
 AVARV3 DS F  
 AMELICR2 DS F  
 SAUVR DS F  
 BORNINF DS 4F  
 ADRPRODI DC V (PRODUIT1)  
 ADRPRODZ DC V (PRODUIT2)  
 ADRSPG DC V (SPG)  
 ADELIM DC V (ELIMINI)  
 APCRHER DC V (CHERSIJ)  
 APCRHERX DC V (CHERSIJX)  
 ADRCHI DC V (CHERSI)  
 ADRCHX1 DC V (CHERSX1)  
 ADRXIXJ DC V (XIXJS22)  
 ADRPG DC V (PG)  
 ADRPG1 DC V (PG1)  
 BITS3 DS F  
 BITS22 DS F  
 BITINF DC XL4:F8000000'  
 BITSUP DC XL4:07FF0000'  
 BITUN DC XL4:FFFF0000'  
 AMELIGR DS 100F  
 SORTS22 DC CL13:SORTIE S22 :  
 SORTS3 DC CL7:S22-S23:  
 CHAINE DC CL70:  
 CHAINEM DS CL20  
 BLANC DC CL40:  
 PARENT2 DC CL15:11 ) ( )  
 SIGNE DS CL1  
 SIGNES22 DS CL1  
 S1J DS CL3  
 PAS DC F:4'  
 BS11 DC CL4:S11 ,F:0'  
 BD11 DC CL4:D11 ,F:0'  
 BS12 DC CL4:S12 ,F:0'  
 BD12 DC CL4:D12 ,F:0'  
 BS21 DC CL4:S21 ,F:0'  
 BD21 DC CL4:D21 ,F:0'  
 BS22 DC CL4:S22 ,F:0'  
 BD22 DC CL4:D22 ,F:0'  
 BS13 DC CL4:S13 ,F:0'  
 BD13 DC CL4:D13 ,F:0'  
 BS23 DC CL4:S23 ,F:0'

```

8023 DC CL4*D23 *F*0*
8021 DC CL4*S31 *F*0*
8022 DC CL4*D31 *F*0*
8032 DC CL4*S32 *F*0*
8033 DC CL4*D32 *F*0*
8035 DC CL4*S33 *F*0*
8033 DC CL4*D33 *F*0*
8014 DC CL4*S14 *F*0*
8014 DC CL4*D14 *F*0*
8024 DC CL4*S24 *F*0*
8024 DC CL4*D24 *F*0*
8034 DC CL4*S34 *F*0*
8034 DC CL4*D34 *F*0*
8015 DC CL4*S15 *F*0*
8015 DC CL4*D15 *F*0*
8025 DC CL4*S25 *F*0*
8025 DC CL4*D25 *F*0*
8035 DC CL4*S35 *F*0*
8035 DC CL4*D35 *F*0*
PARENTEZ DC CL5*( )
TRIPLEY DC CL5
SORTIE DS OF
CL8*TYPLIN*
DC ALI(1)
DC ALI(CHAIN)
DC C*K*
DC ALS(10)
DC OD
CL8*CONWAY*
DC CL4*CONI*
DC 800F*0*
DC CL48*X Y Z T U V W A B C D E
DC CL48*E G H I J K L M N O P Q
DS CL2000
GSECT
BALR R15*0
USING *R15
PG ALI*ADONNEES
PG USING ZONE*RII
L R3*MINF NBRE DE MONNIES DE LA BORNE INF
EXTRN DONNEES
ENTRY PG
A R3*RSUP NBRE DE MONNIES DU COMPL. DE LA BORNE SUP.
ST R3*M M=MINF+RSUP
SLA R3*2
ST R3*DEC4H DEC4H=M*4
L R3*N
SLA R3*2
ST R3*DEC4H DEC4H=4*N
S R3*F*4*
ST R3*DEC4NI DEC4NI=4*(N-1)
* FORMATION DES N PREMIERS CHAINES
LA R7*TAB
SR R2*R2
LA R8*PUISS
AR R5*R2
L R6*N
GLI OIRP)*C*
BE SUITE
DC O14*R7)*OIR8) INSERE UN I CORRP. AU MONNIE DS TAB

```

```

SUITE A R8*F*4*
A BCI R5*N MONNIE SUIVANT
A R2*F*1*
A R7*F*4*
C R2*N
BL VARSUIV
* FORMATION DES X,Y/X*Y/
L R5*DEC4N
ST R5*BORNA
L R1*ADRPRODI
L R15*ADRSPG
BALR R2*R1
L R5*N2
ST R5*BORNA
ST R5*BORNIINF+12
MVC BORNIINF(4)*ZERO
MVC BORNIINF(4)*DEC4N
L R10*AVARV2
L R1*ADRPROD2
L R15*ADRSPG
L R2*R1
BALR R2*R1
* CHOIX DE LA SORTIE S22 PARMI LES ((X Y) Z) ((X Y)
PSPSZL L R2*N2
PSPSZL L R1*DEC4N
BAL R4*ET
BAL R4*TESTI
C R5*NOMBREI
BNH PPSZ22
MVI SIGNE*C**
BAL R4*CHARGE3
C R3*BITUN
BE IMPRES
BAL R4*DU
BAL R4*TESTI
C R5*NOMBREI
C SUIPSI
MVI SIGNE*C**
BAL R4*CHARGE3
C R3*BITUN
BE IMPRES
A R1*F*4*
C R1*N2
BNE PSPSZ1
A R2*F*4*
C R2*N3
BNE PSPSZ1B
*****FORMATION DES ((X*Y) Z) ((X*Y) Z)
L R7*AVARVZ
LA R10*AMELIOR
CL1 R1R7)*C**
BNE CARREI
L R9*AVARVZ
CL1 R6R9)*C**
BNE CARRE2
CLC O11,R7)*OIR9)
BE CARRE5
CLC O11,R7)*2IR9)
BE CARRE4

```

CARRÉ4  
 CARRÉ5  
 CARRÉ6  
 CARRÉ7  
 CARRÉ8  
 CARRÉ9  
 CARRÉ10  
 CARRÉ11  
 CARRÉ12  
 CARRÉ13  
 CARRÉ14  
 CARRÉ15  
 CARRÉ16  
 CARRÉ17  
 CARRÉ18  
 CARRÉ19  
 CARRÉ20  
 CARRÉ21  
 CARRÉ22  
 CARRÉ23  
 CARRÉ24  
 CARRÉ25  
 CARRÉ26  
 CARRÉ27  
 CARRÉ28  
 CARRÉ29  
 CARRÉ30  
 CARRÉ31  
 CARRÉ32  
 CARRÉ33  
 CARRÉ34  
 CARRÉ35  
 CARRÉ36  
 CARRÉ37  
 CARRÉ38  
 CARRÉ39  
 CARRÉ40  
 CARRÉ41  
 CARRÉ42  
 CARRÉ43  
 CARRÉ44  
 CARRÉ45  
 CARRÉ46  
 CARRÉ47  
 CARRÉ48  
 CARRÉ49  
 CARRÉ50  
 CARRÉ51  
 CARRÉ52  
 CARRÉ53  
 CARRÉ54  
 CARRÉ55  
 CARRÉ56  
 CARRÉ57  
 CARRÉ58  
 CARRÉ59  
 CARRÉ60  
 CARRÉ61  
 CARRÉ62  
 CARRÉ63  
 CARRÉ64  
 CARRÉ65  
 CARRÉ66  
 CARRÉ67  
 CARRÉ68  
 CARRÉ69  
 CARRÉ70  
 CARRÉ71  
 CARRÉ72  
 CARRÉ73  
 CARRÉ74  
 CARRÉ75  
 CARRÉ76  
 CARRÉ77  
 CARRÉ78  
 CARRÉ79  
 CARRÉ80  
 CARRÉ81  
 CARRÉ82  
 CARRÉ83  
 CARRÉ84  
 CARRÉ85  
 CARRÉ86  
 CARRÉ87  
 CARRÉ88  
 CARRÉ89  
 CARRÉ90  
 CARRÉ91  
 CARRÉ92  
 CARRÉ93  
 CARRÉ94  
 CARRÉ95  
 CARRÉ96  
 CARRÉ97  
 CARRÉ98  
 CARRÉ99  
 CARRÉ100

CARRÉ7  
 IMPRES  
 \* IMPRESSION DES RESULTATS  
 NOCHA  
 NOUILLE  
 CARRÉ1  
 CARRÉ2  
 CARRÉ3  
 CARRÉ4  
 CARRÉ5  
 CARRÉ6  
 CARRÉ8  
 CARRÉ9  
 CARRÉ10  
 CARRÉ11  
 CARRÉ12  
 CARRÉ13  
 CARRÉ14  
 CARRÉ15  
 CARRÉ16  
 CARRÉ17  
 CARRÉ18  
 CARRÉ19  
 CARRÉ20  
 CARRÉ21  
 CARRÉ22  
 CARRÉ23  
 CARRÉ24  
 CARRÉ25  
 CARRÉ26  
 CARRÉ27  
 CARRÉ28  
 CARRÉ29  
 CARRÉ30  
 CARRÉ31  
 CARRÉ32  
 CARRÉ33  
 CARRÉ34  
 CARRÉ35  
 CARRÉ36  
 CARRÉ37  
 CARRÉ38  
 CARRÉ39  
 CARRÉ40  
 CARRÉ41  
 CARRÉ42  
 CARRÉ43  
 CARRÉ44  
 CARRÉ45  
 CARRÉ46  
 CARRÉ47  
 CARRÉ48  
 CARRÉ49  
 CARRÉ50  
 CARRÉ51  
 CARRÉ52  
 CARRÉ53  
 CARRÉ54  
 CARRÉ55  
 CARRÉ56  
 CARRÉ57  
 CARRÉ58  
 CARRÉ59  
 CARRÉ60  
 CARRÉ61  
 CARRÉ62  
 CARRÉ63  
 CARRÉ64  
 CARRÉ65  
 CARRÉ66  
 CARRÉ67  
 CARRÉ68  
 CARRÉ69  
 CARRÉ70  
 CARRÉ71  
 CARRÉ72  
 CARRÉ73  
 CARRÉ74  
 CARRÉ75  
 CARRÉ76  
 CARRÉ77  
 CARRÉ78  
 CARRÉ79  
 CARRÉ80  
 CARRÉ81  
 CARRÉ82  
 CARRÉ83  
 CARRÉ84  
 CARRÉ85  
 CARRÉ86  
 CARRÉ87  
 CARRÉ88  
 CARRÉ89  
 CARRÉ90  
 CARRÉ91  
 CARRÉ92  
 CARRÉ93  
 CARRÉ94  
 CARRÉ95  
 CARRÉ96  
 CARRÉ97  
 CARRÉ98  
 CARRÉ99  
 CARRÉ100

DANS R6 ADRESSE DE CHARGEMENT DU MONOHE

TEST S1 S12=X.Y.Z.

CAS (Y.Z) XI (Y.Z) II

CAS S22=(Y.Z) XI (A B)



```

R9,ADRCHER
LA R7,BD11+4
L R15,ADRSPG
BALR R8,R9
LA R7,BD12+4
CLI CHAINEB+11,C*()
BE DIZA
MVC TRIPLE1(3),BS11
MVC TRIPLE1(4(1)),SIJ
MVC TRIPLE1(3(1)),CHAINEB*5
R9,ADRCHX1
L R15,ADRSPG
L R8,R9
BALR R7,BD21+4
MVC TRIPLE1(3),CHAINEB+11
L R9,ADRCHERX
L R15,ADRSPG
BALR R8,R9
ELIMBIS3
MVC TRIPLE1(3),BD11 CAS ((X Y) 2) ((X Y) 1)
LA R7,BD21+4
L R9,ADRCHX1
L R15,ADRSPG
BALR R8,R9
LA R7,BD12+4
MVI TRIPLE1,C*5
L R9,ADRCHX1
L R15,ADRSPG
BALR R8,R9
ELIMINATION DES COLONNES INTERIEURES A S22
R1,BS22+4 CHARGE S22
L R2,ADRELIN
L R15,ADRSPG
BALR R3,R2
L R10,AVARN
MVC O(3,R10),SORTS22+7 S22 CARACBORNELINE+*5.
LA R5,TAB
A R5,DECAN
MVC O(4,R5),BS22+4 S22 BITS
MVC BITS22(4),BS22+4
L R5,DECANDIS
ST R5,BORNA
L R1,ADPRODI
L R15,ADRSPG
BALR R2,R1
LA R3,20
LA R5,TAB
A R5,N2
L R10,AVARV2
MVC O(4,R5),BS11+4
MVC O(3,R10),BS11
CLI CHAINEB+11,C*()
BNE ZUT
A R2,=F'4'
MVC 4(4,R5),BD11+4
MVC 4(3,R10),BD11
A R5,=F'4'
A R10,=F'4'
MVC 4(4,R5),BS12+4
MVC 4(3,R10),BS12

```

```

MVC R(4,R5),BD12+4
MVC R(3,R10),BD12
MVC 1(4,R5),BS21+4
MVC 1(3,R10),BS21
MVC 1(4,R5),BD21+4
MVC 1(3,R10),BD21
L R4,N2
AR R4,R3
ST R4,N3
A R4,AVAR
ST R4,AVARV2
***ECRITURE DES COLONNES RESTANTES
L R10,AVAR
SR R5,R5
L R8,CHAINEB DANS R8 ADRESSE DE CHARGEMENT DU MONOME
MVC 1(40,R8),BLANC
BAL 1(4,EBITA
A R10,=F'4'
A R5,=F'4'
C R9,N3
BL NOCHAI
BAL 1(4,ASPEVA
L R1,=F'1'
L R1,AMELIOR+4
MVI AMELIOR(2,C*() IMPRINS3
L R10,N3
MVC BORNA(4),N2
L R2,DECANDIS
MVC NCHERE(4),ZERO
L R4,ADRXIXJ
L R15,ADRSPG
BALR R3,R4
CLC PR(4),NCHERE1
BE IMPRINS3
L R1,N2 FORMATION DES SIJ S22
A R1,=F'4'
SR R2,R2
MVI SIGNE,C*()
BAL 1(4,EY
MVI SIGNES2(2,C*()
BAL 1(4,EIS22A
BAL 1(4,TESTA
BAL 1(4,COMPAR
MVI SIGNES22,C*()
BAL 1(4,OUS22A
BAL 1(4,TEST1
BAL 1(4,COMPAR
CLI SIGNE,C*()
BNE SIJX1
MVI SIGNE,C*()
BAL 1(4,OU
B SIJX3
A R2,=F'4'
C R2,DECAN
BL SIJX
C R5,M
DE FINP
A R1,=F'4'
C R1,N3
BL SIJX2

```

EBIT SR R2,R2  
 EBIT1 SRL R1,TAB(R5)  
 EBIT2 SLA R3,F=240  
 R1,F1 TEST DU (R2,F1)-EME BIT  
 BND EBIT2  
 A R3,F=1; SI CE BIT VAUT 1 ON AJOUTE UN  
 R3,CHAINED(R2)  
 A R2,F=1;  
 R2,M  
 C EBIT1  
 BNE I,SORTIE  
 LA X,CA;  
 SVC AL4(#+4)  
 DC I4  
 BR I,ATTENTE  
 LA X,CA;  
 SVC AL4(#+4)  
 DC I4  
 \* FORMATION DE X,Y  
 R3,TAB(R1)  
 R4,R3  
 R3,TAB(R2)  
 R4,TAB(R2)  
 R4,BITINF  
 R3,R4  
 I4  
 R5,NGMBREI  
 R6,N3  
 R3,TAB(R6)  
 R1,S22  
 R2,S22+2  
 S22+4(I),SIGNE  
 I4  
 R6,M  
 R5,R5  
 R7,R3  
 SRL R7,I  
 SLA R7,I  
 SUIEST  
 R5,F=1;  
 R6,RETEST  
 I4  
 R7,CHAINEM  
 I(I3,R7),SORTS22  
 I4,I5,R7,PARENT2  
 R9,S22  
 R9,N2  
 CASNUMI  
 R9,N2  
 R9,I2  
 RB,F=5;  
 R9,AVARV2

R7,CHAINEM  
 R5,AMELIOR  
 R5,AVAR  
 I(R7),C  
 Z(I,R7),O(R5)  
 SORTS22-I(I3),O(R5) CHARGE X13  
 R5,AMELIOR+4  
 R5,AVAR  
 ANELIOR+2,C  
 CASZ  
 SORTS22-I(I3),BLANC  
 S(I7),C  
 S(I7),AMELIOR+3  
 T(I7,R7),SORTS3  
 IMPRIM  
 S(I7,R7),AMELIOR+2 SIGNE DANS 2(I9)  
 S(I7,R7),O(R5)  
 T(I7),C  
 S(I7,R7),AMELIOR+3 SIGNES22 DANS 3(I9)  
 S(I7,R7),SORTS3  
 R5,N3  
 I4,EBIT  
 I4,ASPETA  
 M(I4),NGMBREI  
 FINPG  
 R15,ADPPI  
 R15  
 O  
 R5,NGMBREI  
 COMPARI  
 BL R6,R3  
 LR R6,BITS22  
 O R6,BITS22  
 C COMPARI  
 BE R5,NGMBREI  
 ST R1,AMELIOR  
 STH R2,AMELIOR+4  
 STH R10,N3  
 L R3,TAB(I10)  
 ST R4,B5I3+4 CHARGEMENT DE S13  
 ST AMELIOR+2(I),SIGNE  
 MVC AMELIOR+3(I),SIGNES22  
 MVC I4  
 BR R4,R3  
 LR R3,BITS22  
 N R5,R4  
 LR R5,BITS22  
 O R5,BITSUP  
 N R3,R5  
 OR I4  
 BR R3,R4  
 LR R3,BITS22  
 N R5,R4  
 LR R5,BITS22  
 O R5,BITINF A  
 N R3,R5  
 OR I4  
 BR I(I3,R8),O(R10) IMPRIME LES X

\* ECRITURE D'UNE CHAINE DE M BITS ET DU MCNDOME CORRESPONDANT  
 IMPRIME LES X



DC CL4\*023 \*F\*0\*  
 DC CL4\*S31 \*F\*0\*  
 DC CL4\*D31 \*F\*0\*  
 DC CL4\*S32 \*F\*0\*  
 DC CL4\*D32 \*F\*0\*  
 DC CL4\*S33 \*F\*0\*  
 DC CL4\*D33 \*F\*0\*  
 DC CL4\*S14 \*F\*0\*  
 DC CL4\*D14 \*F\*0\*  
 DC CL4\*S24 \*F\*0\*  
 DC CL4\*D24 \*F\*0\*  
 DC CL4\*S34 \*F\*0\*  
 DC CL4\*D34 \*F\*0\*  
 DC CL4\*S15 \*F\*0\*  
 DC CL4\*D15 \*F\*0\*  
 DC CL4\*S25 \*F\*0\*  
 DC CL4\*D25 \*F\*0\*  
 DC CL4\*S35 \*F\*0\*  
 DC CL4\*D35 \*F\*0\*  
 DC CL5\*0  
 DC CL5  
 DS OF  
 DC CL8\*TYPLIN\*  
 DC ALI(1)  
 DC AL3(CHAINES)  
 DC C\*K\*  
 DC AL3(10)  
 DS OD  
 DC CL8\*CONWAIT\*  
 DC CL4\*CONI\*  
 DC 800F\*0\*  
 DC CL48\*X Y Z T U V W A B C D E Q  
 DC CL48\*F G H I J K L M N O P Q  
 DC CL2000  
 CSECT  
 BALR R15\*0  
 USING \*R15  
 ENTRY PGI  
 EXTRA DONNEES  
 LA R1,PG1  
 ST R1,ADRPG  
 \*\*\*\*\*RANGEMENT DE S23 ET D22 ET S13  
 MVC SORTS22-9(9),BLANC  
 MVC TRIPLET(5),BLANC  
 L R1,TAB(R5)  
 L R1,BS23+4 CHARGEMENT DE S23  
 L R3,BS12+4  
 LR R4,R3  
 MVC BITS22(4),BS21+4  
 CLI CHAINES+9,C\*  
 BE D220U  
 BAL 14,ETS22  
 D220UA  
 BAL 14,DUS22  
 ST R3,SD22+4 CHARGEMENT DE D22  
 MVC CHAINES(8),I(8)\*  
 CLI CHAINES+4,C\*  
 BE SIBETIQ  
 CLI CHAINES+4,C\*+  
 BE SIBETIQ

LA R1,YAB  
 AH R1,AMELIOR  
 MVC BS13+4(14),O(1) CHARGEMENT DE S13  
 MVC TRIPLET(3),CHAINES+1  
 LA R7,BD13+4  
 L R9,ADRCH1  
 L R15,ADRSPG  
 BALR R8,R9  
 B ETIOS  
 B R7,SD13+4  
 LA TRIPLET(5),CHAINES+1  
 MVC R9,ADRCHX1  
 L R15,ADRSPG  
 L R8,R9  
 BALR SORTS22-9(3),CHAINES+1  
 MVC BITS22(4),BS22+4  
 MVC R3,BS13+4  
 L CLI AMELIOR+3,C\*  
 BE D23A  
 BAL 14,ETS22  
 B D23B  
 B 14,DUS22  
 ST R3,SD23+4  
 \*\*\*\*\*ELIMINATION DES COLONNES INFERIEURES A S22/S23  
 L R1,BS23+4  
 L R2,ADRELIM  
 L R15,ADRSPG  
 L BALR R3,R2  
 LA R4,YAB  
 A R4,DECAN  
 MVC O(4),R4),BS23+4  
 L R5,AVARN  
 MVC O(3),R5),BS23  
 L R5,DEC4NBIS  
 ST R5,SBORNA  
 L R1,ADRPRODI  
 L R15,ADRSPG  
 BALR R2,R1  
 L R10,AVAR  
 SR R5,R5  
 L R8,CHAINEM DANS R15 ADRESSE DE CHARGEMENT DU MONOME  
 MVC 1(40),R1),BLANC  
 BAL 14,EBIYA  
 A R10,=F\*4\*  
 A R5,=F\*4\*  
 C R5,N2  
 BL NOCHAI  
 \*REMISE A ZERO DES SORTIES 11 12 21 QUI NE PEUVENT PLUS  
 \* ETRE REUTILISEES  
 CLI CHAINES+4,C\*  
 BE D21A  
 CLI CHAINES+2,C\*1\*  
 BNE D11A  
 CLC CHAINES+1(3),BD11  
 BE D11B  
 CLC CHAINES+1(3),BS11  
 BE S11A  
 MVI BS11+3,C\*N\*  
 CLC CHAINES+1(3),BS12  
 BE S11A  
 CLC CHAINES+1(3),BS12  
 BE S12A

```

S12A MVI BS12+3,C*N*
      CLC CHAINEB+1(3),S012
      BE D21A
D11A MVI BD12+3,C*N*
      B D21A
      CLC CHAINEB+1(3),BD11
      BE D11B
D11B MVI BD11+3,C*N*
      CLC CHAINE+1(3),BS21
      BE S21A
S21A MVI BS21+3,C*N*
      CLC CHAINEB+1(3),BD21
      BE D21A
      MVI BD21+3,C*N*
      *****FORMATION DES (X Y) S3
D21A MVC BITS22(4),BS23+4
      MVC PARENT2(3),BS23
      MVC NOMBRE1(4),ZERO
      BAL 14,Z
      MVC BITS22(4),BD23+4
      MVC PARENT2(3),BD23
      BAL 14,Z
      *****FORMATION DES X(1S11),S12,D12,D11,S21,D21
L R5=F*8;
ST R5,PAS
L R5,NZ
ST R5,BORNA
MVC BORNINF(4),ZERO
MVC BORNINF+4(4),DEC4N
LA R2,TAB
LA R1,BS11+4
SR R1,R2
ST R1,BORNINF+8
LA R1,BD21+12
SR R1,R2
ST R1,BORNINF+12
L R10,AVARV2
L R1,ADRPRD2
L R15,ADRSPG
BAL R2,R1
      *****FORMATION DE S23 & ((XGV),XBS1J) & (S12,S21,S22,D22)
BAL 14,Z1
MVC BITS22(4),BS23+4
MVC PARENT2(3),BS23
BAL 14,Z1
      *****FORMATION DES S23 & (X & S13,D13)
BAL 14,Z2
MVC BITS22(4),BD23+4
MVC PARENT2(3),BD23
BAL 14,Z2
      MVI R5,N3
      MVI R4,EBIT
      CLC M(4),NOMBRE1
      BE FINPG1
      *****ELIMINATION DES COLONNES INFERIEURES A S24/S33
L R1,TABR5)
L R2,ADRELIM
L R15,ADRSPG
BAL R3,R2
LA R4,TAB
A R4,DEC4N

```

```

ST R1,DRP4)
L R5,AVARN
MVC R(3,R5),SORTS3+4
A R5,F*4;
ST R5,AVARV2
L R5,DEC4NBIS
ST R5,BORNA
CLC ZERO(4),DEC4N1
BL SUITES
ST R5,N2
L R1,ADRPRODI
L R15,ADRSPG
BAL R2,R1
      *****ECRITURE DES COLONNES RESTANTES
L R10,AVAR
SR R5,R5
L R8,CHAINEM
MVC R(40,R8),BLANC
A R10,F*4;
C R5,N2
BL MOCHA2
L R15,ADRPG2
BR R15
SVC Q
ADRPG2
*
R1 EQU 12
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R15 EQU 13
R9 EQU 9
R10 EQU 10
R11 EQU 11
R8 EQU 8
      *****SOUS-PROGRAMMES
Z ST 14,SAUVR
L R2,DEC4NBIS
L R10,N3
MVI AMELIDR+2,C*
MVC BORNA(4),N2
L R4,ADRXXJ
L R15,ADRSPG
BAL R3,R4
L R5,TAB(R10)
ST R5,SORTS22
CLC M(4),NOMBRE1
BE IMPRIMS4
L R4,SAUVR
BR 14
ST 14,SAUVR
MVC BORNINF(4),DEC4NBIS
MVC BORNA(4),N3
MVC S1(3),BS12
LA R9,N3
LA R2,BS12+4

```

LES REGISTRES SYMBOLIQUES

ADR. CHARG. DU RESULTAT

ETIQ9 LA R10,S12B  
 ETIQ8 L R1,BORNINF  
 ETIQ7 BAL MVI SIGNE,C\*  
 14,RY  
 BAL MVI SIGNES22,C\*  
 14,ETS22  
 BAL 14,TESTI  
 BAL 14,COMP  
 MVI SIGNES22,C\*  
 14,OUS22  
 BAL 14,TESTI  
 BAL 14,COMP  
 CLI SIGNE,C\*  
 BNE ETIQ6  
 MVI SIGNE,C\*  
 SAL 14,DU  
 B ETIQ7  
 A R1,F\*4  
 C R1,BORNA  
 BL ETIQ8  
 BR R10  
 MVC SIJ(3),BS21  
 LA R2,BS21+4  
 LA R10,S21B  
 B ETIQ9  
 MVC SIJ(3),BS22  
 LA R2,BS22+4  
 LA R10,S22B  
 B ETIQ9  
 MVC SIJ(3),BD22  
 LA R2,BD22+4  
 LA R10,D22B  
 B ETIQ9  
 CLC H(4),NOMBREI  
 BE IMPRIMS4  
 L 14,SAUVR  
 BR 14  
 ST 14,SAUVR  
 MVC BORNA(4),DEC4N  
 MVC BORNINF(4),ZERO  
 LA R2,BS13+4  
 MVC SIJ(3),BS13  
 LA R10,S13B  
 B ETIQ9  
 MVC SIJ(3),BD13  
 LA R2,BD13+4  
 LA R10,D13B  
 B ETIQ9  
 CLC SORTS22-9(3),BLANC  
 BE IMPRIMS4  
 LA R2,BS11 CAS OU S23=(SIJ & XI & S22  
 CLC SORTS22-9(3),O(R2)  
 BNE ETIQ13  
 MVC SIJ(3),O(R2)  
 A R2,F\*4  
 LA R10,IMPRIMS4  
 B ETIQ9  
 A R2,F\*8  
 B ETIQ14  
 L 14,SAUVR  
 BR 14

ETS22 LR R4,R3  
 N R3,BITS22  
 LR R5,R4  
 O R5,BITS22  
 N R5,BITSUP  
 OR R3,R5  
 BR 14  
 LR R3,R4  
 N R3,BITS22  
 LR R5,R4  
 O R5,BITS22  
 N R5,BITINF  
 OR R3,R5  
 BR 14  
 COMPAR C R5,NOMBREI  
 SNH COMPARI  
 L R6,BS23+4  
 OR R6,R3  
 C R6,BS23+4  
 BE COMPARI  
 ST R5,NOMBREI  
 STH R1,AMELIIOR  
 ST R3,TAB(R9) S33  
 ST R4,BITS S32  
 ST R2,AMELIIOR+8  
 MVC AMELIIOR+2(1),SIGNE  
 MVC AMELIIOR+3(1),SIGNES22  
 MVC TRIPLE(3),SIJ  
 MVC PARENT2+3(3),PARENT2  
 BR 14  
 COMPARI BR SORTS3+2,C\*3  
 IMPRIMS4 MVI R8,CHAINEX  
 L MVI I(R8),C\*4  
 L AH R7,AVAR  
 LA R7,AMELIIOR  
 LA R10,YAB  
 A R10,N3  
 LA R9,TAB  
 AH R9,AMELIIOR  
 CLI AMELIIOR+2,C\*  
 BNE ETIQ10  
 MVI SORTS3+6,C\*4 CAS (X Y) & S23 =S24  
 MVI 5(R8),C\*4  
 MVC 2(3),R8,O(R7)  
 MVC SORTS3(3),PARENT2+3  
 MVC 7(1),R8,SORTS3  
 MVC 6(1),R8,AMELIIOR+3  
 MVC BS24+4(4),SORTS22  
 MVC BS14+4(4),O(R9)  
 MVC SORTS22-9(3),BLANC  
 MVC TRIPLE(3),O(R7)  
 LA R7,BD14+4  
 L R9,ADRCHI  
 L R8,R9  
 L R15,ADRSPG  
 B BALR  
 ETIQ10 B ETIQ17  
 CLI TRIPLE+2,C\*3  
 BNE ETIQ11  
 MVI SORTS3+6,C\*4  
 MVC 6(1),R8,O(R7) CAS (X S13/D13) S23 = S24  
 MVC 5(1),R8,AMELIIOR+2

Label	Code	Instruction	Comment
MVC	2(3,R8),	TRIPLET	
MVI	7(R8),	C'1'	
MVC	8(1,R8),	AMELIOR+3	
MVC	SORTS3(3),	PARENT2+3	
MVC	9(7,R8),	SORTS3	
MVC	BS14+4(4),	BITS3	
MVC	BS24+4(4),	DIR10	
MVC	SORTS22-9(3),	TRIPLET	CHARGE X14
MVC	TRIPLET(5),	2(R8)	
LA	R7,	BD14+4	
L	R9,	ADRCHX1	
L	R15,	ADRSPG	
L	R8,	R9	
L	R3,	BS14+4	
LR	R4,	R3	
MVC	BITS22(4),	BS23+4	
CLI	AMELIOR+3,	C'1'	
BE	ETIQ18		
BAL	14,	ETS22	
B	ETIQ19		
BAL	14,	OUS22	
ST	R3,	BD24+4	
LA	R5,	SORTS22	
LA	R3,	TAB	
SR	R5,	R3	
B	IMPRIM1		
MVI	SORTS3+5,	C'3'	
MVC	BS33+4(4),	O(R10)	
MVC	BS22+4(4),	BITS3	
MVC	BS31+4(4),	O(R9)	
MVC	SORTS22-3(3),	TRIPLET	CHARGE Y32
L	R3,	BS23+4	
LR	R4,	R3	
MVC	BITS22(4),	BS23+4	
CLI	AMELIOR+3,	C'1'	
BE	ETIQ15		
BAL	14,	ETS22	
B	ETIQ16		
BAL	14,	OUS22	
ST	R3,	BD33+4	
LH	R1,	AMELIOR	
C	R1,	N2	
BNE	ETIQ12		
MVI	2(R8),	C'1'	CAS (X Y) & S12/S21/S22/D22) & S23=S33
MVC	3(2,R8),	O(R7)	
MVI	6(R8),	C'1'	
MVC	8(3,R8),	AMELIOR+2	
MVI	11(R8),	TRIPLET	
MVC	12(1,R8),	C'1'	
MVC	12(1,R8),	AMELIOR+3	
MVC	13(7,R8),	SORTS3	
MVC	TRIPLET(3),	O(R7)	
LA	R7,	BD31+4	CALCUL DE D31
L	R9,	ADRCHI	
L	R15,	ADRSPG	
RALP	R8,	R9	
B	K		
LH	R7,	AMELIOR	
S	R7,	N2	
SRA	R7,	2	
M	R6,	=F'15'	
MVC	2(R8),	C'1'	
MVI	3(5,R8),	O(R7)	
MVC	SORTS22-6(3),	O(R7)	CHARGE Y31
MVC	10(1,R8),	C'1'	
MVC	11(1,R8),	AMELIOR+2	
MVC	12(3,R8),	TRIPLET	
MVI	15(R8),	C'1'	
MVC	16(1,R8),	AMELIOR+3	
MVC	SORTS3(3),	PARENT2+3	
MVC	17(7,R8),	SORTS3	
MVC	TRIPLET(5),	O(R7)	
LA	R7,	BD31+4	CALCUL DE D31
L	R9,	ADRCHX1	
L	R15,	ADRSPG	
L	R8,	R9	
MVC	BITS22(4),	BS31+4	CALCUL DE BD32
L	R3,	AMELIOR+8	
CLI	AMELIOR+3,	C'1'	
BE	X1		
BAL	14,	ETS22	
B	K2		
BAL	14,	OUS22	
ST	R3,	BD32+4	
B	IMPRIM		
MVC	1(3,R8),	O(R10)	
SR	R2,	R2	
L	R1,	TAB(R5)	
SRL	R1,	1	
L	R3,	=F'240'	
SLA	R1,	1	TEST DU (R2+1)-EME BIT
BNO	EBIT2		
A	R3,	=F'1'	SI CE BIT VAUT 1 ON AJOUTE UN
STC	R3,	CHAINE(R2)	
A	R2,	=F'1'	
C	R2,	N	
BNE	EDIT1		
LA	1,	SORTIE	
SVC	X'CA'		
DC	AL4(#+4)		
LA	1,	AITENTE	
SVC	X'CA'		
DC	AL4(#+4)		
BR	14		
* FORMATION DE X.Y	R4,	O(R2)	
ET	R4,	TAB(R1)	
L	R3,	O(R2)	
L	R3,	TAB(R1)	
N	R3,	BITSUP	
OR	R3,	R4	
BR	14		
* FORMATION DE X+Y	R4,	O(R2)	
OU	R4,	TAB(R1)	
L	R3,	O(R2)	
L	R3,	TAB(R1)	
O	R3,	TAB(R1)	
N	R3,	BITINF	
OR	R3,	R4	
BR	14		
TEST1	L		

SR R5,R5  
 LR R7,R3  
 SRL R7,I  
 SLA R7,I  
 BND SUITE5  
 A R5,R7,I  
 BCT R6,RETEST  
 BR 14  
 END PGIA

\*\*\*\*\*  
 \*\*\*\*\*

11/19/70 09.21.44

CPATUSERID MONTAGNO  
 ZONE DSECT F\*0\*

NCMBRE1	DC	2F	
N	DC	F*6*	
MINF	DC	F*8*	
MSUP	DC	F*9*	
DEC4N	DS	F	
DEC4NBIS	DS	F	
DEC4NI	DS	F	
N2	DC	F*10000*	
N3	DS	F	
BORNA	DS	F	
AVAR	DC	A(VAR)	
ZERO	DC	F*0*	
AVARN	DS	F	
M	DS	F	
AVARV2	DS	F	
AVARV3	DS	F	
AMELIDR2	DS	F	
SAUVR	DS	F	
BORNIINF	DS	4F	
ADRPRODL	DC	V(PRODUIT1)	
ADRPRODZ	DC	V(PRODUIT2)	
ADRSPG	DC	V(SPG)	
ADRELIM	DC	V(ELIMINI)	
ADRCHER	DC	V(CHERSIJ)	
ADRCHERX	DC	V(CHERSIJX)	
ADRCHI	DC	V(CHERSI)	
ADRCHXI	DC	V(CHERSKI)	
ADRCHXJ	DC	V(XIXJS22)	
ADRP	DC	V(PG)	
ACRPG1	DC	V(PG1)	
BIYS3	DS	F	
BITIS22	DS	F	
BITINF	DC	XL4*FF00000*	
BIYSUP	DC	XL4*0GFF800*	
BITUN	DC	XL4*FFFF800*	
AMELIDR	DS	100F	
SORTIS22	DC	CL13*SORTIE S22.*	
SORTIS3	DC	CL7*S22=S23*	
CHaine	DC	CL70*	
CHAINED	DS	CL20	
CHAINEM	DS	F	
BLANC	DC	CL40*	
PAREN2	DC	CL15*(( ) )	
SIGNE	DS	CL1	
SIGNES22	DS	CL1	
SIF	DS	CL3	
PAS	DC	F*4*	
BS11	DC	CL4*S11*,F*0*	
BD11	DC	CL4*D11*,F*0*	
BS12	DC	CL4*S12*,F*0*	
BD12	DC	CL4*D12*,F*0*	
BS21	DC	CL4*S21*,F*0*	
BD21	DC	CL4*D21*,F*0*	
BS22	DC	CL4*S22*,F*0*	
BD22	DC	CL4*D22*,F*0*	
BS13	DC	CL4*S13*,F*0*	
BD13	DC	CL4*D13*,F*0*	
BS23	DC	CL4*S23*,F*0*	
BD23	DC	CL4*D23*,F*0*	



DC CL4'D23 'F'0,31  
 DC CL4'S31 'F'0  
 DC CL4'D31 'F'0  
 DC CL4'S32 'F'0  
 DC CL4'D32 'F'0  
 DC CL4'S33 'F'0  
 DC CL4'D33 'F'0  
 DC CL4'S14 'F'0  
 DC CL4'D14 'F'0  
 DC CL4'S24 'F'0  
 DC CL4'D24 'F'0  
 DC CL4'S34 'F'0  
 DC CL4'D34 'F'0  
 DC CL4'S15 'F'0  
 DC CL4'D15 'F'0  
 DC CL4'S25 'F'0  
 DC CL4'D25 'F'0  
 DC CL4'S35 'F'0  
 DC CL4'D35 'F'0  
 DC CL5'4 'F'  
 DC OF  
 DC CL4'TYPLIN'  
 DC ALI111  
 DC AL31CHAYNE1  
 DC C'R'  
 DC AL31701  
 DC CD  
 DC CL8'CONWAIT'  
 DC CL4'CONI'  
 DC 809F'0  
 DC CL48'X Y Z T U V W A B C D P Q  
 DC CL48'F G H I J K L M N O  
 DC CL2C00  
 CSECT  
 BAR R15,0  
 USING #R15  
 USING ZONE,R11  
 ENTRY YC2  
 EXTRN DOWNEES  
 LA R1,PG2  
 ST R1,ADPPG  
 MVC NOMBRE1(4),ZERO  
 MVI SIGNES22C'  
 CLI SORT53'5,C'3'  
 BE CAS33  
 \*\*INFORMATION DES X & (S14 D14 X14)  
 LA R10,LABELL  
 LA R2,AMELIDR+24  
 MVC AMELIDR(16),BS34  
 LA R1,BS11  
 LA R1,BS13  
 MVC AMELIDR+16(3),SORT522-9  
 CLI AMELIDR+16,C'  
 BE ETI01  
 CLC AMELIDR+16(3),OPR11  
 SUE ETI02 A  
 MVC AMELIDR+20(4),4PR11  
 B ETI04  
 A R1,'F'0  
 CR R1,'F3

ETI01  
 ETI04  
 BL ETRQ3  
 S RZ,'F'0'  
 LA R1,TAB  
 SR R2,RI  
 ST R2,BORNINF+12  
 ST R10,SAUVR  
 LA R2,AMELIDR+4  
 SR R2,RI  
 ST R2,BORNINF+8  
 MVC BORNINF(4),ZERO  
 L R5,N2  
 ST R5,BORWA  
 L R10,AVARV2  
 L R1,ADRPROD2  
 L R15,ADRSPG  
 L R2,RI  
 BALR R10,SAUVR  
 BR R10  
 \*\*INFORMATION DES S24 & (IX Y), X & (S14,DI4,XI4) I  
 LABELL MVC BITS22(4),BS24+4  
 L R2,DEC4NBIS  
 L R10,N3  
 MVI MEMOIR+18,C'  
 ST R10,9ORNA  
 L R4,ADR'IXJ  
 L R15,ADRSPG  
 BALR R3,R4  
 MVC MEMOIR+21(1),AMELIDR+3  
 L R6,YABIR10  
 ST R6,SORT522  
 LH R5,AMELIDR  
 STH R5,MEMOIR+16  
 C R5,N2  
 BL SUITE  
 S R5,N2  
 SRA R5+2  
 M R4,'F'5'  
 A R5,AVARV2  
 MVC MEMOIR+23(5),O(R5)  
 B SUIT  
 A R5,AVAR  
 MVC MEMOIR+23(7),O(R5)  
 CLC N(4),NOMBRE1  
 BE IMPRIMS5  
 \*\*INFORMATION DES X(S11,S12,D12,D11,S21,D21)  
 \*\*INFORMATION AU MOIS CEUX PERNIS (R,AYARI PAS 'H'1)  
 L R5,N2  
 ST R5,5ORNA  
 LA R2,TAB  
 MVC BORNINF(4),ZERO  
 MVC BORNINF+4(4),DEC4N  
 LA R1,BS11+4  
 SR R1,R2  
 ST R1,BORNINF+8  
 LA R1,BS21+12  
 SR R1,R2  
 ST R1,BORNINF+12  
 L R10,AVARV2  
 L R1,ADRPROD2

BIBLIOGRAPHIE

- ACKERS S.B. \* A diagrammatic approach to multilevel logic, synthesis.  
IT n° 2 1965.
- \* Synthesis of combinational logic using 3-input majority gates.  
C.R. sept./ oct. 1964.

AMAREL - BRZOWSKI J.A.

Theoretical considerations on reliability properties of recursive triangular switching networks.  
Redundancy techniques for computing systems.  
Spartan Books Washington DC 1962.

ANDREOLI G.

Formazioni algebriche booleane monotone.  
La Ricerca Janvier/avril 1961.

ARNOLD T.F. - CHUNG JEN TAN - NEWBORN N.M.

Iteratively realized sequential circuits  
IT Janvier 1970.

BASU M.S. - DAS S.R. - CHOUDHURY A.K.

A chart method for the determination of one of the minimal forms of a switching function.  
J. Electron. Control. G.B. 1964 n° 4.

On a method of simplification of multiple output switching functions.  
J. Electron. Control. vol. 16 1964.

BEAN A.

Réseaux microcellulaires.  
Rapport interne CII/IMAG février 1969.

BENZAKEN C.

Algorithmes de dualisation d'une fonction booléenne.  
RFTI Chiffres vol. 9 2ème trimestre 1966.

Contribution des structures algébriques ordonnées à la théorie des réseaux.  
Thèse Grenoble 1968.

BERGE

Théorie des graphes.  
Dunod.

BIVAL L.G.

Réalisation de fonctions booléennes par un simple élément somme modulo 2.  
A. and R.C. Juin 68

BOWMAN R.M. - Mc VEY E.S.

A method for the fast approximate solution of large prime implicant charts.  
IT février 1970.

BREUER M.A.

Leuristic switching expression simplification.  
Proc. ACM 23 rd. Nat. Conf. Août 1968.

CALDWELL S.H.

Switching circuits and logical design.  
New York Wiley 1958.

CANADAY R.H.

Two dimensional iterative logic.  
- MIT project DSR 9891 septembre 1964.  
- Proc. AFIPS 1965 Vol. 27 pp. 343-353.

CHAKRABARTI K.K. - CHOUDHURY A.K. - BASU M.S.

Complementary function approach to the synthesis of three-level NAND network.  
IT juin 1970.

CHEIN M.

Etude des décompositions d'un réseau.  
Application à l'écriture des fonctions booléennes en sommes et produits.  
Thèse Grenoble octobre 1967.

CHOUDHURY A.K. - DAS S.R.

Direct determination of all the minimal prime implicant covers of a switching function.

J. Electron. Control G.B. n° 5 1964.

CHOUDHURY A.K. - DAS S.R. - ROY K.K.

Simplification of switching functions involving a very large number of "don't care" states.

Intern. J. Control. G.B. n° 1, 1966.

CHU J.T.

Some methods for simplifying switching circuits using don't care conditions.

Journal of ACM octobre 1961.

CIOFFI G. FALZONE V.

Un nuovo metodo per la ricerca degli implicanti primi.

AF. janvier 1969. p. 21.

Mc CLUSKEY E.J.

Logical design theory of NOR gate networks with no complemented inputs.

IEEE Symposium Octobre 1963.

Minimal sums for boole functions having many unspecified fundamental products

Symposium Detroit juin 1962.

Introduction to the theory of switching circuits

Mc Graw Hill Book Company 1965.

Mc CLUSKEY E.J.

Essential multiple output prime implicants.

Math. Theory of Automata New York 1963.

DESCHIZEAUX P.

Synthèse de fonctions booléennes générales.

Thèse Grenoble 1967.

DIETMEYER D.L. - SCHNEIDER P.R.

A computed oriented factoring algorithm for NOR logic design.  
IT decembre 1965.

An alg. for synthesis of logical combinational multiple output function.  
IT 1968 p. 117.

DIETMEYER D.L. YUEH HSUNG SU

Logic design automation of fan-In limited NAND networks.  
IT janvier 1969.

DVORAK V.

A two nail cascade synthesis of boolean functions.  
IT juin 1968.

ELLIS D.T.

A synthesis of combinational logic with NAND or NOR elements.  
IT octobre 1965.

ELSPAS B.

Topological constraints on interconnection limited logic.  
Symposium on switching circuit theory and logical design.  
Octobre 1964. IEEE publication.

ELSPAS B. - KAUTZ W.H. - TURNER J.

Theory of cellular logic networks and machines.  
Final Report Stanford Institute.  
Mels. Park Calif. 1968.

FANTAUZZI G.

A semi-group theory for the Maitra cascade  
IFIP Congress Août 1968.

FERRARI

Fast carry propagation iterative networks.  
IT 1968 p. 136.

FLORINE J.

Automatismes a sequences et commandes numériques  
DUNOD 1969.

FRISIANI A.L.

Applicazione della teoria della composizione alla semplificazione delle espressioni booleane.  
AF. 1967 p. 910.

GALIMBERTI R. - GRASSELLI A. - MORPURGOR

On the number of core coverable switching functions calculs.  
avril-juin 1965.

GANSSEN F.

Möglichkeiten einer minimalen zweistufigen synthese Zeitunabhängiger Schaltfunktions-systeme.  
Elektr. Rechenanlage decembre 1966.

GILLI L. MEO A.R.

Algorithmi per la minimizatione di funzioni logiche.  
R.C. LXVII Riunione AEI ottobre 1966.

Elaboratore speciale per la sintesi di circuiti logici combinatori.  
idem.

methodi topologici per il calcolo di valori massimi concernenti la sintesi dei circuiti logici.  
idem.

GIMPEL J.F.

A reduction technic for prime implicant table.  
IT Août 1965.

The minimization of TANT networks.  
IT février 1967.

COLOMB S.W.

On the classification of boolean functions.  
IRE Trans. mai 1959.

GORDON B.B. - HOUSE R.W. - LECHLER A.P. - NELSON L.D. - RADO T.

Simplification of the covering problems for multiple output logical networks.  
IT decembre 1966.

GRASSELLI A.

Un procedimento per la sintesi di rete logiche combinatorie.  
AF. n° 10 octobre 1962.

An approximate method for the synthesis of multiple output combi-  
national networks.  
Calcols tome 2 1965.

HAMMER P.L. - RUDEANU S.

Boolean method in operations research.  
Springer Verlag Berlin - Heidelberg New York 1968.

HANSEL G.

Construction d'un schéma de contacts bipolaires pour une fonction  
booléenne isotone arbitraire de n variables.  
CRAS 263 - 1966 pp. 651-654.

HARRISSON M.A.

Combinatorial problems in bool. algebras and application to the  
theory of switching.  
1963 - Thèse University of Michigan.

HENNIE F.C.

Iterative networks with memory.  
Res. lab. of Electronics MIT Cambridge nov. 1959.

Iterative arrays of logical circuits.  
MIT Press Cambridge 1961.

Two iterative networks and their decision problems.  
Gen. Electr. Res.Lab. Rep. Schenectady New York 1960.

Finite-state models for logical machines.  
John Wiley 1968.

HOCKNEY R.

An intersection alg. giving all irredundant normal forms from a P.  
implicant list.  
IRE Trans. Avril 1962. Corresp.

HOUSE R.W. - RADO T.

A generalization of Nelson's alg. for obtaining P. implicants.  
J. Synt. logic. n° 1 1965.

On a computer program for obtaining irreducible representations for  
two-level multiple input-output logical systems.  
J. ass. Comptg. Mach. vol. 10 janvier 1963.

HURST S.L.

Establishment of irredundant P. Impli. in Bool. minimiz. techniques.  
Electronic Letters G.B. 1966 n° 2, n° 8.

IGNATUSHCHENKO

Synthèse de fonctions de commutation dans certains types de struc-  
tures homogènes.  
A and RC mai 1968.

KARP

Functional decomposition and switch. circuit design.  
J. SIAM juin 1963.

KAUTZ W.H.

Digital design empolying large arrays.  
Stanford Research Institute Menlo Park California.

Testing and diagnosis of cellular arrays.  
Symposium on switch. Theory and automata theory 1967.

KAUTZ W.H. - TURNER J.

Universal connecting networks and the synthesis fo canonical sequen-  
tial circuits.  
Symp. on switch. Theory and automata theory 1968.

KAUTZ W.H. - LEWITT K.N. - WAKSMAN A.

Cellular interconnection arrays.  
IT n° 5 1968.

KAZAKOV V.D.

The minimization of logical function of a large number of variables.  
A and RC n° 9 septembre 1962.



KILMER W.

Iterative switching networks composed of combinational cello.  
IRE Trans. 1962.

Topics in the theory of one-dimensional iterative networks.  
Information and Control 7, 1964.

KRIEGER M.

Basic switching circuit theory.  
New York Mac Millan 1967.

KUNTZMANN J.

Algèbre de Boole.  
2ème Edition - Dunod 1968.

LAPSCHER F.

Application de la notion de fermeture à l'étude des fonctions booléennes.  
Thèse Grenoble, septembre 1968.

Nombre moyen de majorants d'une fonction booléenne incomplète. Remarques  
concernant le nombre moyen de monômes premiers.  
Calculs janvier 1967.

LAWLER E.L.

The minimal synthesis of tree structures.  
IEEE Symposium septembre 1963.

Minimal boolean expression with more than two levels of sums and products.  
Symp. on Switch. Th. and logical design, 1962.

LECHNER R.J.

A transform approach to logical design.  
IT juillet 1970.

LEDLEY R.S.

Digital computer and control engineering.  
Mc Graw Hill Book new-York 1960.

LEVIEN R.E.

The synthesis of cascade switch. circuits.  
AIEEE Symposium octobre 1962.

On determining the best ordering of variables in cascade switch.  
circuits.  
IEEE Symposium octobre 1963.

LISS D.

A test for unate truth funct.  
IT n° 4 1963.

LUCCIO F.

A method for the selection of P. implicant.  
IT avril 1966.

LUSTMAN F.

Réalisation de fonctions booléennes avec l'opérateur Majorité.  
Thèse Grenoble 1966.

MACHERAS J.

Synthèse d'une fonction booléenne par l'opérateur U.  
Thèse Grenoble 1966.

MAGE J.M.

Application of iterative consensus to multiple output functions.  
IT avril 1970. corresd.

MAITRA K.K.

Cascaded switching networks of two-input flexible cells.  
IRE Trans. Elect. Corresp. avril 1962.

MEO A.R.

Sul problema della di una funzione logica.  
AF n° 10 1963.

Un procedimento di semplificazione delle funzione logiche.  
AF n° 7 1962.

MEO A.R.

Un procedimento rapido per la riduzione di una rete logica a tre livelli.

AF. n° 4 1963.

On the determination of the ps maximal implicants of a swtiching functions.

IT n° 6 1965.

Alcune importanti proprieta delle funzioni booleane monotone.  
Ann. Accad. Sc. Torins 1963/64.

Sulla sintesi di reti NAND o NOR a molti livelli.  
Calcolo tome 2 janvier 1965.

Structures d'arbres modulaires  
IT 1968 p. 432.

MERMET J.

Synthèse de fonctions booléennes par des opérateurs en ligne.  
Bull. Inst. Politech. Iasy 1967 pp. 31-40.

MILETO F. PUTZOLU G.

Average values of quantities appearing in boolean functions minimization.

IT avril 1964/ août 1965.

Statistical complexity of algorithms for boolean functions minimization.

J. ACM 1965 n° 3.

MILLER R.E.

Combinational circuits.

New York - Wiley 1965.

MINNICK - SHORT - STONE - GOLDBERG - GREEN - YOELI - KAUTZ

Cellular arrays for logic and storage.

Stanford Res. Instit. Menlo Park. California.

Rapports de 64 et 66.

MINNICK R.C.

A survey of micro-cellular research.  
J. ACM avril 1967.

Outpoint cellular logic.  
IT decembre 1964.

Cobwed cellular arrays.  
AFIPS 1965.

MORREALE E.

Une méthode pour la détermination des implicants premiers à partir de formes canoniques totalement ordonnées.  
Colloque Grenoble 1965.

Recursive operators for prime implicant and irredundant normal form determination.  
Juin 1970.

MOTT T.H.

An algorithm for determining minimal forms of an incomplete truth functions.  
AIEE Chicago octobre 1959.

MOTT - CAROLL C.C.

Numerical procedures for boolean functions minimization.  
IT Août 1964.

MUKHOPADHAY A.

Unate cellular logic.  
IT février 1969.

Minimization of Exclusive-OR and logical equivalence switching circuits.  
IT février 1970.

NATAPOFF A.

Irreductible topological components of an arbitrary bool. truth function and generation of their minimal coverings.  
J. ACM avril 1967.

Mc NAUGHTON R.

Unate truth functions.  
IRE Trans. 1961.

NECULA N.N.

A numerical procedure for determination of the prime implicants of a boolean functions.  
IT octobre 1967.

An algorithme for the automatic approximate minimization of boolean functions.  
IT Août 1968.

OZERNOY

Minimization of combinational logic circuits constructed from uni-fonctional elements.  
Engineering Cybernetics p. 23.

PICHAT E.

Décomposition d'une fonction booléenne.  
Thèse janvier 1966.

PINTER C.

Sur l'obtention de formes minimales de fonctions booléennes conjonctives.  
Thèse Paris 1964.

POLANSKY R.B.

Minimization of multiple output switching circuits.  
AIEE mars 1961.

POLOSUKHIN

Algorithme pour la synthèse de circuits combinatoires utilisant des structures homogènes.  
A and RC juin 1968.

PRANGISHVILI I.V.

Structures planaires et itératives et graphes correspondants.  
A and RC mai 1968.

PRANGISHVILI - BABICHEVA - IGNATUSHCHENKO

New principles for the realization of logical and computing devices of the basis of homogeneous microelectronic structures.  
Avtomatika i telemekhanika vol. 26 n° 10 octobre 1965.

PRATHER R.E.

On the tree circuits.  
IT 1965 p. 841.

ROTH J.P.

Minimization over boolean graphs.  
IBM J. Res. 1962.

SAILLARD J.C.

Etude des formes lexicographiques des fonctions booléennes simples.  
Représentation à l'aide de l'opérateur U.  
Thèse Grenoble décembre 1968.

SHANNON C.

The synthesis of two-terminal switching circuit.  
Bell sys. tech. jour. janvier 1949.

SHEINMAN A.H.

Numerical graphical method for multiterminal switching circuits.  
Trans. AIEE janvier 58/novembre 59.

SHORT R.A.

A theory of relations between sequential and combinatorial realization of switching functions.  
Stanford Electr. Lab. Tech. Report décembre 1960.

Two-rail cellular cascades.  
AFIPS Proc. vol. 27 1965.

SKLANSKY J.

General synthesis of tributary switching networks.  
IT octobre 1963.

SKLANSKY - KORENJAK - STONE

Canonical tributary networks.  
IT décembre 1965.

SLAGLE - CHIN LIANG CHANG - RICHARD LEE

A new algorithm for generating prime implicants.  
IT avril 1970.

SMITH R.A.

Minimal three-variable NOR and NAND logic circuits.  
IT février 1965.

SPANDORFER - MURPHY

Synthesis of logic functions on an array of integrated circuits.  
Rapport contract. AF 19 (628) 2907 octobre 1963.

SPANDORFER - TONIK

Planar interconnected logic.  
Proc. of Symposium on microelectronics and large systems.  
Spartan books Washington 1965.

STONE H.S.

An algorithm for modular partitioning.  
Journ. of ACM janvier 1970.

On the number of equivalence classes of functions realizable by  
cellular cascades.  
1965 - Proc. of Nat. Symp. on the impact of batch fabricat. on future  
computers pp. 81-87.

STONE - KORENJAK

Canonical form in synthesis of cellular arrays.  
IT décembre 1965.

SVOBODA A.

Ordering of prime implicants.  
IT 1967 p. 18.

SZE - TSEN HU

Mathematical theory of switching circuits and automata.  
University of California Press, Berkeley 1968.

TISON

Théorie des consensus.  
Thèse Grenoble 1965.

URBANO

Boolean matrices and stability of neural nets.  
IT 1963 pp. 61-66.

Structures et fonctions dans les réseaux polyfonctionnels.  
IT 1968 p. 152.

VANDLING G.C.

The simplification of multiple output switching networks composed  
of unilateral devices.  
IRE Trans. vol. 9 1960 n° 4.

VINCENT CARREFOUR

Optimisation d'un schéma en logique modulaire.  
Chiffres 3 1964 pp. 217-250.

WAKSMAN A.

A permutation network.  
J. ACM n° 1 1968.

WARLITZ G.

Über eine Schaltungs algebra für NOR und NAND.  
Elektron. Datenverarbeitung 1967 pp. 59-65.

WEISS C.D.

Optimal synthesis of arbitrary switching functions with regular  
arrays of two-input one-output switching elements.  
IT septembre 1969.

A characterization and properties of cascade realizable switching  
functions.  
IT juillet 1969.



YAU - ORSIC

Fault diagnosis and repair of outpoint cellular logic.  
IT mars 1970.

YOELI

A group theoretical approach to two-rail cascades.  
IT décembre 1965.

Generalized Maitra cascades.  
AFIPS conf. proc. Washington Spartan 1965 p. 815.

Cascades de cellules ternaires.  
IT 1968 p. 66.

YUN SHEN SHEN - Mc KELLAR

An algorithm for the disjunctive decomposition of switching functions.  
IT mars 1970.

ABREVIATIONS

IT : IEEE Trans. on Elec. Comp.

CR : Computing Review

A. and R.C.: Automation and remote control

AF : Alta Frequenza

VU

Grenoble, le

Le Président de la Thèse

VU

Grenoble, le

Le Doyen de la Faculté des Sciences

VU, et permis d'imprimer

Le Recteur de l'Académie de GRENOBLE

