Étude des formes lexicographiques des fonctions booléennes simples, représentation à l'aide de l'opérateur U

Jean-Claude Saillard

N° d'ordre

T H E S E

Présentée à la Faculté des Sciences
de l'Université de GRENOBLE

*

pour obtenir

le grade de Docteur de Troisième Cycle
"MATHEMATIQUES APPLIQUEES"

*

par

JEAN CLAUDE SAILLARD

*

ETUDE DES FORMES LEXICOGRAPHIQUES DES FONCTIONS BOOLEENNES SIMPLES - REPRESENTATION A L'AIDE DE L'OPERATEUR U

* * *

Thèse soutenue le 23 Décembre 1968, devant la Commission d'examen :

Monsieur	J. KUNTZMANN	Président
Messieurs	B. VAUQUOIS	Examinateur
	C. BENZAKEN	Examinateur

N° d'ordre

T H E S E

Présentée à la Faculté des Sciences
de l'Université de GRENOBLE

*

pour obtenir

le grade de Docteur de Troisième Cycle
"MATHEMATIQUES APPLIQUEES"

*

par

JEAN CLAUDE SAILLARD

*

ETUDE DES FORMES LEXICOGRAPHIQUES DES FONCTIONS BOOLEENNES SIMPLES - REPRESENTATION A L'AIDE DE L'OPERATEUR U

* * *

Thèse soutenue le 23 Décembre 1968, devant la Commission d'examen :

Monsieur	J. KUNTZMANN	Président
Messieurs	B. VAUQUOIS	Examinateur
	C. BENZAKEN	Examinateur

FACULTE DES SCIENCES — UNIVERSITE DE GRENOBLE

L I S T E des P R O F E S S E U R S

DOYEN HONORAIRE : M. MORET

DOYEN : M. BONNIER

PROFESSEURS TITULAIRES

MM.	NEEL Louis	Chaire de Physique Exéprimentale
	HEILMANN René	Chaire de Chimie
	KRAVTCHENKO Julien	Chaire de Mécanique Rationnelle
	CHABAUTY Claude	Chaire de calcul différentiel et Intégral
	BENOIT Jean	Chaire de Radioélectricité
	CHENE Marcel	Chaire de Chimie Papetière
	FELICI Noël	Chaire d'Electrostatique
	KUNTZMANN Jean	Chaire de Mathématiques Appliquées
	BARBIER Reynold	Chaire de Géologie Appliquée
	SANTON Lucien	Chaire de Mécanique des Fluides
	OZENDA Paul	Chaire de Botanique
	FALLOT Maurice	Chaire de Physique Industrielle
	KOSZUL Jean Louis	Chaire de Mathématiques
	GALVANI O.	Mathématiques
	MOUSSA André	Chaire de Chimie Nucléaire
	TRAYNARD Philippe	Chaire de Chimie Générale
	SOUTIF Michel	Chaire de Physique Générale
	CRAYA Antoine	Chaire d'Hydrodynamique
	REULOS R.	Théorie des Champs
	BESSON Jean	Chaire de Chimie
	AYANT Yves	Physique Approfondie
	GALLISSOT	Mathématiques
Mlle	LUTZ Elisabeth	Mathématiques
	BLAMBERT Maurice	Chaire de Mathématiques
	BOUCHEZ Robert	Physique Nucléaire
	LLIBOUTRY Louis	Géophysique
	MICHEL Robert	Chaire de Minéralogie et Pétrographie
	BONNIER Etienne	Chaire d'Electrochimie et d'Electrométallurgie
	DESSEAUX Georges	Chaire de Physiologie Animale
	PILLET E	Chaire de Physique Industrielle et Electrotechnique
	YOCCOZ Jean	Chaire de Physique Nucléaire Théorique
	DEBELMAS Jacques	Chaire de Géologie Générale
	GERBER R.	Mathématiques
	PAUTHENET R.	Electrotechnique
	VAUQUOIS B.	Chaire de calcul électronique
	BARJON R.	Physique Nucléaire
	BARBIER Jean-Claude	Chaire de Physique
	SILBER R.	Mécanique des Fluides
	BUYLE-BODIN Maurice	Chaire d'Electronique
	DREYFUS B.	Thermodynamique

PROFESSEURS TITULAIRES suite

MM. KLEIN J.	Mathématiques
VAILLANT F.	Zoologie et Hydrobiologie
ARNAUD Paul	Chaire de Chimie
SENGEL P.	Chaire de Zoologie
BARNOUD F.	Chaire de Biosynthèse de la Cellulose
BRISSONNEAU P.	Physique
GAGNAIRE	Chaire de Chimie Physique
Mme KOFLER L.	Botanique
DEGRANGE Charles	Zoologie
PEBAY-PEROULA J.C.	Physique
RASSAT A.	Chaire de Chimie Systématique
DUCROS P.	Chaire de Cristallographie Physique
DODU Jacques	Chaire de Mécanique Appliquée I.U.T.
ANGLES D'AURIAC P.	Mécanique des Fluides
LACAZE A.	Thermodynamique

PROFESSEURS SANS CHAIRE

MM. GIDON P.	Géologie et Minéralogie
GIRAUD P.	Géologie
PERRET R.	Servomécanisme
Mme BARBIER M.J.	Electrochimie
Mme SOUTIF J.	Physique
COHEN J.	Electrotechnique
DEPASSEL R.	Mécanique des Fluides
GASTINEL A.	Mathématiques Appliquées
GLENAT R.	Chimie
BARRA J.	Mathématiques Appliquées
COUMES A.	Electronique
PERRIAUX J.	Géologie et Minéralogie
ROBERT A.	Chimie Papetière
BIAREZ J.P.	Mécanique Physique
BONNET G.	Electronique
CAUQUIS G.	Chimie Générale
BONNETAIN L.	Chimie Minérale
DEPOMMIER P.	Etude Nucléaire et Génie Atomique
HACQUES Gérard	Calcul Numérique
POLOUJADOFF M.	Electrotechnique

PROFESSEURS ASSOCIES

MM. NAPP-ZINN	Botanique
RODRIGUES Alexandre	Mathématiques Pures
STANDING Kenneth	Physique Nucléaire

MAITRES DE CONFERENCES

MM. LANCIA Roland	Physique Atomique
Mme KAHANE J.	Physique
DEPORTES C.	Chimie
Mme BOUCHE L.	Mathématiques
SARROT-REYNAUD	Géologie Propédeutique

MAITRES DE CONFERENCES suite

Mme	BONNIER M.J.	Chimie
	KAHANE A.	Physique Générale
	DOLIQUE J.M.	Electronique
	BRIERE G.	Physique M.P.C.
	DESRE G.	Chimie S.P.C.N.
	LAJZROWICZ J.	Physique M.P.C.
	VALENTIN P.	Physique M.P.C.
	BERTRANDIAS J.P.	Mathématiques Appliquées T.M.P.
	LAURENT P.	Mathématiques Appliquées T.M.P.
	CAUBET J.P.	Mathématiques Pures
	PAYAN J.J.	Mathématiques
Mme	BERTRANDIAS F.	Mathématiques Pures M.P.C.
	LONGEQUEUE J.P.	Physique
	NIVAT M.	Mathématiques Appliquées
	SOHM J.C.	Electrochimie
	ZADWORNY F.	Electronique
	DURAND F.	Chimie Physique
	CARLER G.	Biologie Végétale
	AUBERT G.	Physique M.P.C.
	DELPUECH J.J.	Chimie Organique
	PFISTER J.C.	Physique C.P.E.M.
	CHIBON P.	Biologie Animale
	IDELMAN S.	Physiologie Animale
	BOUVARD Maurice	Hydrologie
	RICHARD Lucien	Botanique
	PELMONT Jean	Physiologie Animale
	BLOCH D.	Electrotechnique I.P.
	BOUSSARD J.Claude	Mathématiques Appliquées I.P.
	MOREAU René	Hydraulique I.P.
	BRUGEL L.	Energétique I.U.T.
	SIBILLE R.	Construction Mécanique I.U.T.
	ARMAND Yves	Chimie I.U.T.
	BOLLIET Louis	Informatique I.U.T.
	KUHN Gérard	Energétique I.U.T.
	GERMAIN Jean Pierre	Construction Mécanique I.U.T.
	CONTE René	Thermodynamique
	JOLY Jean René	Mathématiques Pures
	PIERY Yvette	Biologie Animale
	BENZAKEN Claude	Mathématiques Appliquées

MAITRES DE CONFERENCES ASSOCIES

MM.	SAWCZUK A.	Mécanique des Fluides
	CHEEKE J.	Thermodynamique
	YAMADA O.	Physique du Solide
	NATR Lubomir	B.M.P.V.
	NAYLOR Arch	Physique Industrielle
	SILBER Léo	Radioélectricité
	NAZAKI Akihiro	Mathématiques Appliquées
	RUTLEDGE Joseph	Mathématiques Appliquées
	DONOHO Paul	Physique Générale
	EGGER Kurt	B.M.P.V.

Je tiens à exprimer ici toute ma reconnaissance à

Monsieur le Professeur J. KUNTZMANN, Directeur du Service de Mathématiques Appliquées de l'Université de Grenoble.

qui a dirigé ce travail et qui par son aide et 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 le Professeur B.VAUQUOIS, Directeur du Centre d'Etudes pour la Traduction Automatique,

et,

Monsieur C. BENZAKEN, Maître de Conférences à la Faculté des Sciences de Grenoble qui ont bien voulu accepter de faire partie du Jury.

Je tiens encore à remercier tous ceux qui m'ont aidé dans mon travail de programmation ainsi que le Secrétariat et le Service Tirage qui ont eu pour tâche ingrate, la réalisation matérielle de cette thèse.

P L A N

Pages

CHAPITRE I - DEFINITIONS ET PROPRIETES GENERALES DES FORMES LEXICOGRAPHIQUES I-1

1 - DEFINITIONS GENERALES
- 1.a. Lettre
- 1.b. Variable
- 1.c. Fonction $\emptyset$-Booléenne
- 1.d. Arborescence
- 1.e. Arborescence lexicographique locale
- 1.f. Relation entre arborescence lexicographique locale et les expressions booléennes en somme de produit.
- 1.g. Forme lexicographique locale
- 1.h. Forme lexicographique
- 1.i. Forme lexicographique irréductible
- 1.j. Bifurcation
- 1.k. Fermetures indépendantes

2 - REPRESENTATIONS ARBORESCENTES DES FORMES LEXICOGRAPHIQUES IRREDUCTIBLES . I-10
- 2.a. Réseau lexicographique local
- 2.b. A l'aide U
  - 2.b.1. Définition de U
  - 2.b.2. Représentation de U
  - 2.b.3. Propriétés de U
  - 2.b.4. Réseaux U-arborescents

3 - OPERATIONS SUR LES FORMES LEXICOGRAPHIQUES I-16
- 3.a. Complément d'une forme irréductible
  - 3.a.1. Sur le réseau lexicographique
  - 3.a.2. Sur le réseau $\mathcal{U}$-arborescent
  - 3.a.3. Sur le réseau U-arborescent
- 3.b. Duale d'une écriture lexicographique irréductible
  - 3.b.1. Sur les réseaux $\mathcal{U}$-arborescents
  - 3.b.2. Sur les réseaux U-arborescents
  - 3.b.3. Sur les réseaux lexicographiques
- 3.c. Structure des Ecritures lexicographiques dont les variables générales n'ont pas d'argument commun
  - 3.c.1. Somme
  - 3.c.2. Produit
- 3.d. Structures des Ecritures lexicographiques relatives à un ordre donné.

Pages

CHAPITRE II - OPTIMISATION DES ECRITURES LEXICOGRAPHIQUES D'UNE MEME FONCTION CRITERES DE COUT II-31

1 - DIVERSES DEFINITIONS.. II-32
- 1.a. Différents coûts
- 1.b. Forme lexicographique locale c-optimale

2 - EVALUATION DES DIFFERENTS COUTS D'UNE ECRITURE LEXICOGRAPHIQUE II-33
- 2.a. Réseaux $\mathcal{U}$, Coût $\mathcal{A}$
  - 2.a.1. A partir du réseau de E_1
  - 2.a.2. A partir de l'écriture polynomiale de E_1
- 2.b. Réseaux U, Coût A
  - 2.b.1. Réseau lexicographique de E
  - 2.b.2. Ecriture polynomiale de E
- 2.c. Correspondance des divers coûts d'une même écriture
  - 2.c.1. Expression liant A et $\mathcal{A}$
  - 2.c.2. Expression liant A et L

CHAPITRE III - ALGORITHME DE RECHERCHE DES ECRITURES LEXICOGRAPHIQUES LOCALES OPTIMALES DES FONCTIONS INCOMPLETES III-48

1 - PRINCIPE ET MISE EN OEUVRE III-49
- 1.a. Remarque sur une écriture lexicographique
- 1.b. Principe de l'Algorithme
- 1.c. Mise en oeuvre
- 1.d. Obtention des solutions
- 1.e. Cas particulier des fonctions complètes

2 - ASPECT PRATIQUE III-54
- 2.a. Réalisation
- 2.b. Remarques
- 2.c. Résultat
- 2.d. Remarque

3 - AUTRE MANIERE D'OBTENIR UNE SOLUTION OPTIMALE...................... III-58
- 3.a. Grandes lignes de la méthode
- 3.b. Restrictions
- 3.c. Résultat
- 3.d. Remarque

4 - SIMPLIFICATIONS PRELIMINAIRES III-63
- 4.a. Parallèle en fonctions booléennes et Ø-Booléennes
- 4.b. Conséquences
- 4.c. Théorème
- 4.d. Remarque

5 - EXEMPLE III-69

Pages

CHAPITRE IV - DIVERS PROGRAMMES ET RESULTATS RELATIFS AUX FONCTIONS COMPLETES IV-79

1 - PROGRAMMES ET RESULTATS MACHINE IV-80

1.a. Programmes
1.b. Résultats
1.c. Elimination heuristique

2 - COMPARAISON DES OPTIMISATIONS SUR LES DIFFERENTS COUTS IV-89
2.a. A et $\mathcal{a}$
2.b. $\mathcal{a}$ et L
2.c. A et L

3 - FORMES LEXICOGRAPHIQUE DE f ET DE f' IV-99
3.a. Ordres optimaux de f et f'
3.b. Etude machine

CHAPITRE V - QUELQUES PROPRIETES RELATIVES AUX FORMES LEXICOGRAPHIQUES IRREDUCTIBLES DES FONCTIONS COMPLETES V-101

1 - FONCTION DE LA FORME xf(Y) + x'g(Z).................................... V-102
1.a. Théorème
1.b. Remarque
1.c. Exemple

2 - PROPRIETES DUES AUX MONOMES PREMIERS V-103
2.a. Théorème
2.b. Thèorème
2.c. Théorème

3 - ENCADREMENT DU NOMBRE DE MONOMES ET DU NOMBRE DE LETTRES DES SOLUTIONS L-OPTIMALES D'UNE FONCTION COMPLETE DE N LETTRES V-106
3.a. Encadrement
3.b. Détermination des bornes supérieures en fonction de n
3.c. Exemple montrant que la borne supérieure est souvent trop élevée

4 - SOMME DE MONOMES DISJOINTS ET FORME LEXICOGRAPHIQUE V-109
4.a. Définition et propriétés
4.b. Applications avec des vecteurs de vérité

5 - RECHERCHE DE LA LEXICOGRAPHICITE D'UNE FONCTION φ DONNEE SOUS FORME POLYNOMIALE V-112
5.a. Méthode
5.b. Exemple

INTRODUCTION

Le présent travail a pour but l'étude des formes lexicographiques des fonctions booléennes simples complètes ou incomplètes. Ces écritures ont pour application directe la synthèse arborescente des fonctions booléennes à l'aide de l'opérateur U.

Nous présentons un algorithme de construction des telles formes que nous avons programmé ainsi qu'un certain nombre de propriétés s'y rapportant.

B I B L I O G R A P H I E

[1] BERGE Théorème des graphes, Dunod, 1958

[2] M. CARVALLO Monographie des treillis et Algèbre de Boole. Gauthier Villars, 1962

[3] M. CHEIN Etude des décompositions d'un réseau. Applications à l'écriture des fonctions booléennes en sommes et produit.
Thèse de Docteur Ingénieur, Grenoble, oct. 1967

[4] P. DESCHIZEAUX Synthèse de fonctions booléennes générales
Thèse de Docteur Ingénieur, Grenoble, juin 1967

[5] Mme DUBREIL JACOTIN Leçon d'Algèbre Moderne
Dunod, 1964

[6] J. KUNTZMANN Théorie des réseaux
Cours Faculté des Sciences, Grenoble, oct. 1966

[7] J. KUNTZMANN Algèbre de Boole, Dunod, Avril 1965

[8] J.P. LAMOITIER Formes arborescentes des fonctions booléennes.
Thèse de 3ème Cycle, Grenoble, Juin 1963

[9] J. MACHERAS Synthèse d'une fonction booléenne par l'opérateur U.
Thèse de 3ème Cycle, Grenoble, Juin 1966

CHAPITRE I

DEFINITIONS GENERALES ET PROPRIETES DES FORMES LEXICOGRAPHIQUES

1. - Définitions générales

2. - Représentations arborescentes des formes irréductibles

3. - Opérations sur les formes irréductibles.

CHAPITRE I

DEFINITIONS GENERALES ET PROPRIETES DES FORMES LEXICOGRAPHIQUES

-:-:-:-

I - DEFINITIONS GENERALES

Tout d'abord un rappel de quelques définitions.

1.a.- une lettre est une quantité booléenne simple ne tenant pas compte de l'accentuation et faite d'un seul caractère

a et a' sont la même lettre a.

1.b.- une variable est une quantité booléenne simple obtenue à partir d'une lettre, avec ou sans accentuation

a et a' sont deux variables différentes d'ailleurs complémentaires.

1.c.- une fonction Ø-booléenne est une fonction φ définie par ses bornes booléennes $\underline{f}$ et $\overline{f}$ respectivement inférieure et supérieure, et telle que toutes les valeurs que φ puisse prendre vérifient

$$\underline{f} \leq \varphi \leq \overline{f}$$

on dira de plus qu'une fonction booléenne g est compatible avec φ si l'on a pour toutes les valeurs de la variable générale dont φ et g dépendent

$$\underline{f} \leq g \leq \overline{f}$$

1.d.- une arborescence est un arbre (réseau connexe sans cycle) orienté dans lequel un seul noeud a un degré inférieur nul, tous les autres l'ont égal à 1.

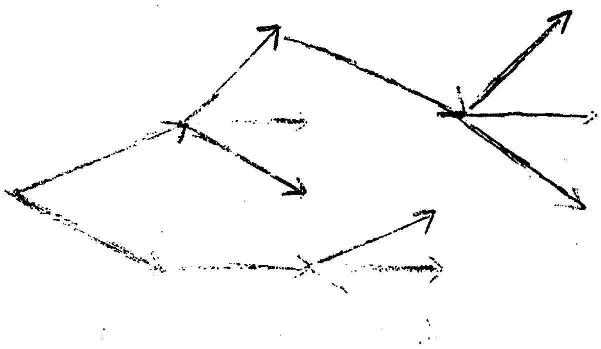
Le degré inférieur d'un noeud est égal au nombre d'arêtes orientées qui arrivent à lui, tandis que le degré supérieur d'un noeud est égal au nombre d'arêtes orientées qui en partent.

Le noeud particularisé dans la définition de l'arborescence est appelé racine de l'arborescence. On peut appeler feuille un noeud dont le degré supérieur est nul.

Propriété.

Une arborescence a une seule racine mais peut avoir plusieurs feuilles.

Exemple d'arborescence.



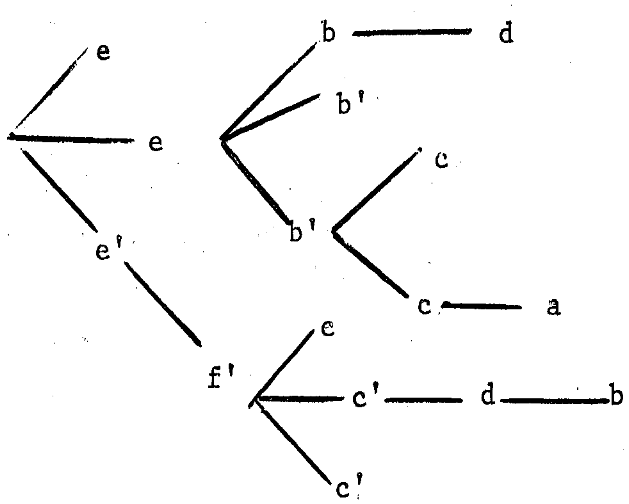
pour des raisons de dessin, nous considérerons toujours que l'orientation va de la racine aux feuilles et rendant le dessin clair, nous ne mettrons pas de signe d'orientation.

1.e.- Une arborescence lexicographique locale sur un ensemble $\mathcal{E}$ de lettres est un réseau arborescent obtenu en affectant des noeuds à des variables de telle manière que :

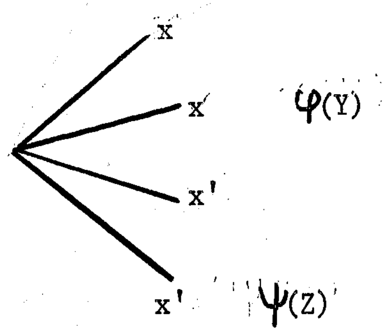
* d'un noeud partent vers les feuilles, soit uniquement des branches liées à des noeuds correspondant à la même lettre, soit aucune branche pour les feuilles.

**deux chemins quelconques de longueur q et r $q \leq r$ allant de la racine aux feuilles, ayant leurs p premières variables identiques et dans le même ordre, $p \leq q$ (si p = q prendre p = q-1) se servent effectivement des mêmes p premiers noeuds, sont distincts et n'ont jamais, chacun plusieurs fois la même lettre

Exemple. $\mathcal{E}$={ a,b,c,d,e,f}



<u>Propriété</u>. De chaque noeud n'appartenant pas aux feuilles, il part au plus 4 branches.



S'il y avait en plus x.y(T) par exemple, la condition ** ne serait pas vérifiée.

1.f.- <u>Relation entre une arborescence lexicographique locale et les expressions booléennes en somme de produit</u>.

1.f.1.- Si nous faisons la somme de tous les chemins de l'arborescence écrits sous la forme de concaténation (produit) de variables, on obtient une fonction booléenne telle que

- tous les monômes ont la même première lettre
- tous les monômes de plus de p lettres ayant p variables identiques ont la même p+1ème lettre.

1.f.2.- Réciproquement : il est toujours possible d'associer à une fonction booléenne f(X) une arborescence lexicographique locale. Pour cela il faut mettre en facteurs des couples de variables dans un certain ordre que nous préciserons plus tard.

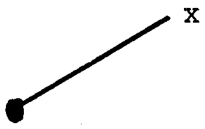
$$f(X) = f_1(x,Y) = x\ f_1(1,Y) + x'\ f_1(0,Y)$$

Si nous ne nous permettons pas de simplification il est possible que l'on ait

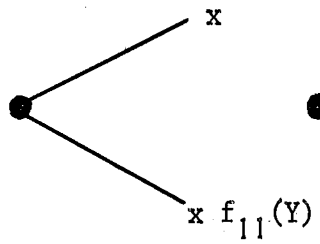
soit $f_1(1,Y) = 1 + f_{11}(Y)$

soit $f_1(0,Y) = 1 + f_{12}(Y)$

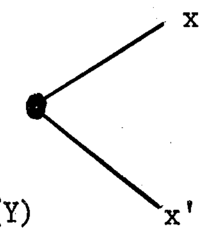
soit les deux d'ailleurs. Nous distinguerons l'apparition de ces constantes pour avoir de f(X) une représentation donnée par un des 15 réseaux



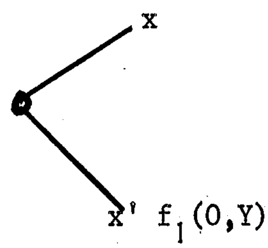
- 1 -



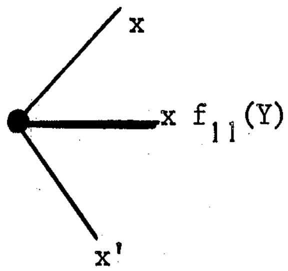
- 2 -



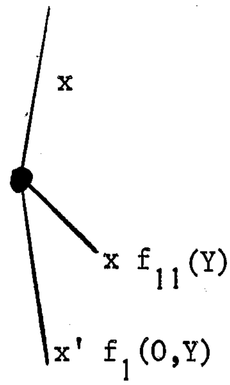
- 3 -



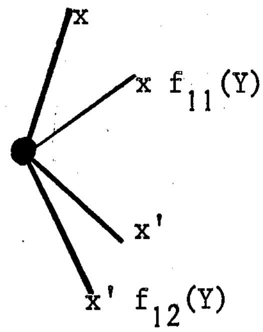
- 4 -



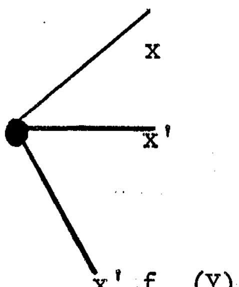
- 5 -



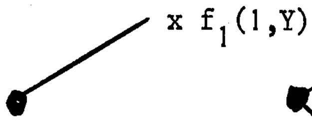
- 6 -



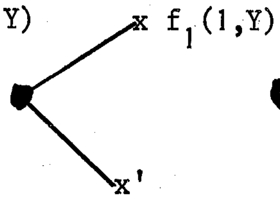
- 7 -



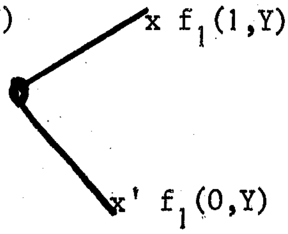
- 8 -



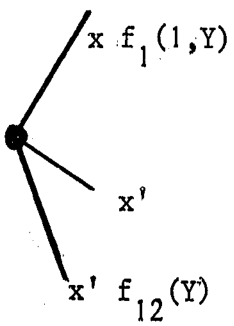
- 9 -



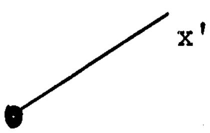
- 10 -



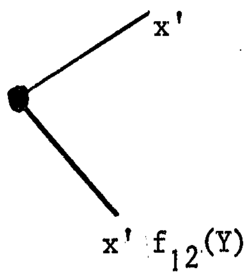
- 11 -



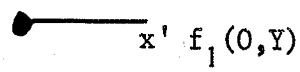
- 12 -



- 13 -



- 14 -



- 15 -

il y a en effet

$$C_4^4 + C_4^3 + C_4^2 + C_4^1 = 15.$$

Nous pouvons recommencer cette mise en facteurs des quantités $f_1(1,Y)$ et $f_1(0,Y)$ ou si c'est le cas des quantité $f_{11}(Y)$ et $f_{12}(Y)$ jusqu'à ce que les quantités obtenues soient réduites à la constante 1 auquel cas il n'y a plus de mise en facteurs possible. La représentation obtenue vérifie les conditions de définition de l'arborescence lexicographique.

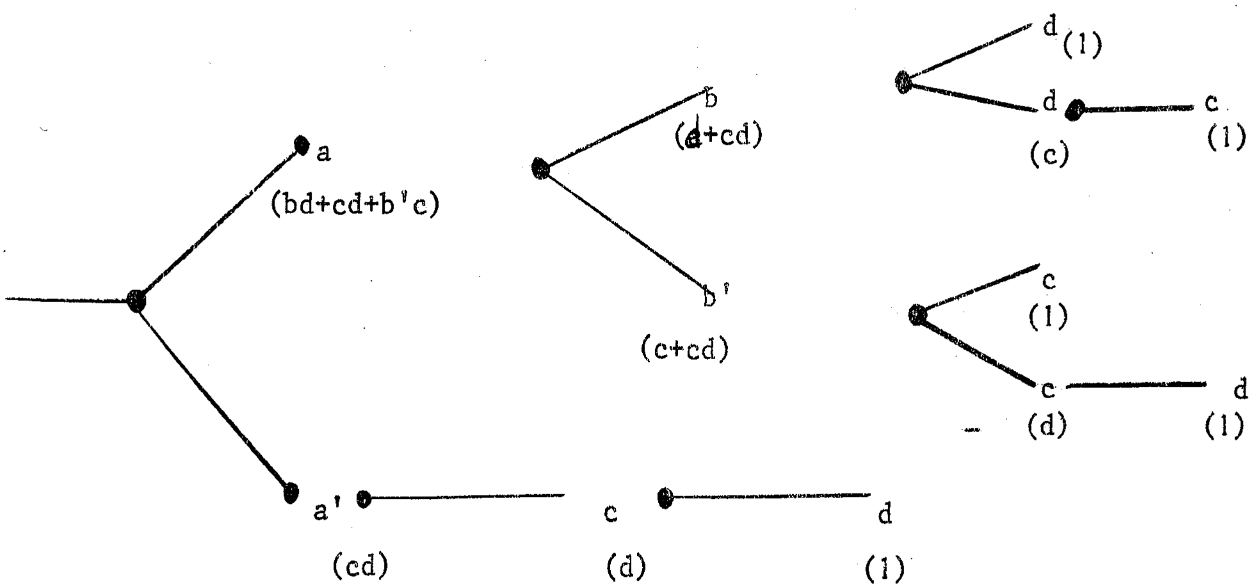
En effet, d'un noeud il ne part des branches que vers des noeuds affectés à la même lettre celle qui est mise en facteurs; de plus sont regroupés sur un même chemin de longueur p, tous les monômes de q variables, $q \geqslant p$ ayant les mêmes p premières variables et dans le même ordre.

<u>Propriété</u>.

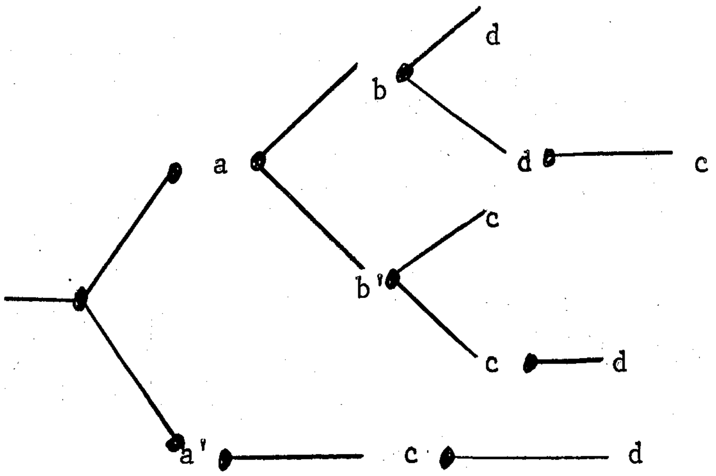
Quelle que soit f(X) l'arborescence lexicographique locale peut ne pas être unique. Elle dépend de l'ordre des variables suivant lequel nous effectuons des mises en facteurs. Nous qualifierons cet ordre d'ordre lexicographique local par opposition à l'ordre lexicographique où toutes les mises en facteurs se font dans le même ordre quelle que soit la branche choisie.

<u>Exemple</u>.

$$f = abd + cd + ab'c$$



d'où l'arborescence lexicographique locale



que l'on peut écrire

$$abd + abdc + ab'c + ab'cd + a'cd$$

d'où les définitions :

1.g.- <u>une forme lexicographique locale</u> d'une expression booléenne en somme de produits est l'écriture d'une arborescence lexicographique locale obtenue par mise en facteurs suivant un ordre lexicographique local.

<u>Remarque</u> : elle peut ne pas être unique.

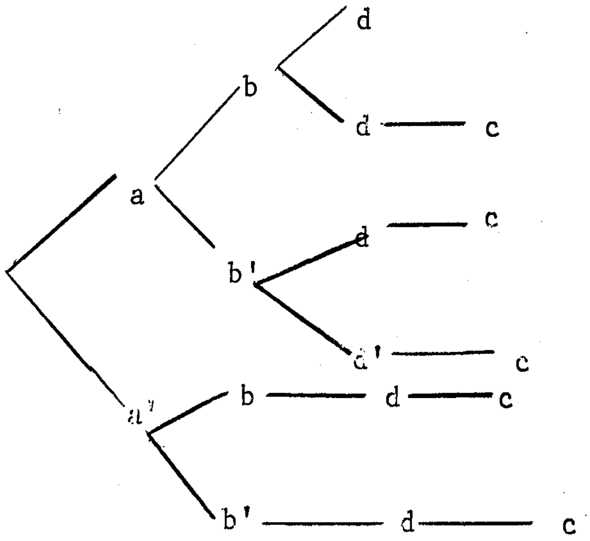
1.h.- une <u>forme lexicographique</u> d'une expression booléenne en somme de produits est une forme lexicographique locale avec un ordre lexicographique.

<u>Propriétés des formes lexicographiques</u> (locale ou non)

cf. § 1.f.1. elles sont en effet les mêmes.

Exemple de forme lexicographique sur l'expression précédente avec l'ordre lexicographique a - b - d - c

une forme lexico.(locale) d'une fonction φ-booléenne φ est une forme lexico. (locale) d'une fonction booléenne compatible avec φ



1.i.- une <u>forme lexicographique</u> (resp. locale) <u>irréductible</u> F_1 d'une forme lexicographique (resp. locale) F est l'écriture obtenue en appliquant à F les deux opérations suivantes :

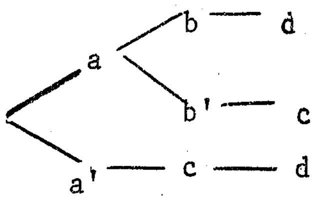
- allègement : remplacement de Am Θ+ Am par Am
- réduction : remplacement de Amd + Amd' par Am

et ceci jusqu'à ne plus pouvoir les appliquer.

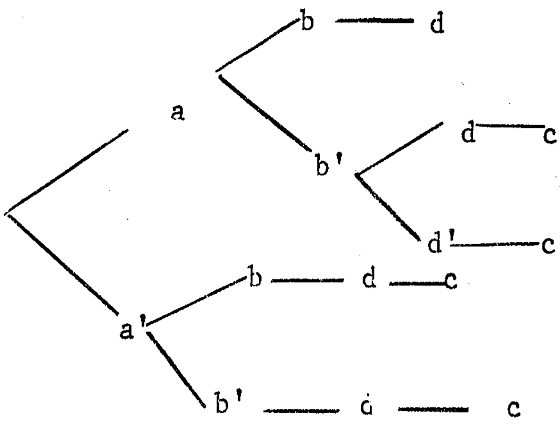
N.B. *Nous dirons indifféremment <u>forme</u> et <u>écriture</u>.*

Reprenons l'exemple précédent.

Une forme irréductible locale est



Une forme irréductible est



Propriété de F_1

Parmi tous les monômes de F_1, il n'y en a aucun qui soit multiple d'un autre.

Conséquence.

Le produit d'un monôme quelconque de F_1 par un autre monôme quelconque de F_1 est nul. Ce que l'on peut écrire en posant

$$F_1 = \sum_I m_i$$

$$\forall j \in I, \quad \forall k \in I \quad j \neq k \qquad m_j \cdot m_k = 0$$

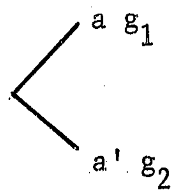
Parce que deux monômes quelconques ont une tête de monôme commune, peut-être vide, et une variable sous la forme directe dans l'un, complémentée dans l'autre.

1.j.- La structure d'arborescence nous amène à parler encore de deux notions.

Une bifurcation sur la lettre a sera l'écriture

$$a\ g_1 + a'\ g_2$$

* obtenue à partir de deux écritures g_1 et g_2 dont le réseau est



Lorsque g_1 (ou g_2) sera nulle nous aurons un cas particulier de la bifurcation appellée aussi adjonction de la variable x à l'écriture g la quantité

$$x \cdot g$$

on peut considérer chaque variable x comme l'adjonction de x à la constante 1.

1.k.- Fermetures indépendantes

1.k.1.- Définitions cf. [3] p.65.

$\varphi(X.Y)$ est une fonction complète, on note par $\overline{\varphi}^{X}(X,Y)$ sa fermeture supérieure indépendante de X et par $\underline{\varphi}_{X}(X,Y)$ l'inférieure définies par

$$\overline{\varphi(X,Y)}^{X} = \sum_{x_1\ldots x_p} \varphi(x_1\ldots x_p,Y)$$

$$\underline{\varphi(Z,Y)}_{Z} = \prod_{z_1\ldots z_q} \varphi(z_1\ldots z_q,Y)$$

les opérations de sommation et de produit sont étendues aux 2^p de la variable générale X, (2^q pour Z)

Remarque.

- La fermeture supérieure est aussi une projection
- Ces fermetures sont uniques.

1.k.2.- Obtention des fermetures

- Inférieure. Nous obtenons $\underline{\varphi}_{X}$ à partir de la base complète de φ en supprimant tous les monômes contenant une ou plusieurs lettres dont X dépend. Nous sommes contraints de partir de la base complète de φ car nous devons supprimer des monômes et obtenir un résultat unique. Nous ne pouvons partir de bases premières non uniques.

En effet,

$$\varphi(X,Y) = ac + b'c'e'$$

$\underline{\varphi}_{c}$ donnerait 0

Tandis que la base complète vaut

$$ac + b'c'e' + ab'e'$$

où $$\underline{\varphi}_{c} = ab'e'$$

.. Supérieure. Nous obtenons $\overline{\varphi}^{X}$ à partir d'une forme polynomiale quelconque de φ en supprimant dans tous les monômes toutes les lettres dont X dépend.

Ainsi
$$\varphi(XY) = ac + b'c'e'$$
$$\overline{\varphi}^{c} = a + b'e'$$

en partant de la base complète nous aurions le même résultat.

Propriété des fermetures.

Quelle que soit la fonction complète $\varphi(XY)$, nous avons toujours

$$\boxed{\underset{X}{\sqcap}\varphi\,(X,Y) \leqslant \overline{\varphi}^{X}(X,Y)}$$

2. REPRESENTATIONS ARBORESCENTES DES FORMES LEXICOGRAPHIQUES IRREDUCTIBLES.

Nous nous attachons à représenter les formes lexicographiques irréductibles le plus fidèlement possible et non à réaliser des synthèses arborescentes de fonctions booléennes (cf. §2-b-2).

2.a.- Représentation au moyen de l'arborescence lexicographique locale du § 1-e. Comme il s'agit de formes irréductibles nous nommerons un tel réseau, réseau lexicographique local ou non.

Propriétés.

2.a.1. - Pour chaque noeud n'appartenant pas aux feuilles le degré supérieur est au plus égal à 2.

Au § 1.f dans les 6 réseaux ce degré était au plus égal à 4. Maintenant il s'agit de formes irréductibles les seuls dessins sont ceux numérotés 1-3-4-9-10-11-13-15-, c'est-à-dire les seuls où il n'est pas possible de faire des allègements.

2.a.2.- A toute écriture irréductible (resp. locale) il ne correspond q qu'un seul réseau lexicographique (resp. local).

2.b.- <u>Représentation à l'aide de l'opérateur U</u>

2.b.1.- <u>Définition</u>

$$U(\gamma, \varphi, \Psi) = \gamma\varphi + \gamma'\Psi$$

où γ, φ_i et Ψ sont des fonctions booléennes. γ peut être appellée fonction de commande : en effet

Si $\gamma = 1$ $\qquad U(1,\varphi,\Psi) = \varphi$

Si $\gamma = 0$ $\qquad U(0,\varphi,\Psi) = \Psi$

φ et Ψ en sont alors les entrées.

Comme il s'agit de formes lexicographiques et non de synthèses de fonctions avec cet opérateur, nous ne mettrons comme fonction de commande une seule lettre appellée lettre de commande.

$$U(a,\varphi,\Psi) = a\varphi + a'\Psi$$

En effet,

$$U(a',\varphi,\Psi) \text{ donnerait } a'\varphi + a'\Psi = U(a,\Psi,\varphi)$$

où les entrées sont échangées.

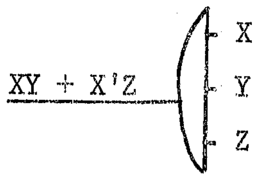
2.b.2.- <u>Représentation de l'opérateur U</u>

Nous en utiliserons deux

- celui noté $\mathcal{U}$ à 3 entrées, la supérieure étant la commande

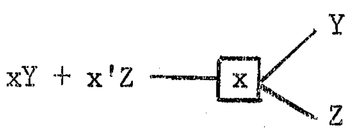
$$\mathcal{U}(X,Y,Z)$$

représenté par



- celui noté U à 2 entrées, la lettre de commande étant mise à l'intérieur

$U_x(Y,Z)$ représenté par

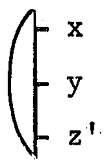


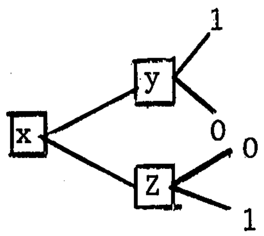
On remarquera que l'ordre des entrées est très important.

- <u>différence entre ces représentations</u>.

La différence fondamentale vient de la commande où tout est permis pour $\mathcal{U}$, seule une lettre pour U. Seulement, nous ne nous servirons pas de cette distinction. Seul subsiste le fait que les variables pour $\mathcal{U}$ sont supposées réalisées sans le concours d'opérateur ce qui n'est pas le cas pour U.

xy + x'z' se représente en effet par





2.b.3.- <u>Propriétés de ces opérateurs</u>

Toute fonction booléenne se décompose à l'aide des opérateurs U et $\mathcal{U}$

En effet,

$f(X)$ où X dépend de n lettres $a_1, a_2, \dots, a_n$

on peut poser

$$f_1(a_1, Y_1) = f(X)$$

Y_1 dépendant des arguments de X saut a, donc de n-1 lettres

$$f(X) = f_1(a_1, Y_1) = a_1\ f_1(1, Y_1) + a'\ f_1(0, Y_1)$$
$$= U\ (a_1,\ f_1(1, Y_1),\ f_1(0, Y_1))$$

de même

$$f_2(1, a_2, Y_2) = f_1(1, Y_1)$$

Y_2 dépendant des n-2 lettres $a_3, a_4, \dots, a_n$

D'une manière générale

$$f_i(S_{i-1},a_i,Y_i) = f_{i-1}(S_{i-1}, Y_{i-1})$$

S_{i-1} est une suite des $i-1$ premières variables égales à 0 ou 1.
Y_i dépend des $(n-i)$ lettres $a_{i+1},\ldots,a_n$
et on a

$$f_{i-1}(S_{i-1},Y_{i-1}) = U\ (a_i;\ f_i(S_{i-1},1,Y_i),\ f_i(S_{i-1},0,Y_i))$$

nous devons maintenant distinguer deux cas distincts relatifs aux opérateurs $\mathcal{U}$ et U.

1°/ Nous ne voulons comme entrées terminales c'est-à-dire des opérateurs des décompositions les plus poussées que les constantes 0 et 1. Nous nous arrêtons au plus à f_n ou Y_n est vide; pour parvenir à f_1 il faut au maximum une décomposition

aux f_2, 2, aux f_3 2^2 aux f_i 2^{i-1}

et aux f_n 2^{n-1} au plus.

Soit au plus $1 + 2 + 2^2 \ldots + 2^{i-1} \ldots + 2^{n-1} = \sum_{j=0}^{n-1} 2^j = 2^n-1$

D'où le résultat suivant

> Toute fonction booléenne $f(X)$ dépendant de n variables peut être réalisée au moyen des constantes 0 et 1 et d'au plus 2^n-1 opérateur U

La décomposition peut ne pas être unique.

Nous réalisons la fonction $f(X)$ à l'aide de ces opérateurs U ayant comme entrées soit les constantes 0 et 1 soit les sorties d' autres opérateurs.

2°/ Parmi les deux entrées de chaque opérateur terminal nous prenons <u>au moins une variable</u>. Nous décomposons alors jusqu'aux f_{n-1}

Il y a donc au plus

$$1 + 2 + 2^2 \ldots + 2^{n-2} = \sum_{j=0}^{n-2} 2^j = 2^{n-1}-1 \text{ décompositions}$$

d'où le résultat.

Toute fonction booléenne f(X) dépendant de n lettres peut être réalisée au moyen de variables construites sur les lettres dont f dépend, des constantes 0 et 1 et d'au plus $2^{n-1}-1$ opérateurs $\mathcal{U}$ à la condition que l'une au moins des entrées d'opérateurs terminaux soit une variable.

Là aussi, la décomposition peut ne pas être unique, nous réalisons la fonction f(X) à l'aide de ces opérateurs $\mathcal{U}$ ayant comme entrées soit une des constantes 0 ou 1 soit des variables, soit encore des sorties d'autres opérateurs.

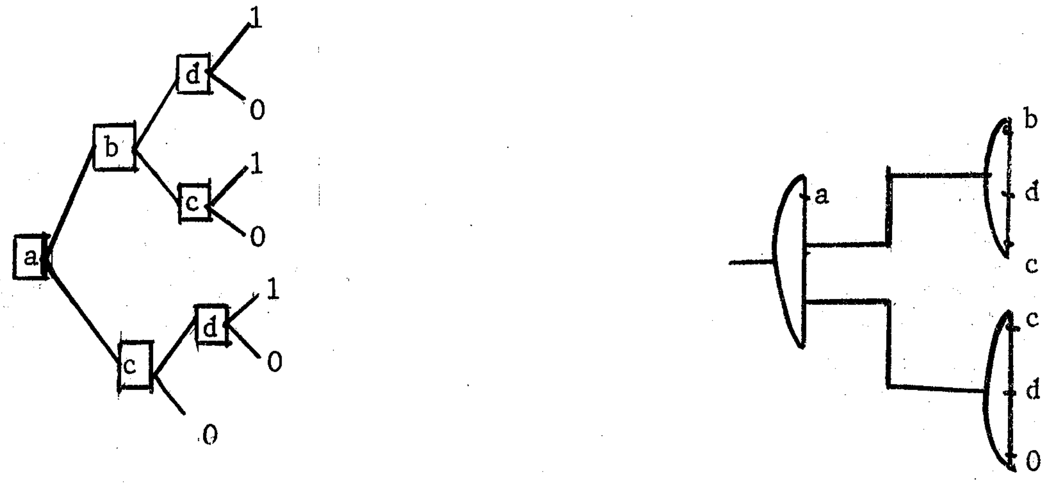
2.b.4.- <u>Réseaux U-arborescents.</u>

Nous n'appellerons ainsi des réseaux réalisés uniquement avec des opérateus U ou $\mathcal{U}$, il n'est pas question de faire un mélange d'opérateurs dans un réseau.

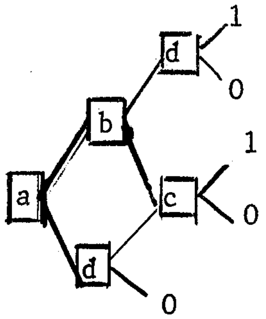
Un réseau sera dit arborescent si la sortie d'un opérateur ne sert d'entrée qu'à un seul opérateur au plus.

<u>Exemples de réseaux U-arborescents.</u>

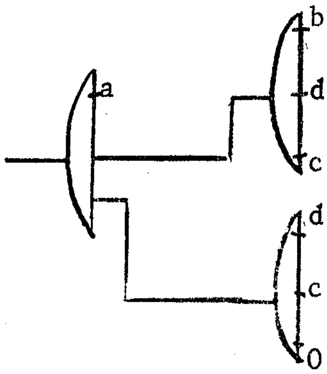
Exemple du § 1-i



Il est bien entendu que les réseaux U-arborescents ne sont pas les meilleurs ni les seuls pour réaliser les fonctions, ici par exemple on peut donner le réseau suivant



ou



celui-ci est resté arborescent.

Remarque 1.

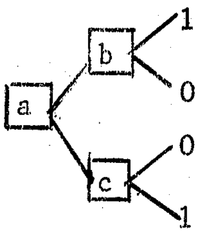
La décomposition en opérateurs U est unique dès qu'on se donne l'ordre des décompositions successives. Or une forme lexicographique (-locale) possède un et un seul ordre lexicographique(-local) qui est celui des décompositions successives d'où le résultat.

Propriété. A chaque forme lexicographique (-locale) il ne correspond qu'un seul réseau U-arborescent.

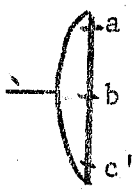
Remarque 2.

Il faut toujours moins d'opérateurs $\mathcal{U}$ que d'opérateurs U pour les réseaux arborescents car on se permet comme entrées des variables et non uniquement des constantes.

Exemple : ab + a'c'



ou



3. OPERATIONS SUR LES FORMES LEXICOGRAPHIQUES

On peut en considérer quatre :

- le complément d'une forme
- la duale d'une forme
- la somme de deux formes
- le produit de deux formes.

3.a.- Complément d'une forme irréductible

Nous trouvons dans [7] pp.58-59 une manière d'obtenir le complément d'une forme lexicographique E irréductible ou non, locale ou non. Il suffit pour celà de prendre les monômes de E les uns après les autres, de les amputer d'un nombre p de variables $p \geq 0$ et de complémenter dans les monômes obtenus la dernière variable conservée. Le complément E' est la somme des monômes ainsi obtenus qui ne sont diviseur d'aucun monôme de E.

Exemple.

$$E = agbc + agb'd + agb'df + a'ec + a'e'f + a'e'f'd$$

$\underline{agbc'}$	$\underline{agb'd'}$	$\underline{agb'df'}$	$\underline{a'ec'}$	$a'e'f'$	$\underline{a'e'f'd'}$
agb'	agb		$a'e'$	$a'e$	$a'e'f$
$\underline{ag'}$			a		
a'					

$$E' = agbc' + agb'd' + ag'bdf' + ag' + a'ec' + a'e'f'd'$$

3.a.1.- Sur le réseau lexicographique

En voici une nouvelle se servant des réseaux lexicographiques de formes irréductibles et basée sur le théorème suivant :

THEOREME

Si on extrait deux sommes, sans aucun monôme commun, d'une somme de monômes tous disjoints égale à 1, et ceci d'une manière quelconque l'une est le complément de l'autre.

Soient $\sum_{I} m_i$ et $\sum_{J} m_j$ les deux sommes

On a par hypothèse

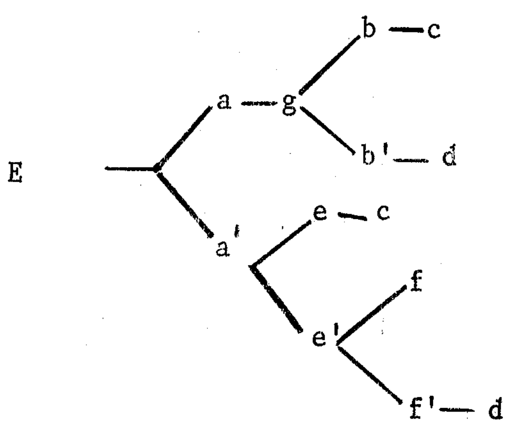
$$\sum_{I} m_i + \sum_{J} m_j = \sum_{I+J} m_k = 1$$

$$\sum_{I} m_i \cdot \sum_{J} m_j = \sum_{I \times J} m_i\, m_j = 0$$

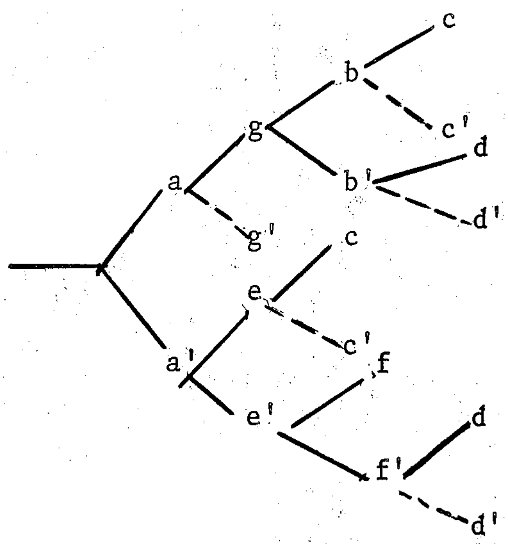
en particulier si la somme de départ est réduite à la quantité 1 la méthode donne 0 et 1 car 0 est la seule quantité disjointe avec 1.

3.a.1.b. - Obtention ou complément

Prenons le réseau lexicographique correspondant à la forme lexicographique locale irréductible considérée soit par exemple



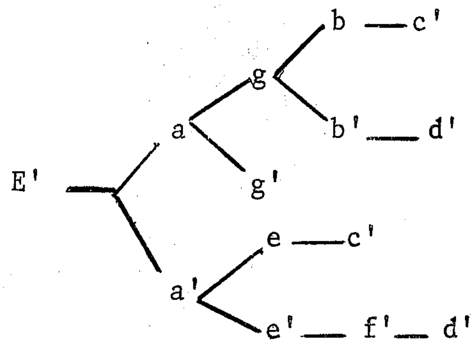
si l'on complète en pointillés ce réseau de manière à le rendre homogène (tous les degrés supérieurs non nuls égaux à deux) il vient



Ce réseau représente la fonction 1 (la somme de tous les monômes associés aux chemins est égale à 1) tous les monômes sont disjoints. Les deux arborescences extraites en prenant :

- d'une part les chemins n'ayant pas de pointillés
- de l'autre tous ceux ayant une arête en pointillés.

sont les réseaux arborescents des écritures lexicographiques locales <u>irréductibles</u> E et du complément E' car tous les monômes sont disjoints. Ici



$$E' = agbc' + agb'd' + ag' + a'ec' + a'e'f'd'$$

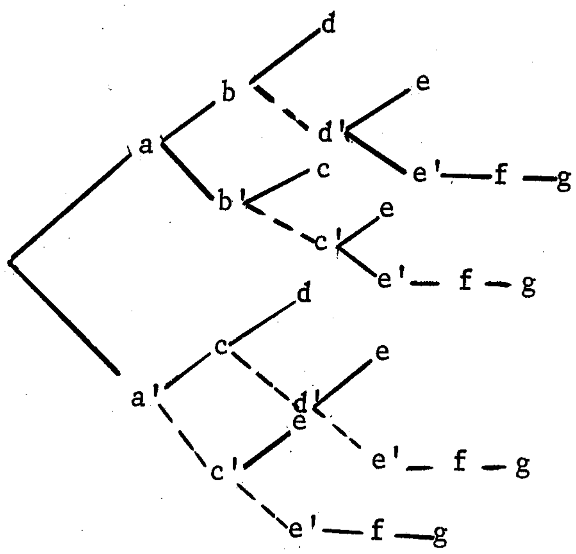
Remarque. Réseau arborescent de la fonction

$$f(Y) + f'(Y)\ g(Z)$$

Y et Z n'ayant aucun argument commun; pour obtenir un réseau, il suffit de mettre le réseau correspondant à g au bout de toutes les branches en pointillés du réseau de f + f'.

Exemple :

$$f(Y) = abd + cd + ab'c \qquad g(Z) = e + fg$$



3.a.2. Sur le réseau $\mathcal{U}$-arborescent

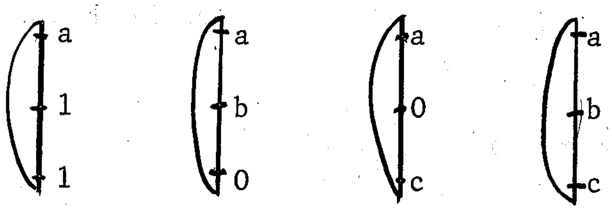
THEOREME

Pour obtenir le complément d'un tel réseau on peut complémenter les entrées inférieures des opérateurs $\mathcal{U}$ qui n'ont que des variables ou des constantes 0 et 1

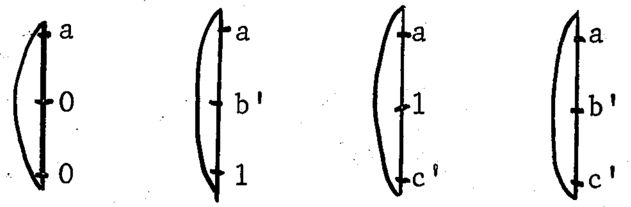
<u>Démonstration</u>. Nous procédons par induction sur le nombre M d'opérateurs.

Pour M = 1 il ne peut y avoir que

$1 - ab - a'c$ ou $ab + a'c$



les représentent



en sont les compléments

car

$0 + 1 = 1$

$ab + ab' + a' = 1$

$a'c + a + a'c' = 1$

$ab + a'c + ab' + a'c' = 1$

Supposons la propriété vraie pour M opérateurs et démontrons la pour M+1

Le plus général se présente sous la forme :

où l'opérateur dessiné est le M+1ème et φ et Ψ ont au plus M opérateurs. (φ ou Ψ éventuellement réduit à 0). φ et Ψ sont indépendantes de a.

Ceci s'écrit aussi $a\varphi + a'\Psi$

dont le complément est

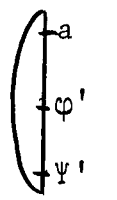
$$a\varphi' + a'\Psi'$$

puisque φ et Ψ ne dépendent pas de a.

Le réseau représentant comme précédemment.

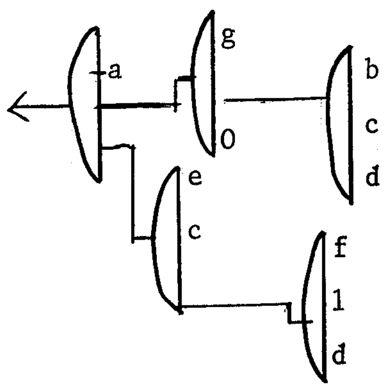
$$a\,\varphi' + a'\,\Psi'$$

a comme M+1 opérateur

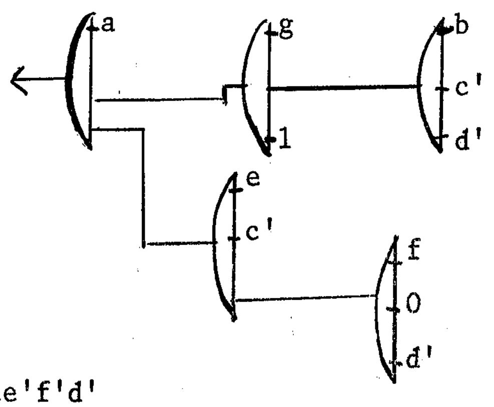


en appliquant la propriété à φ et Ψ on l'a pour cet opérateur ce qui achève la démonstration.

<u>Exemple</u> précédent (§ 3.a.1.β)



dont le complément est



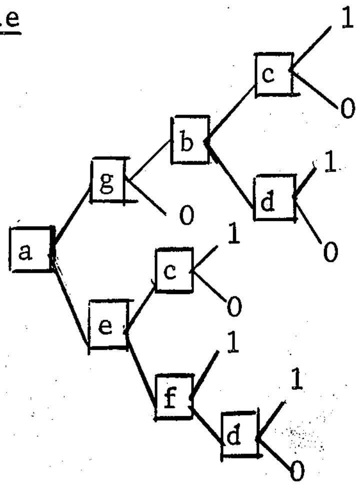
soit : $agbc' + agb'd' + ag' + aec' + ae'f'd'$

ce qui est bien exact.

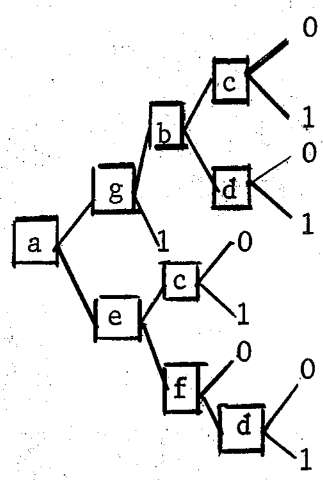
3.a.3.- <u>Sur le réseau U-arborescent</u>

On trouve dans [9] p. 10 une méthode qui consiste à complémenter toutes les entrées 0 ou 1 de chaque opérateur.

Exemple



donne



qui donne

agbc' + agb'd' + ag' + aec' + ae'f'd', c'est bien le résultat espéré.

3.b.- Duale d'une écriture lexicographique irréductible

Nous noterons cette opération * dont on rappelle l'idempotence

$$(f^*)^* = f$$

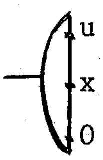
3.b.1.- Sur les Réseaux $\mathcal{U}$ arborescents

THEOREME

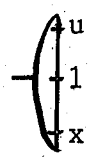
Si dans un réseau $\mathcal{U}$-arborescent représentant f on échange les deux entrées de chaque opérateur en prenant soin de complémenter l'entrée éventuellement constante, on obtient un réseau $\mathcal{U}$-arborescent de la duale de f.

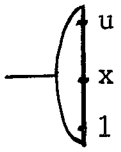
Nous démontrerons ce théorème par induction sur le nombre M d'opérateurs $\mathcal{U}$.

M = 1

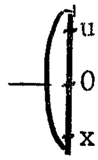


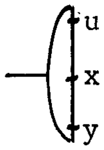
$u\,x \overset{*}{\Longleftrightarrow} u + x = u + u'x$



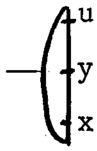


$u\ x + u'$ $\overset{*}{\Longleftrightarrow}$ $(u+x)u' = u'x$





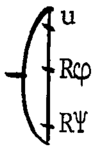
$ux + u'y$ $\overset{*}{\Longleftrightarrow}$ $(u+x)(u'+y) = uy + u'x$



x et y $\neq$ de constantes.

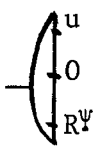
Supposons la propriété vraie pour M et démontrons la pour M + 1.

Soit un tel réseau où le M+1$^{\text{ème}}$ opérateur est par exemple

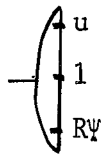


où $R\varphi$ et R^{Ψ} sont des sous arbres ayant au plus M opérateurs représentant φ et Ψ.

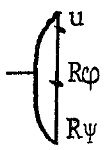
Si $R\varphi$ représente la fonction 0



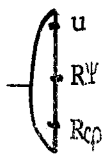
$u\ \Psi$ $\overset{*}{\Longleftrightarrow}$ $u + \Psi = u + u'\ \Psi$



et



$u\ \varphi + u'\ \Psi$ $\overset{*}{\Longleftrightarrow}$ $(u+\varphi)(u'+\Psi) = u\varphi + u'\Psi$

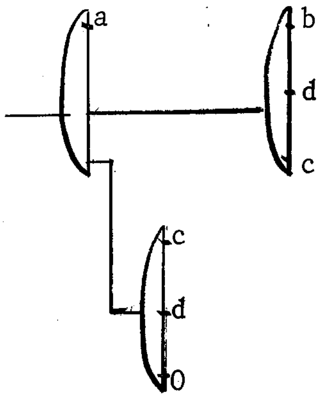


φ et Ψ pas constantes.

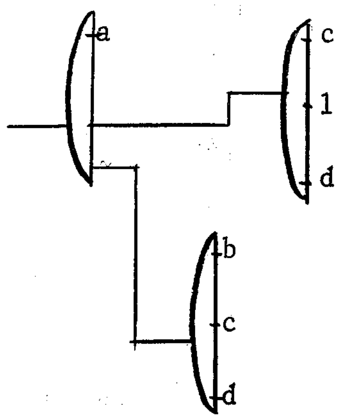
Ce qui démontre le théorème.

Exemple

R :=



R* :=



$f = abd + ab'c + a'cd$

$f^* = ac + ac'd + a'bc + a'b'd$

3.b.2.- Sur les réseaux U-arborescents

COROLLAIRE 1

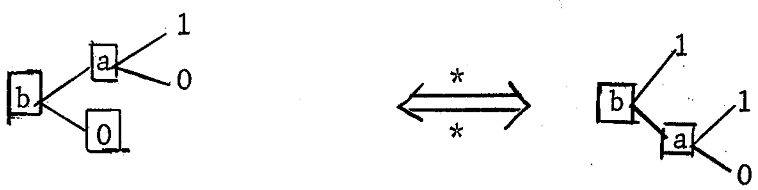
> Si dans un réseau U-arborescent représentant f on échange les deux entrées de chaque opérateur en prenant soin de complémenter les entrées éventuellement constantes, on obtient un réseau U-arborescent de la duale de f.

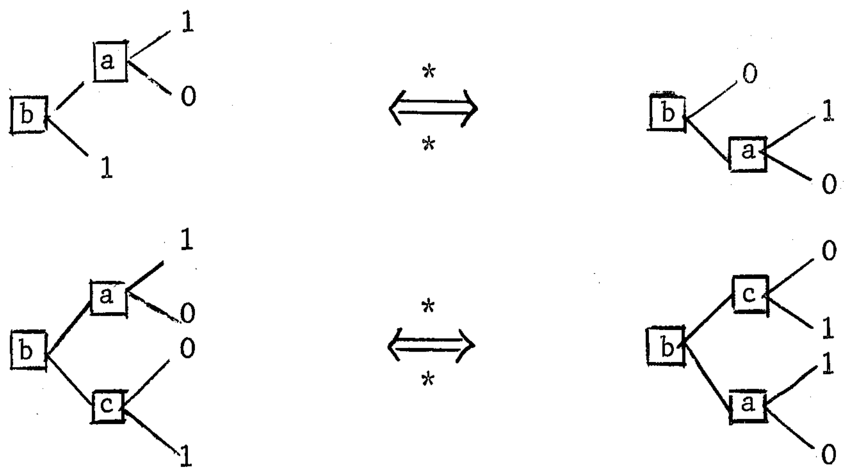
Ceci se déduit du théorème précédent. En effet, on agit de même.

M = 1

Les variables de bout de branche sont conservées

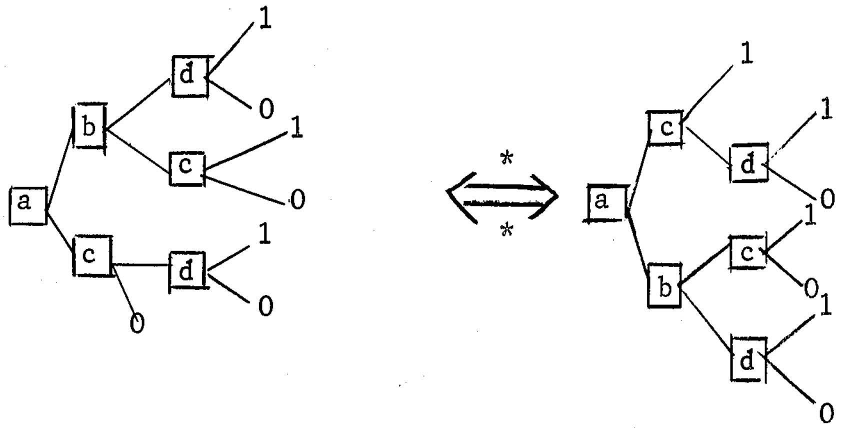
M = 2





pour 2 comme pour $\forall M$ il en est de même que pour les réseaux $\mathcal{U}$.

Exemple



$$f^* = ac + ac'd + a'bc + a'b'd$$

3.b.3.- Sur les réseaux lexicographiques

COROLLAIRE 2

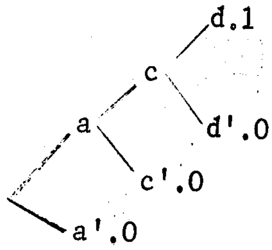
Si dans un réseau lexicographique représentant f on échange les deux entrées de chaque bifurcation en prenant soin de complémenter l'entrée éventuellement constante on obtient un réseau lexicographique de la duale de f.

Remarque

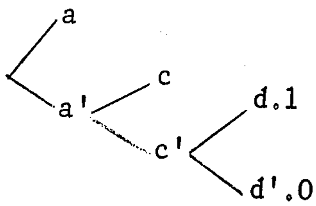
Il faut considérer chaque adjonction comme une bifurcation dont une entrée est nulle.

Exemple

a — c — d s'écrit aussi

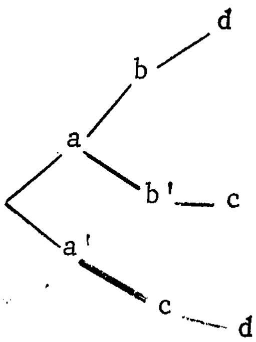


dont le dual est

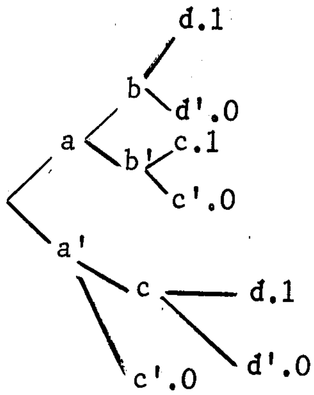


$$a+c+d = a + a'c + a'cd$$

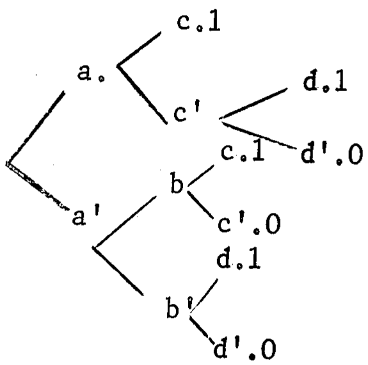
Exemple



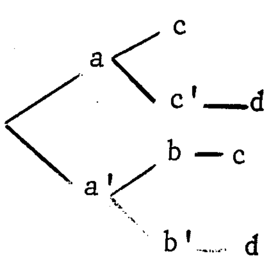
s'écrit



dont le dual est :



ou encore



$$f^* = ac + ac'd + a'bc + a'b'd$$

On remarquera que le calcul de la duale par le moyen habituel sur la forme polynômiale lexicographique ne donne pas une forme lexicographique, les simplifications perturbent la lexicographicité.

Exemple :

$$f = abd + ab'c + a'cd$$

$$f^* = ac + ad + bc + b'd$$

propriétés des réseaux U.

Une fonction, son complément, sa duale ont des représentations U-arborescentes de même coût.

On passe de l'une à l'autre sans changer le nombre d'opérateurs.

3.c.- Structure des Ecritures lexicographiques dont les variables générales n'ont pas d'argument commun.

3.c.1.- Somme

Soient $\Phi(Y)$ et $\psi(Z)$ deux écritures lexicographiques locales où Y et Z n'ont pas d'argument commun.

On définit la somme de ϕ et ψ notée $\pm$ par

$$\Phi(Y) \pm \psi(Z) \underset{\text{déf}}{=} \Phi(Y) + \Phi'(Y) \cdot \psi(Z)$$

dans l'expression de droite les opérateurs + et . sont les opérateurs booléens habituels et $\phi'(Y)$ le complément de $\phi(Y)$.

Exemple

$\Phi(Y) = abd + ab'c + a'cd$

$\phi'(Y) = abd' + ab'c' + a'cd' + a'c'$ cf.§3.a.1.b.

$\psi(Z) = e + e'f$

$$\Phi(Y) + \emptyset'(Y) \cdot \psi(Z) = abd + abd'e + abd'e'f + ab'c + ab'c'e + ab'c'e'f + a'cd + a'cd'e + a'cd'e'f + a'c'e + a'c'e'f$$

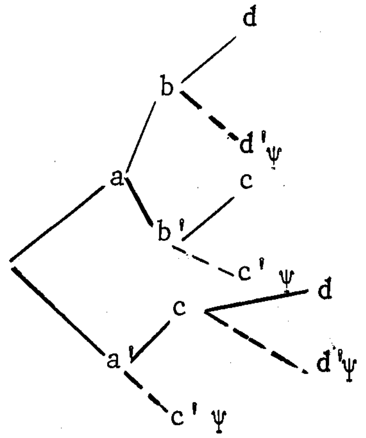
Notons que

$\psi(Z) + \phi(Y) = e + e'f + e'f'\ abd + e'f'ab'c + e'f'a'cd$

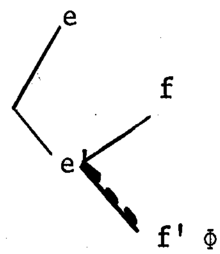
ce qui démontre la non-commutativité.

0 et 1 jouent leur rôle habituel. 0 sera l'élément neutre du monoïde et 1 l'élément spécial.

Sur les réseaux lexicographiques d'écriture <u>irréductibles</u> la somme de 2 réseaux s'obtient facilement.



ou



<u>Propriétés de la somme</u>

La somme de deux écritures lexicographiques locales (respt. irréductibles) sans variable commune est encore une écriture lexicographique locale (respt. irréductible).

En effet si

$$\phi\,(Y) = \sum_{i\in I} \mu_i$$

$$\psi\,(Z) = \sum_{j\in J} \lambda_j$$

$$\phi(Y) \underline{+} \psi(Z) = \sum_{i\in I} \mu + \sum_{i\times J} \mu'_i \, \lambda_j$$

Tous les monômes de cette somme sont disjoints. En effet, les μ_i le sont entre eux, les λ_j le sont entre eux, les $\mu'_i\ \lambda_j$ aussi , les μ_i et les $\mu'_i\,.\,\lambda_j$ le sont aussi, ce qui assure l'irréductibilité.

$$(\phi \underline{+} \psi)' \underline{+} \xi = \phi + \phi'\psi + (\phi+\phi'\psi)'\ \xi$$

$$= \phi + \phi'\psi + (\phi+\phi'\psi')\ \xi$$

$$= \phi + \phi\ \xi + \phi'\psi + \phi'\psi'\xi$$

$$= \phi + \phi'(\ \psi + \psi'\ \xi)$$

$$= \phi \pm (\ \psi \underline{+}\xi)$$

nous avons donc l'associativité.

La somme donne donc une structure de <u>Monoïde</u>.

3.c.2. <u>Produit de deux écritures n'ayant aucunevariable commune.</u>

Le produit de ϕ par Ψ noté $\phi \underline{X} \Psi$ s'obtient en concaténant chauqe monôme de ϕ avec chacun des monômes de Ψ. C'est le produit booléen normal puisqu'aucune simplification n'intervient, chaque écriture n'ayant aucune variable en commun.

<u>Propriétés du produit</u>

- il n'est pas commutatif
  l'écriture $\phi \underline{X} \Psi$ n'est pas égale à $\Psi \underline{X} \phi$, l'ordre lexicographique n'étant pas le même.
- il est interne
- il est associatif puisqu'il s'agit du produit booléen

$$\phi \underline{X} (\Psi \underline{X} \varphi) = (\phi \underline{X} \Psi) \underline{X} \varphi$$

- il possède un élément neutre, le 1 booléen habituel
- aucun élément différent de l'élément neutre n'a d'inverse car on ne pourrait faire le produit de deux écritures ayant les mêmes variables.
- il n'est pas distributif par rapport à la somme comme le montre l'exemple suivant

$$a \underline{X} (b \underline{+} c) = a \underline{X} (b + b'c)$$

$$= ab + ab'c$$

$$(a \underline{X} b) \underline{+} (a \underline{X} c) = ab \underline{+} ac$$

le complément de l'écriture ab étant

$$a' + b'$$

nous avons

$$ab + (a' + b')\, ac = ab + b'ac$$

l'ordre lexicographique n'est plus respecté.

Le produit donne donc une structure de <u>monoïde</u> 1 en est l'élément neutre, 0 l'élément spécial. L'ensemble des formes lexicographiques (restp. irréductibles) possède par rapport des opérations $\underline{+}$ et $\underline{X}$ une double structure de monoïde.

3.d.- Structures des écritures lexicographiques relatives à un ordre donné

Si l'on fait la somme et le produit booléens on obtient une écriture ayant l'ordre donné.

L'addition est interne
associative
commutative
idempotente

il n'y a pas d'élément neutre ayant l'ordre donné.

Le produit est interne
associatif
distributif par rapport à l'addition.

---

(1) Si l'on ne se permet aucune simplification booléenne (allègement, réduction ou par consensus) on a un gerbier d'écritures non irréductibles.

(2) Par contre si l'on se permet toutes les simplifications et si on les fait toutes on a un pseudo treillis des écritures irréductibles car $\phi + \phi\,\psi = \phi + \psi\phi = \phi$ (absorption).

Exemple

$$\phi = abd + ab'c' + a'dc'g + a'dc$$

$$\psi = ab'c'f + a'dc' + a'd'e'$$

(1) $$\begin{cases} \phi + \psi = abd + ab'c' + ab'c'f + a'dc' + a'dc'g + a'd'e' + a'dc \\ \phi\,\psi = ab'c'f + a'dc'g \end{cases}$$

(2) $$\begin{cases} \phi + \psi = abd + ab'c' + a'd + a'd'e' \\ \phi\,\psi = ab'c'f + a'dc'g \end{cases}$$

Tous les algorithmes qui suivent travaillent sur des écritures lexicographiques locales irréductibles pour des questions de rédaction le mot irréductible ne figure pas toujours.

C H A P I T R E   II

OPTIMISATIONS DES FORMES LEXICOGRAPHIQUES

1 - Définitions diverses

2 - Evaluation des différents coûts d'une écriture lexicographique

1 - DEFINITIONS DIVERSES

1.a. Différents coûts relatifs aux écritures lexicographqieus locales irréductibles.

Nous en établirons trois qui sont :

- le coût en lettres L sera le nombre de variables utilisées pour une écriture sous forme polynomiale en somme de produits

Exemple

edcba + edc' + ed'a L = 11

- les deux coûts U-arborescents A et α seront les nombres d'opérateurs U et $\mathcal{U}$ des réseaux U-arborescents associés à la forme lexicographique.

Notons qu'à chaque écriture lexicographique locale irréductible il ne correspond qu'un seul réseau arborescent associé à celle-ci.

1.b. On appellera forme lexicographique locale C-optimale d'une fonction f(X) une forme lexicographique locale irréductible de f(X) réalisant le plus petit coût C.

Remarque.

1.- Il y a toujours une telle forme optimale quels que soient la fonction et le coût, le minimum de ce dernier existant toujours.

2.- La forme lexicographique locale C-optimale peut ne pas être unique.

Propriétés.

Le réseau lexicographique d'une forme lexicographique C-optimale est tel que

- tout sous arbre obtenu en ne prenant qu'une entrée d'une bifurcation quelconque est optimal sinon le réseau ne le serait pas.

- tout autre sous arbre peut ne pas être optimal.

Exemple

f = abc'd' + cde'f' + a'b'ef.

E = adf'e'c + ad'c'b + a'f'e'dc + a'feb'.

$$\left\langle\begin{array}{l} a \left\langle \begin{array}{l} d\,f'\,e'\,c \\ d'\,c'\,b \end{array}\right. \\ a' \left\langle \begin{array}{l} f\,e\,b' \\ f'\,e'\,dc \end{array}\right. \end{array}\right.$$

$\left\langle \begin{array}{l} f\,e\,b' \\ f'\,e'\,d\,c \end{array}\right.$ est optimal mais $\left\langle \begin{array}{l} a\,df'\,e'\,c \\ a'\,f'\,e'\,cd \end{array}\right.$ ne l'est pas

car il s'écrit df'e'c à une permutation près des variables.

2. EVALUATION DES DIFFERENTS COUTS D'UNE ECRITURE LEXICOGRAPHIQUE

Quels que soient la méthode adoptée et le coût arborescent cherché il est bon de regrouper d'abord les monômes ayant le plus grand nombre de variables de tête consécutives communes. C'est d'ailleurs ce que nous faisons habituellement.

2.a. Réseaux $\mathcal{U}$-Evaluation de $\mathcal{A}$

$\mathcal{A}$ est le nombre d'opérateurs $\mathcal{U}$, donc le nombre de lettres de commande, à chaque opérateur, il correspond en effet une lettre de commande et réciproquement.

$$\boxed{\mathcal{A} \text{ est égal au nombre de lettres de commande}}$$

nous avons besoin de deux définitions.

(α) On appelle <u>variable terminale de 1ère catégorie d'un monôme</u>, une variable x_i telle qu'existent :

- le monôme $x_{j_1} \; x_{j_2} \dots x_{j_k} \;\; x_i$ où $\forall_s \;\; j_s \neq i$
- au moins un monôme commençant par

$$x_{j_1} \; x_{j_2} \dots x_{j_k} \; x'_i$$

<u>Remarque</u>. La lettre qui correspond aux variables x_i et x'_i est ainsi lettre de commande d'un opérateur.

On appelle <u>variable terminale de 2ème catégorie d'un monôme</u>, sa variable terminale si elle n'est pas de 1ère catégorie.

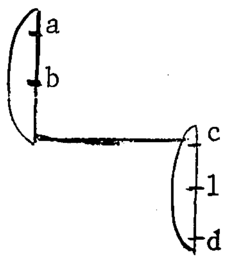
Les lettres correspondant à ces variables ne seront jamais des commandes d'opérateur.

<u>Exemple</u> : $E = ab + a'c + a'c'd$

les variables terminales sont b, c, d

c est de 1ère catégorie, b et d de 2ème.

Voyons le réseau $\mathcal{U}$:



\+ les variables de 2ème catégorie ne compterons jamais, on appelle E_1 l'écriture obtenue à partir de E en supprimant dans les monômes leur éventuelle variable terminale de 2ème catégorie.

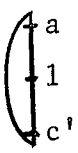
$$\mathcal{A}_E = \mathcal{A}_{E_1}$$

dans E_1 il y a encore les variables de 1ère catégorie de E.

++ Considérons dans E_1 ces variables terminales; elles apparaissent dans le réseau $\mathcal{U}$ comme une entrée égale à 1. On ne peut toutefois la remplacer par 0 et donc supprimer les monômes contenant ces variables en effet l'exemple précédent donnerait

$$a + a'\,c'$$

dont la représentation est



de coût 1 alors que le coût cherché est 2.

Nous proposons plusieurs méthodes de deux types

- à partir du réseau lexicographique local de E_1
- à partir d'une écriture par ligne de E_1

2.a.1. - <u>A partir du réseau représentant</u> E_1

2.a.1.a.

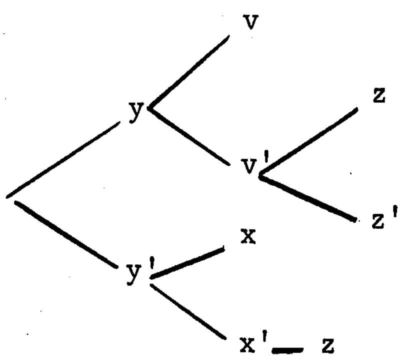
$\mathcal{A}$ est égal au nombre de bifurcations (y compris les adjonctions cas particuliers des bifurcations) autrement dit

$\mathcal{A}$ est égal au nombre de figures $<^{\alpha}_{\alpha'}$ augmenté du nombre de variables n'entrant pas dans cette figure.

<u>Exemple I.</u>

$$E = y\,v + y\,v'\,z\,u + y\,v'\,z'\,x + y'\,x'\,z\,u + y'\,x\,u'$$

$$E_1 = y\,v + y\,v'\,z + y\,v'\,z' + y'\,x'\,z + y'\,x$$



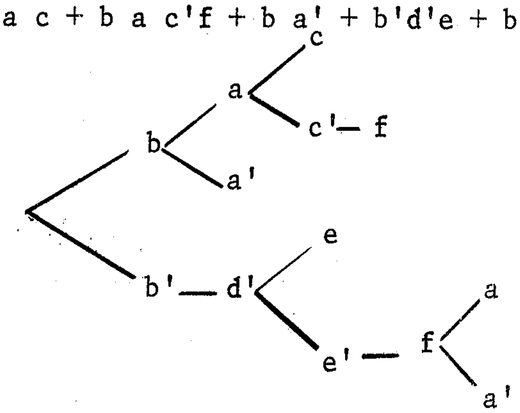
$\mathcal{A} = 2 + 3 = 5$

à chaque bifurcation correspond un et un seul opérateur $\mathcal{U}$ et réciproquement d'où le résultat.

<u>Exemple II.</u>

$$E = b\,a\,c + b\,a\,c'\,f\,d + b\,a'\,d' + b'\,d'e + b'd'e'\,f\,a + b'd'e'f\,a'\,c$$

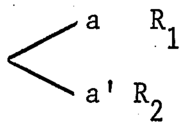
$$E_1 = b\,a\,c + b\,a\,c'f + b\,a' + b'd'e + b'd'e'f\,a + b'd'e'f\,a'$$



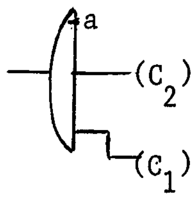
$\mathcal{A} = 5 + 3 = 8$

2.a.1.b. <u>Evaluation récursive du coût</u>

<u>Définition</u>. On appelle coût d'une bifurcation la somme des coûts de ses entrées augmentée de 1; si nous avons



où R_1 et R_2 sont deux sous réseaux lexicographiques de coûts C_1 et C_2 le coût total correspond bien à $C_1 + C_2 + 1$ c'est-à-dire un opérateur $\mathcal{U}$ de plus



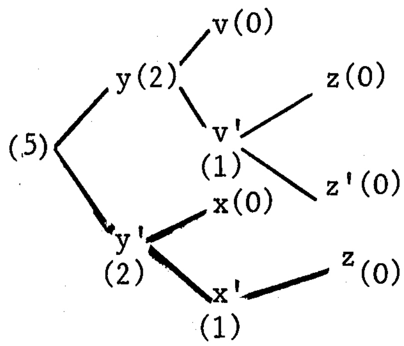
Si l'on attribue aux variables les plus à droite le coût 0 (0 opérateur pour les réaliser), on est capable de tout calculer

> $\mathcal{A}$ est égal au coût de la bifurcation la plus à gauche (racine de l'arborescence).

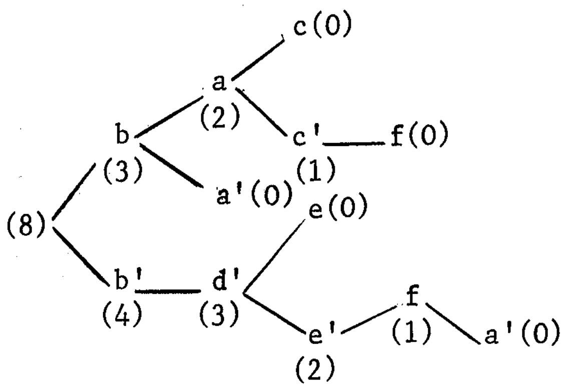
N.B. *Le coût d'une adjonction est évidemment le coût de + l'entrée augmentée de 1.*

<u>Remarque</u>. Cette méthode est effectivement utilisée dans les programmes qui suivent.

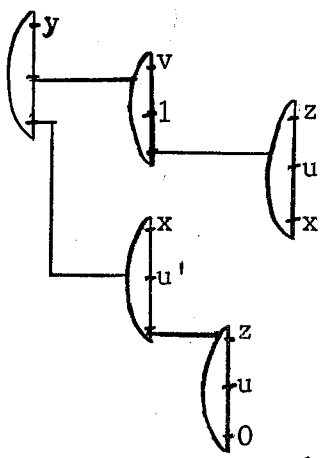
Exemple I

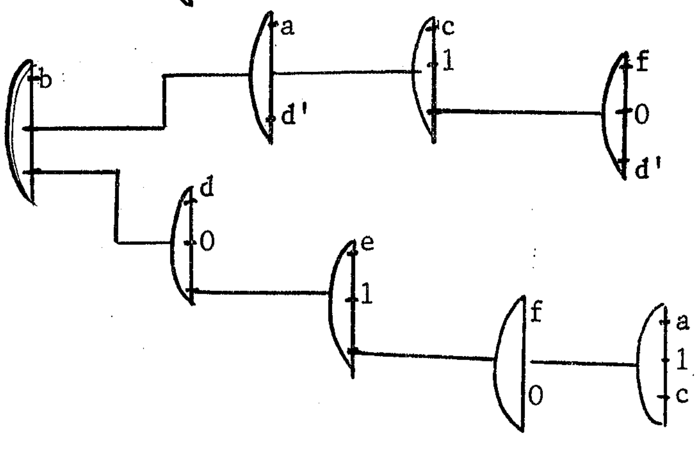


Exemple II



d'où les réseaux





2.a.2. - A partir de l'écriture lexicographique E_1 de E

2.a.2.a. Ecriture par ligne

On considère E_1; chaque monôme de E_1 est écrit sur une ligne différente de telle manière que les deux monômes

$$x_1\ x_2\ x_3\ \ldots\ x_k\ x_{k+1}\ x_{k+2}\ \ldots\ x_m$$

et $$x_1\ x_2\ x_3\ \quad x_k\ x'_{k+1}\ x_{k_{1+2}}\ \ldots\ x_n$$

soient écrits

$$x_1\ x_2\ x_3\ \ldots\ x_k\ x_{k+1}\ x_{k+2}\ \ldots\ x_m$$

et $$x_1\ x_2\ x_3\ \quad x_k\ x'_{k+1}\ \qquad x_{k_{1+2}}\ \ldots\ x_n$$

le premier sur m colonnes, le second sur n+m-k-1.

$\boxed{\alpha \text{ est égal au nombre total de colonnes}}$

En effet, le nombre de colonnes est exactement le nombre de variables de commande car deux variables opposées d'une même bifurcation sont l'une en dessous de l'autre et les variables terminales de 1ère et de 2ème catégorie sont absentes.

Exemple I

```
y v
y v' z
y v' z'
y'.  . x
y'.  . x' z
-----------
1 2  3 4  5
```

$\alpha = 5$

Exemple II

```
b a c
b a c' f
b a'
b'. .  . d' e
b'. .  . d'e'f a
b'. .  . d'e'f a'
-----------------
1 2 3  4 5 6 7 8
```

$\alpha = 8$

2.a.2.b. Factorisation

Mettons en facteurs dans E_1 à l'aide de parenthèses le plus possible de variables entre deux ou plusieurs monômes afin qu'il n'y ait pas de somme de la forme

$$a\ b + a\ b'\varphi \qquad \text{mais} \qquad a(b + b'\ \varphi)$$

De plus l'expression elle même sera mise entre parenthèses. Dans l'expression obtenue nous parenthèsons totalement les produits booléens. Nous disons:

$\mathcal{A}$ est alors le nombre de parenthèses ouvrantes (ou gauches)

Exemple I

* Mise en facteurs et entre parenthèses :

$$(y(v + v'(z+z')) + y'(x + x'z))$$

** Ouverture supplémentaire de parenthèses :

$$(y(v + v'(z + z')) + y'\ (x + x'(z)))$$

1 2 3 4 5

$$\mathcal{A} = 5$$

Exemple II

* $(b(a\ c + ac'f + a') + b'd'(e + e'f\ a + e'f\ a'))$

** $(b(a(c + c'(f))+ a') + b'(d'(e + e'(f(a + a')))))$ $\quad \mathcal{A} = 8$

1 2 3 4 5 6 7 8

On peut aussi affecter à chaque parenthèse ouvrante (fermante) un nombre entier croissant à partir de 1, le 1 correspondant à la parenthèse ouvrante (fermante) la plus à gauche. $\mathcal{A}$ est égal au nombre affecté à la parenthèse ouvrante (fermante) la plus à droite.

Justification (cf. § 4.b.a.1.)

A est le nombre de bifurcation et d'adjonction. A chacune de ces opérations correspond une et une seule paire de parenthèses et réciproquement.

En effet, soit

$$(a\ \mu + a'\ \nu) \quad \text{une parenthèse}$$

ou bien

- μ et ν sont vides et la parenthèse (a + a') donne bien une bifurcation il n'y en a pas d'autres.
- μ ou ν ne sont pas vides tous les deux
  - μ pas vide, ν vide μ est alors dans au moins une parenthèse et il vient

    $$(a(\mu) + a')$$

  - μ et ν ne sont pas vides, il vient :

    $$(a\ (\mu) + a'\ (\nu)\ )$$

la grande parenthèse correspond bien à une bifurcation sur la lettre a.
Si μ ou ν est nul, on a une adjonction.

Soit (a)   soit (a (μ)).

Remarque.

Obtention de $\mathfrak{A}$ à partir de E par factorisation et parenthèsage précédents sauf pour lesproduits avec des variables terminales de 2ème catégorie.

Exemple I

$$\begin{array}{l} (y\ \ (v + v'(z\ u + z'x)) + y'\ \ (x\ u' + x'(z\ u)) \\ 1\ \ 2\qquad\ 3 \qquad\qquad\qquad 4 \qquad\qquad 5 \qquad = \mathfrak{A} \end{array}$$

Exemple II

$$\begin{array}{l} (b\ \ (a\ \ (c + c'(f\ d) + a'd') + b'(d'(e + e'(\ f(\ a + a'c\ ))))) \\ 1\ \ 2\ \ \ 3\ \ \ \ 4 \qquad\qquad\qquad\qquad\qquad 5\ 6 \qquad 7\ \ 8 \qquad\quad = \mathfrak{A} \end{array}$$

2.a.2.c. - Une dernière méthode.

Ecrivons les monômes de l'écriture E_1 les uns en dessous des autres.

Regroupons ceux qui ont le plus grand nombre de variables identiques en tête si ce n'est déjà fait.

La bifurcation de tête se trouve forcément dans la lère colonne. Tirons un trait horizontal séparant les variables directes des accentuées. Les bifurcations suivantes seront des colonnes non coupées par le trait de bifurcation précédent, et comprenant une seule lettre sous les deux formes, directe et accentuée. Traçons encore le trait de bifurcation. Ainsi de suite des bifurcations seront les colonnes suivantes comprenant la même lettre sous les deux formes, non coupées par les traits les bifurcations précédentes.

En effet

b	a	c
b	a'	
b'	a'	d
b'	a	f

La colonne (b,b,b'b') est une bifurcation mais (a,a',a',a) n'en est pas une mais deux distinctes.

Encerclons les bifurcations d'une part, et d'autre part les restes de colonne pas interrompus par des traits de bifurcation -ceux-ci correspondent à des adjonctions - le coût arborescent est alors la somme des cercles.

Exemple I

y	v'	z
y	v'	z'
y'	x	z
y.'	x	

adjonctions

A = 3 + 2 = 5

bifurcations

Exemple II

b	a	c	f'	
b	a'			
b	d'	e	f	a'

adjonctions

A = 2 + 6 = 8

bifurcations

On a bien A puisque la somme des bifurcations (et des adjonctions)

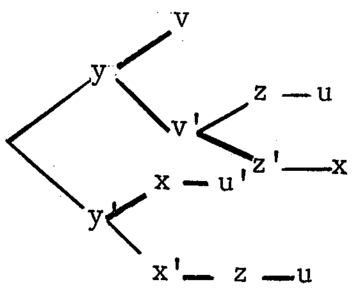
2.b. - Réseaux U-évaluation de A.

Nous retrouvons les deux types de point de départ. Nous partons de E car toutes les variables comptent.

2.b.1. - Réseaux lexicographique de E.

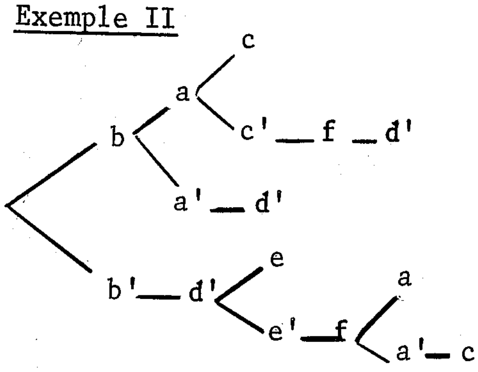
2.b.1.a. - A est égal au nombre de bifurcations (y compris toutes les adjonctions cas particuliers des bifurcations) cf.§ 2.a.1.a.

Exemple I



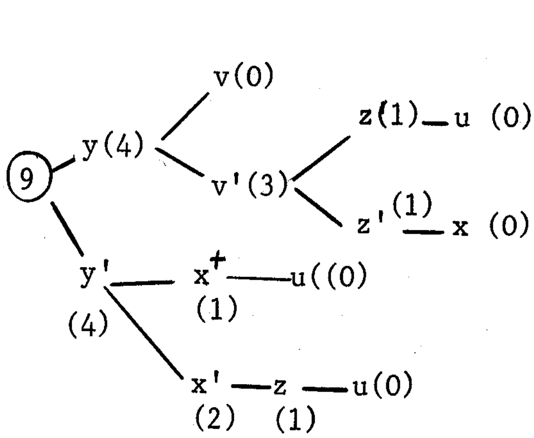
A = 4 + 5 = 9

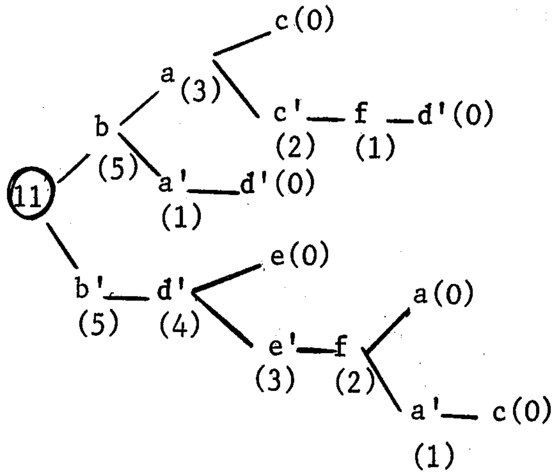
Exemple II



A = 5 + 6 = 11

2.b.1.b. - D'une manière récursive cf. § 2.a.1.b.





2.b.2. - Ecriture lexicographique de E.

2.b.2.a. - Ecriture par ligne cf [9] p.15, cf. § 2.a.2.a.

Exemple I

```
y v
y v'z u
y v'z'. x
y'. . . . x'z u
y'. . . . x . . u'
------------------
1 2 3 4 5 6 7 8 9
```

A = 9

Exemple II

```
b a c
b a c'f d
b a'. . . . d'
b'. . . . . d'e
b'. . . . . d'e'f a
b'. . . . . d'e'f a' c
----------------------------
1 2 3 4 5 6 7 8 9 10 11
```

A = 11

2.b.2.b. - Factorisation cf. § 2.a.2.b.

On fait la même chose que précédemment mais avec E.

Exemple I

(y (v + v'(z u + z'x)) + y'(x' z u + x u'))

(y (v + v'(z(u) + z'(x))) + y'(x'(z(u)) + x(x')))

A = 9

Exemple II

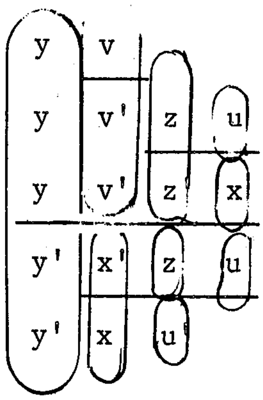
(b(a(c + c'f d') + a'd') + b'd' (e + e'f(a + a'c)))

(b(a(c + c'(f(d'))) + a'd') + b'(d'(e+e'(f(a + a'(c))))))

A = 11

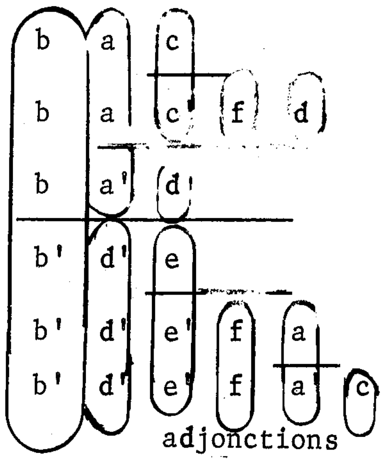
2.b.2.c. - On prend E et on applique la même méthode cf.§2.a.2.c.

Exemple I



adjonctions
A = 4 + 5 = 9
bifurcations

Exemple II



adjonctions
A = 5 + 6 = 11
bifurcations

2.c. - Correspondance des divers coûts d'une même écriture

2.c.1. - Expression liant A et $\mathcal{A}$ pour une écriture E(X)

A et $\mathcal{A}$ ne diffèrent que par le nombre de variables terminales de 2ème catégorie (cf. 2.a.1.)

$$\boxed{A = \mathcal{A} + \mathrm{card}(K).}$$

Si l'on appelle K l'ensemble de toutes ces variables de 2ème catégorie ce qui s'énonce :

> Le coût arborescent A est égal au coût arborescent $\mathcal{A}$ augmenté du nombre de variables terminales de 2ème catégorie de monômes

2.c.2. - Expression de A en fonction de L d'une écriture E(X)

Le gain du coût A par rapport à L provient du fait qu'à chaque bifurcation,

$$a\, g_1 + a' g_2$$

le coût A est augmenté de 1 quelles que soient g_1 et g_2 tandis que L s'accroît de la somme des nombres de monômes de g_1 et de g_2

Le gain à chaque bifurcation est égal à ce nombre de monômes diminué de 1.

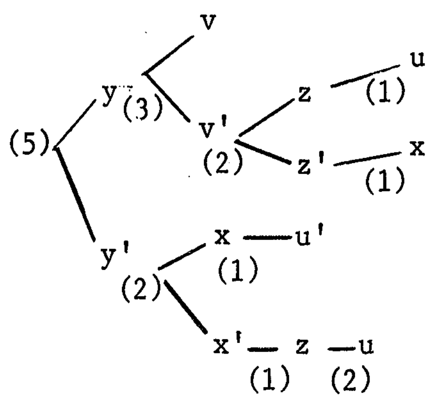
2.c.2.a. Si l'on appelle b_i le nombre effectif de monômes entrant dans la bifurcation i on voit que ce nombre est la somme des nombres de monômes des bifurcations conduisant à ses entrées.

En donnant à 1 le nombre 0 la variable terminale de chaque monôme a un nombre égal à 1 le gain est égal à $b_i - 1$

$$\boxed{A = L - \sum_{i \in I} (b_i - 1)}$$

où card(I) est égal au nombre de bifurcations y compris les adjonctions, chaque bifurcation étant indicée par I.

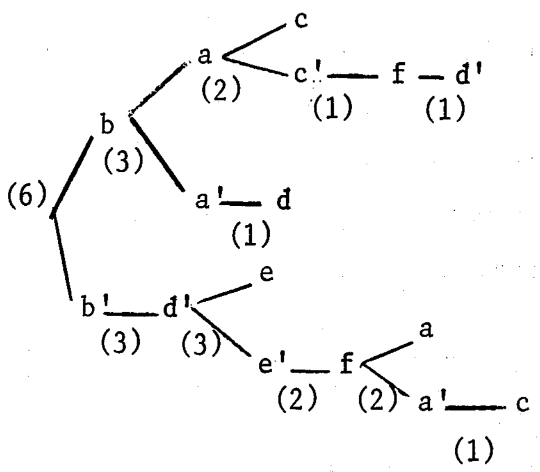
<u>Exemple I</u> - précédent



le nombre entre parenthèses est le b_i.

$$A = 17 - (4+2+1+0+0+1+0+0+0) = 17 - 8 = 9$$

Exemple II



$$A = 25 - (5 +2+1+0+0+0+2+2+1+1+0) = 25 - 14 = 11$$

2.c.2.b. - Autre formulation

On peut ne s'intéresser qu'aux bifurcations a g_1 + a'g_2 où g_1 et g_2 ne sont pas nulles et affecter à chacune un coefficient a_j égal au nombre de variables se trouvant à gauche d'elle même, n'entrant pas dans une autre véritable bifurcation.

Il vient alors

$$\boxed{A = L - \sum_{j \in K} (a_j + 1)\,(b_j - 1)}$$

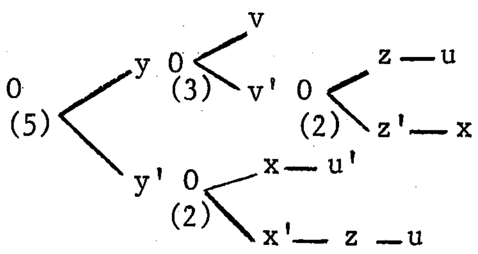
où $K \subseteq I$ représente l'indiçage des seules véritables bifurcations.

En effet, dans la formule du §2.c.2.a. nous répétons le nombre de monômes trouvé à une bifurcation autant de fois qu'il y a de bifurcation à une entrée nulle soit a_j fois donc à chacune de ces opérations on a un gain de

$$(a_j + 1)\,(b_j - 1)$$

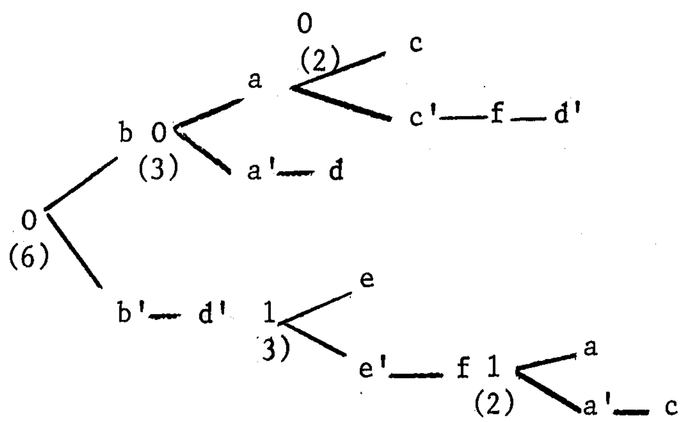
Exemple I

a_j est le nombre placé au dessus de b_j



A = 17 - (4+2+1+1) 1 = 9

Exemple II



A = 25 - (5x1+2x1+1x1+2x2+1x2) = 25 - 14 = 11

CHAPITRE 3

ALGORITHME DE RECHERCHE DES FORMES LEXICOGRAPHIQUES LOCALES IRREDUCTIBLES OPTIMALES DES FONCTIONS INCOMPLETES.

1. - Principe et mise en oeuvre
2. - Aspect pratique
3. - Autre manière d'obtenir une solution optimale
4. - Simplifications préliminaires
5. - Exemple traité par programme

1. PRINCIPE ET MISE EN OEUVRE

1.a. Remarque sur les écriture lexicographiques.

Considérons une écriture lexicographique locale irréductible optimale $E(X)$ d'une fonction incomplète donnée par $(\underline{f}(X), \overline{f}(X))$ où X est une variable générale. Prenons une tête de monôme m figurant dans un terme de $E(X)$ $(m \neq 0)$. Nous pouvons donner les résultats suivants :

- 1°/ L'ensemble des termes de $E(X)$ commençant par m est une écriture lexicographique locale irréductible optimale de $(m\,\underline{f}(X), m\,\overline{f}(X))$

  En effet, il n'y a dans $E(X)$ que deux sortes de monômes, d'après les propriétés des monômes des formes lexicographiques locales tous les monômes de $E(X)$ ont comme premières variables
  - soit un certain nombre de variables de m (il y en a au moins une)
  - soit effectivement toutes les variables de m et dans l'ordre.

  de $\underline{f}(X) \leqslant E(X) \leqslant \overline{f}(X)$

  nous tirons $m\underline{f}(X) \leqslant m\,E(X) \leqslant m\,\overline{f}(X)$

  Si l'on appelle $g(Y)$ la somme des restes des monômes de la 2ème catégorie où l'on a enlevé m, Y étant la variable générale n'ayant aucune lettre de m, l'inégalité précédente devient

  $$m\,\underline{f}(X) \leqslant m\,g(Y) \leqslant m\,\overline{f}(X)$$

  les monômes de la 1ère catégorie ont tous un produit nul avec m.

- 2°/ $g(Y)$ est une écriture lexicographique locale irréductible optimale de la fonction booléenne complète qu'elle représente.

  Si elle ne l'était pas $E(X)$ ne le serait pas non plus.

— Inversement si l'on a

$$m\,\underline{f}(X) \leqslant m\,g(Y) \leqslant m\,\overline{f}(X) \qquad m \neq 0$$

Il y a une écriture lexicographique locale de ([illegible] $\underline{f}(X)$, [illegible] $\overline{f}(X)$) qui comprend les termes $m\,g(Y)$, si l'on choisit $g(Y)$ lexicographique locale irréductible optimale. Remarquons que rien n'assure l'optimalité pour $(\underline{f}(X), \overline{f}(X))$.

1.b. - Principe de l'Algorithme

Il nous faut obtenir une forme lexicographique locale irréductible optimale de la fonction incomplète donnée par $(\underline{f}, \overline{f})$ à partir d'écritures lexicographiques locales optimales irréductibles g telles qu'existe m vérifiant

$$(1) \qquad m\,\underline{f} \leqslant m\,g \leqslant m\,\overline{f} \qquad m \neq 0$$

Propriété

$$\boxed{(1) \text{ est équivalent à } (2) \quad m \leqslant \overline{f}\,g + \underline{f}'\,g' \quad m \neq 0}$$

- . (1) → (2)

  - * $m\,\underline{f} \leqslant m\,g$ donne

$$m' + g' = (m' + \underline{f}')\,(m' + g') = m' + \underline{f}'g'$$

soit en multipliant par m après avoir effectué

(α) $\quad m\,g' = m\,\underline{f}'\,g'$

-** $\quad m\,g \leqslant m\,\overline{f}$ donne

(β) $\quad m\,g = m\,\overline{f}\,.\,m\,g = m\,\overline{f}\,g$

en faisant l'addition membre à membre des égalités (α) et (β)

(γ) $\quad m = m(\overline{f}\,g + \underline{f}'\,g')$

ou encore $\quad \boxed{m \leqslant \overline{f}\,g + \underline{f}'\,g'}$

- .. (2) → (1)

  - * (γ) donne

$$m' = m' + (\overline{f}' + g')\,(\underline{f} + g) = m' + \overline{f}'\,\underline{f} + \overline{f}'\,g + \underline{f}\,g'$$

ajoutons dans chaque membre $\underline{f}'$

$$m' + \underline{f}' = m' + \overline{f}'\,\underline{f} + \overline{f}'\,g + \underline{f}\,g' + \underline{f}'$$

par consensus avec $\underline{f}'$, $\underline{f}\,g'$ donne g'

de même avec g', $\overline{f}'\,g$ donne $\overline{f}'$

il vient alors

$$m' + \underline{f}' = m' + g' + \underline{f}'$$

équivaut à $\quad \boxed{m\,\underline{f} \leqslant m\,g}$

- * (γ) donne

$$m\,g = m\,\overline{f}\,g = m\,\overline{f}\,.\,m\,g$$

donc

$$\boxed{m\,g \leqslant m\,\overline{f}}$$

ce qui achève la démonstration de la propriété. Nous avons donc

$$(2') \qquad 0 < m \leqslant \overline{f}\,g + \underline{f}'\,g'$$

si m = 0, toutes les inéquations (1) sont vérifiées quelles que soient g, ces écritures ne peuvent donc convenir.
Nous voyons que le produit

$$\overline{f}\,g + \underline{f}'\,g'$$

va établir la validité d'une écriture g - g ne sera sélectionnée que si ce produit n'est pas nul.

1.c. - Mise en oeuvre

L'algorithme va consister à construire d'une manière exhaustive toutes les écritures g(Y) à 1,2,3,...,n variables à l'aide de l'opération bifurcation à partir des constantes 0 et 1 d'abord, ensuite des g(Y) sélectionnées auparavant. Ayant obtenu deux écritures $g_1(Y)$ et $g_2(Y)$ on forme les écritures

$$G(Y,x) = x\,g_1(Y) + x'\,g_2(Y)$$

où x représente successivement toutes les variables de X dont Y ne dépend pas.

(2) donne $\quad m \leqslant \overline{f}\,G + \underline{f}'\,G'$

$$m \leqslant \overline{f}(x\,g_1 + x'\,g_2) + \underline{f}'(x\,g_1' + x'\,g_2')$$

$$m \leqslant x(\overline{f}\,g_1 + \underline{f}'\,g_1') + x'(\overline{f}\,g_2 + \underline{f}'\,g_2')$$

nous remarquons que m dépend des monômes relatifs aux écritures précédentes g_1 et g_2.

Ecrivons les monômes relatifs à $g_1(Y)$ sous la forme

$$m_1 = h_1(x,\ Z_x) \text{ où } Z_x \text{ dépend des mêmes lettres que } Y,$$

sauf x.

De même

$$m_2 = h_2(x,\ Z_x) \text{ pour } g_2(Y)$$

l'inégalité précédente devient

$$m \leq x\ h_1(x,\ Z_x) + x'\ h_2(x,\ Z_x)$$

cependant, la lettre x apparaît dans $G(Y,x)$ nous ne la voulons pas dans m. Pour celà nous devons mettre en facteur x dans h_1 et x' dans h_2 et prendre le consensus des termes ainsi trouvés par rapport à x et x'

$$m < x\ h_1(1,\ Z_x) + x'\ h_2(0,Z_x)$$

$$\boxed{m \leq h_1(1,Z_x)\ .\ h_2(0,Z_x)} \qquad (3)$$

Il ne nous sera pas nécessaire de calculer le produit

$$\bar{f}(X)\ .\ G(Y,x) + \underline{f}'(X)\ .\ G'(Y,x)$$

pour connaître si $G(Y,x)$ sera sélectionnée ou non mais seulement le produit

$$\boxed{K(Z_x) = h_1(1,Z_x)\ .\ h_2(0,Z_x)}$$

c'est pourquoi nous devons garder pour chaque sélection de $G(Y,x)$ les mônomes m qui lui sont relatifs.

1.d. - Obtention des solutions

Démontrons le théorème

THEOREME

Si pour E on a $h_1(1,Z_x)\ .\ h_2(0,Z_x) = 1 \rightleftarrows$ E est une écriture lexicographique locale irréductible

- 1 $h_1 \cdot h_2 = 1 \rightarrow$ solution compatible (cf. réf 1-c. chap. I)

Démontrons la proposition par l'absurde.

Ecriture non compatible $\rightarrow h_1 \cdot h_2 \neq 1$

Deux cas se présentent à nous :

- a $E < \underline{f} \leqslant \overline{f}$

  ou $E' > \underline{f}' \geqslant \overline{f}'$

  (2) donne $$m \leqslant \overline{f}\, E + \underline{f}'\, E'$$
  $$\leqslant E + \underline{f}' < \underline{f} + \underline{f}' = 1$$

  $$h_1\, h_2 < 1$$

- b $E > \overline{f} > \underline{f}$ ou $E' < \overline{f}' < \underline{f}'$

  (2) donne $$m \leqslant \overline{f} \cdot E + \underline{f}'\; E'$$
  $$\leqslant \overline{f} + E' < \overline{f} + \overline{f}' = 1$$

  $h_1\, h_2 < 1$ ici aussi.

- 2 E compatible $\rightarrow h_1\, h_2 = 1$

  $h_1 \cdot h_2 = \overline{f}\, E + \underline{f}'\, E'$ où $\underline{f} \leqslant E \leqslant \overline{f}$

  $= E + E' = 1$

ce qui achève la démonstration.

En résumé à chaque $G(Y,x)$ construit on fait correspondre le produit $h_1 \cdot h_2$.

Si $h_1 \cdot h_2$ est nul G ne sera pas gardée

Si $h_1 \cdot h_2$ est égal à 1 on aura une solution pour $(\underline{f}, \overline{f})$ pas nécessairement optimale.

Si $h_1\, h_2$ est compris entre 0 et 1 et différent de ces deux quantités on le gardera pour construire d'autre G.

1. e. - Cas particulier des fonctions complètes

f fonction complète peut se caractériser par

$$f = \underline{f} = \overline{f}$$

on obtient la même chose que les fonctions incomplètes en remplaçant

$\overline{f}$ par f et $\underline{f}'$ par f'

2. ASPECT PRATIQUE

2.a. - Réalisation

Nous rangeons dans un tableau les $g(Y)$ successifs ainsi que les monômes m associés. Les deux premières lignes sont seules connues au début.

$g(Y) = 0$ m : $\underline{f}'(X)$

$g(Y) = 1$: $\overline{f}(X)$

l'ensemble de ces deux lignes formera la classe 0.

D'une manière générale nous prendrons pour $g_1(Y)$ les écritures $g(Y)$ déjà obtenues dans l'ordre du tableau et pour $g_2(Y)$ celles antérieures strictement à $g_1(Y)$ que nous combinerons sous toutes les formes possibles $a\, g_1 + a'\, g_2$.

Pour ne pas combiner plusieurs fois les mêmes écritures nous sommes amenés à donner des précisions :

* A partir de la classe 0 on forme donc les $g(Y)$ d'une lettre a, a', b, b'... on doit effectuer les produits des différentes fermetures inférieures de $f(X)$ et de $f'(X)$, ce qui revient à ne prendre que les variables effectivement présentes dans la base complète de f.
* Pour former les écritures de la classe i nous prendrons
  - pour $g_1(Y)$ toutes celles, dans l'ordre du tableau, obtenues dans la classe i - 1.
  - pour $g_2(Y)$ toutes les écritures de toutes les classes inférieures à i - 1 jusqu'à 0 et celles de la classe i - 1 qui dans le tableau sont antérieures à $g_1(Y)$ que l'on vient de choisir.

Nous devons alors connaître

a - de quelles lettres dépendent g_1 et g_2 soient Y_1 et Y_2

b - quelles sont les lettres que les monômes m interdisent pour g_1 et pour g_2.

Nous ne pouvons en effet faire des bifurcations sur g_1 et g_2 que pour des lettres n'entrant ni dans Y_1 ni dans Y_2. De plus si pour g_1, par exemple, les monômes relatifs ont tous la variable a' ou si l'on préfère

$$h_1(a, Z_a) = a' \varphi (Z_a)$$

il sera inutile d'essayer des bifurcations comprenant a g_1 car le produit sera toujours nul (en faisant a = 1, a' est nul), avec ces restrictiontions simples nous possédons une écriture valable dont le test de validité est donné par l'inégalité (3). Si la quantité de droite est nulle c'est terminé on ne garde pas le g(Y) formé. Si elle ne l'est pas, nous devons essayer d'optimiser le tableau (au départ il est optimisé, si on le fait à chaque fois il le restera). Nous prenons le vecteur de vérité du produit, sur l'ensemble Z_a et cherchons tous les vecteurs de vérité - calculés sur le même Z_a

- égaux à lui-même

à chaque fois que nous en avons un nous ôtons du tableau celui des g(Y) correspondants de coût le plus fort (s'il s'agit de celui que nous venons de former nous ne le rangeons pas et passons un autre g(Y). Dans le cas où les coûts sont égaux nous ne gardons que le premier rangé. Nous réitérons la méthode jusqu'à ce qu'il ne soit plus possible d'adjoindre une lettre, c'est-à-dire après la formation de la classe N si N est le nombre d'arguements dont dépend X.

Au pas N nous ne testerons que les solutions éventuelles comme aux pas précédents. Nous n'optimiserons pas et nous ne garderons pas les fonctions puisqu'elles ne serviront à rien.

De plus toute écriture dont le coût est supérieur à celui d'une solution sera abandonnée à la formation. En mettant le coût optimal de la solution à n 2^{n-1} (cf. chap. V § 3.b) au départ, on peut entreprendre l'algorithme.

2. b. - <u>Remarque</u>

Dans la pratique nous trouverons G(Y) sous la forme :

$$G(Y_x) = x\, g_1(Y_1) + x'\, g_2(Y_2)$$

où Y est une variable qui comprend d'une part les arguments de Y_1, de l'autre ceux de Y_2.

Si m_1 dépend de Z_{1x} et x et m_2 de Z_{2x} et x on veut que le produit con-convenable ne dépende que de Z_x avec les notations précédentes. Z_x est d'ailleurs la variable n'ayant que les arguments communs à Z_{1x} et Z_{2x}

En conséquence m_1 et m_2 seront remplacées par leur fermeture inférieure sur Z_x. Particularisons en posant x = a.

Nous devons poser Y_{1a} variable générale dont les arguments sont ceux de Z_{1a} qui ne se retrouvent pas dans Z_a et

$$h_1(a,\ Z_{1a}) = H_1\ (a,\ Y_{1a},\ Z_a)$$

$$h_2(a,\ Z_{2a}) = H_2\ (a,\ Y_{2a},\ Z_a)$$

il vient alors

$$\boxed{m \leqslant \underbrace{H_1\ (1,\ Y_{1a},\ Z_a)}_{Y_{1a}} - \underbrace{H_2\ (0,\ Y_{2a},\ Z_a)}_{Y_{2a}}.} \qquad (4)$$

il faudra calculer le produit $\boxed{\underbrace{H_1}_{Y_{1a}} \cdot \underbrace{H_2}_{Y_{2a}} = K\ (Z_a)}$

<u>Cas particuliers.</u>

1° - Une des 2 g(Y) est nulle

On ne restreint pas le problème en disant que $g_1(Y)$ est nulle.

$$m \leqslant a\ \underline{f}' + a'(\overline{f}\ g_2 + \underline{f}'\ g_2')$$

si l'on pose $\underline{f}'(X) = \varphi\,(a, X_a) = (a, Y_{2a}, Z_{2a})$

$$m_2 = h_2(a, Z_{2a})$$

en reprenant les notations ci-dessus.

m ne dépendra que de Z_{2a}. On ne prendra qu'une seule fermeture inférieure soit

$$\boxed{m \leq \underset{Y_{2a}}{\sqcap}\ \Phi\,(1, Y_{2a}, Z_{2a}) \cdot h_2(0, Z_{2a})} \qquad (4')$$

2° - Une des 2 g(Y) est égale à 1

Soit $g_1(Y) = 1$

$$m \leq a\,\overline{f} + a'\,(\overline{f}\,g_2 + \underline{f}'\,g_2')$$

si l'on pose

$$\overline{f}(X) = \Psi\,(a, X_a) = \Psi\,(a, Y_{2a}, Z_{2a})$$

il vient

$$\boxed{m \leq \underset{Y_{2a}}{\sqcap}\ \Psi\,(1, Y_{2a}, Z_{2a}) \cdot h_2(0, Z_{2a})} \qquad (4'')$$

Prenons un exemple.

Pour montrer l'utilité de ces fermetures inférieures X dépend de a, b, c, d, e,

$\underline{f}(X) = \overline{f}(X) = b'd' + ab' + ce + a'de$

on veut former $a'e + ab'$

à e correspond $\quad bc + a'd \qquad Y_1 = \{e\}$

à b correspond $\quad d'e' + c'd' + ae' + ac' \qquad Y_2 = \{b\}$

$Y_1 \vee Y_1 = \{e, b\}$

$Z_1 \wedge Z_2 = \{a, c, d\}$

$m = \overline{(bc + 1d)}\ \{b\} \quad . \quad \overline{(d'e' + c'd' + 1e' + 1c')}\ \{e\}$

$m = c'd$

3 - AUTRE MANIERE D'OBTENIR UNE SOLUTION OPTIMALE

3. a. Grandes lignes de la méthode (cf. [7] p.61)

Nous pouvons obtenir une écriture lexicographique optimale de $(\underline{f}, \overline{f})$ à partir d'écritures lexicographiques locales irréductibles $\underline{E}$ et $\overline{E}$ de $\underline{f}$ et $\overline{f}$ de la manière suivante :

Ne prendre que les monômes de $\overline{E}$ qui sont des diviseurs de ceux de $\underline{E}$.

Exemple.

$$\overline{E} = abc + abc'd + ab'd + a'b'c'$$

$$\underline{E} = abcd + ab'dc'.$$

d'où l'on tire

$$E = abc + ab'd.$$

Formalisons en posant

$$\underline{E} = \sum_{i \in I} \mu_i, \qquad \forall i \in I, \quad \forall j \in J \qquad \mu_i \cdot \mu_j = 0$$

$$\overline{E} = \sum_{j \in J} \lambda_j, \qquad \forall i \in I, \quad \forall j \in J \qquad \lambda_i \cdot \lambda_j = 0.$$

on définit $K \subset I$ par

$$\forall k \in K, \ \exists j \in J \qquad \lambda_k \cdot \mu_j = \mu_j \ (\text{ou } \lambda_k + \mu_j = \lambda_k)$$

La solution est donnée par

$$E = \sum_{k \in K} \lambda_k$$

Propriété

A chaque monôme μ_i de $\underline{E}$ il correspond un et un seul monôme λ_j de $\overline{E}$ dont il est multiple

(1) A chaque μ_i il correspond un λ_j tel que

$$\mu_i + \lambda_j = \lambda_j$$

pour démontrer cette proposition nous montrons que l'opposé est absurde.

Si existe μ_k tel que

$$j \in J \qquad \mu_k + \lambda_j \neq \lambda_j$$

a Revoir

en faisant la somme sur J il vient

$$\mu_k + \sum_J \lambda_j \neq \sum_J \lambda_j$$

ajoutons à chaque membre $\sum_I \mu_i$

il vient

$$\sum_I \mu_i + \sum_J \lambda_j \neq \sum_I \mu_i + \sum_J \lambda_j$$

ce qui est impossible.

(2) Il est unique

Si $\exists\, \ell \in J$ et $k \in J$ tels que

$$\mu_j \cdot \lambda_\ell = \mu_j$$

$$\mu_j \cdot \lambda_k = \mu_j$$

nous aurons alors en faisant le produit membre à membre

$$\mu_j \cdot \lambda_\ell \cdot \lambda_k = \mu_j = 0$$

car $\forall k \in J, \ \forall \ell \in J, \ \ell \neq k \quad \lambda_\ell \cdot \lambda_k = 0$

ce qui achève la démonstration.

Remarque

On peut définir K par

$$\forall k \in K \quad \begin{cases} \exists j \in J & \lambda_k \cdot \mu_j \neq 0 \\ \forall \ell \neq j,\ \ell \in J & \lambda_\ell \cdot \mu_j = 0 \end{cases}$$

Conséquence

(1) $\boxed{\text{E a au plus le nombre de monômes de } \underline{E} \quad \mathrm{card}(K) \leqslant \mathrm{card}(J)}$

En effet il se peut qu'à deux μ_j et μ_i $i \neq j$ il corresponde le même λ_k.

(2) E est une écriture lexicographique locale irréductible optimale de $(\underline{f}, \overline{f})$

En effet $\quad E \leqslant \overline{E}$

$$\underline{E} + E = \sum_{k \in K} \lambda_k + \sum_{j \in J} \mu_j = \sum_{k \in K} \lambda_k = E$$

d'après la propriété précédente

$$\forall k \in K \quad \begin{cases} \exists j \in J & \lambda_k + \mu_j = \lambda_k \\ \forall \ell \in J,\ \ell \neq j, & \lambda_k + \mu_\ell = \lambda_k \end{cases}$$

nous avons finalement

$$\underline{E} \leqslant E \leqslant \overline{E}$$

3. b. Restrictions à apporter

On ne peut toutefois appliquer cette méthode à deux écritures lexicographiques $\underline{E}$ et $\overline{E}$ ayant des ordres lexicographiques quelconques.

En effet soit

$\overline{f} = d + a\ b\ c \qquad\qquad \overline{E} = d + d'\ a\ b\ c$

$\underline{f} = d\ a\ b\ c' + a\ b\ c \qquad\qquad \underline{E} = a\ b\ c + a\ b\ c'\ d$

si on applique la méthode il vient

$$d$$

or $d \leqslant \overline{E}$ $\quad$ car $\quad \overline{E} = d + d'\ a\ b\ c$

mais d et $\underline{E}$ ne sont pas comparables.

$$d\,\underline{E} = d\,a\,b\,c' + d\,a\,b\,c < d$$

$$d + \underline{E} = d + a\,b\,c = \overline{E}$$

tandis que la solution est

$$c\,a\,b + c'd$$

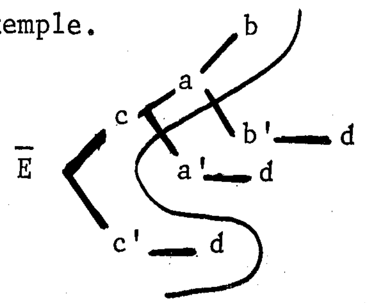
Cette dernière est obtenue par algorithme; on peut la réobtenir par application de la méthode de couverture précédente sur les écritures

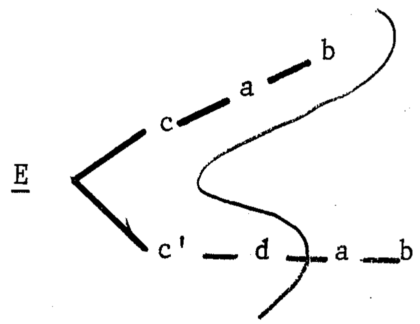
$$\overline{E} = c\,a\,b + c\,a\,b'd + c\,a'd + c'd$$

$$\underline{E} = c'd\,a\,b + c\,a\,b \quad .$$

On peut déjà dire que la méthode précédente s'applique sur des écritures lexicographiques locales optimales ou non de $\underline{E}$ et $\overline{E}$ ayant la même orientation lexicographique locale.

Il est intéressant de voir ce que cette méthode donne sur les réseaux lexicographiques locaux respectifs de $\underline{E}$ et $\overline{E}$ soit sur l'exemple.





On voit que prendre les monômes de $\overline{E}$ qui soient des diviseurs de ceux de $\underline{E}$ revient à faire l'intersection des réseaux c'est-à-dire de ne prendre que la portion commune aux deux réseaux. Dans cette intersection ne figurent que les monômes de $\overline{E}$ qui sont aussi des têtes de monômes de $\underline{E}$.

3. c. - Résultat

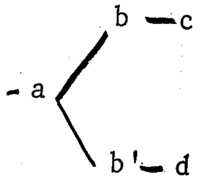
On obtient une écriture lexicographique de $(\underline{f}, \overline{f})$ en prenant l'intersection 2 à 2 des réseaux lexicographiques locaux de $\overline{E}$ et $\underline{E}$ correspondant à la même ordination lexicographique locale et ceci pour toutes les ordinations possibles.

Reprenons l'exemple de 2.a. et écrivons $\underline{f}$ et $\overline{f}$ sous forme de base complète

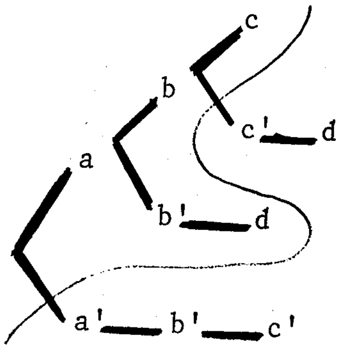
$$\underline{f} = a\ b\ c\ d + a\ b'c'd$$

$$\overline{f} = a\ b\ c + a\ d + a'b'c'$$

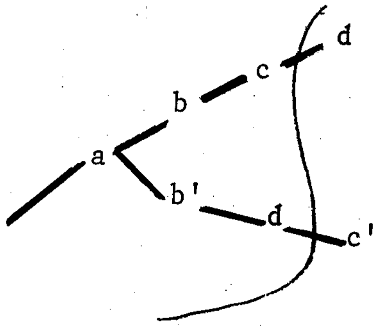
la solution obtenue par programme en 15 secondes est



que l'on peut obtenir par intersection des deux réseaux suivants



et



3. d.-<u>Remarque</u>

Cette méthode est essentiellement intéressante si on s'impose un ordre lexicographique local. En général elle n'a qu'un rôle justificatif après avoir obtenu la solution par l'algorithme précédent. ou par programme. En effet, ayant la solution on peut considérer les écritures lexicographiques $\underline{E}$ et $\overline{E}$ dont les ordres sont les mêmes en ce qui concerne le début des monômes entrant dans l'intersection. Il ne reste plus, ensuite, qu'à faire l'intersection pour retrouver le résultat.

Il est inutile d'avoir pour celà les écritures optimales quel que soit le coût. En effet, reprenons l'exemple § 3.b. les écritures $\underline{E}$ et $\overline{E}$ ne sont ni A-optimales ni $\mathcal{A}$-optimales (avec les opérateurs U et $\mathcal{U}$).

4 - SIMPLIFICATIONS PRELIMINAIRES

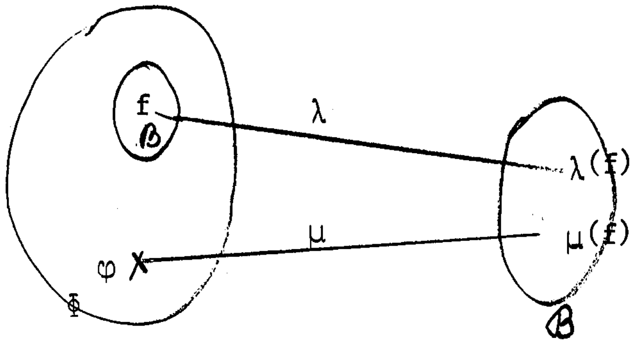
4. a. Parallèle entre fonctions booléennes et ϕ-booléennes

Appelons λ et μ les applications respectivement de $\mathcal{B}$ dans $\mathcal{B}$, de Φ dans $\mathcal{B}$

où $\mathcal{B}$ est l'ensemble des fonctions booléennes

- Φ - - ϕ-booléennes

qui à tout élément de $\mathcal{B}$ (resp. de Φ) fait correspondre son écriture lexicographique optimale



Propriétés

1.a. - λ est surjective

$\forall f \in \mathcal{B}$ on a $\lambda(f) = f$

donc tout $\mathcal{B}$ est atteint.

1.b. - λ est injective

$\forall f \in \mathcal{B}$ $\forall g \in \mathcal{B}$ $\lambda(f) \neq \lambda(g) \Rightarrow f \neq g$

car $\lambda(f) = f$ et $\lambda(g) = g$

est donc une bijection de $\mathcal{B}$ dans $\mathcal{B}$.

2. - μ n'est pas surjective car $\mathcal{B} \subset \Phi$

μ n'est pas injective

autrement dit

$\exists f \in \Phi, \exists g \in \Phi$ telles que $\mu(f) = \mu(g)$ avec $f \neq g$

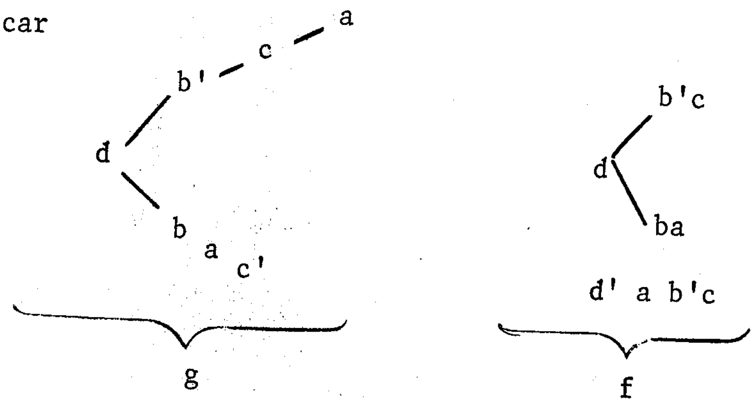
Exemple

$$f = a\,c\,d + a\,b\,c'd + \emptyset(abd + cd + ab'c)$$
$$g = a\,c\,d + a\,b\,c'd + \emptyset(abd + cd)$$

une solution pour les deux fonctions est

$$d\,b'c + d\,b\,a$$

car



4. b. - Conséquences

Ceci amène quelques théorèmes intéressants.

4.b.1. - l'une des bornes est indépendante d'une lettre, on peut imposer la même condition à l'autre borne, donc à la forme lexicographique optimale en prenant de nouvelles bornes. On ne restreint pas le problème en posant

$$\underline{f}(Y) \text{ et } \overline{f}(Y,a)$$

4.b.1.a. - La première idée est de prendre la fonction $(\underline{g},\overline{g})$ où

$$\underline{g} = \underline{f}(Y)$$

$$\overline{g} = \overline{f}(Y,0) + \overline{f}(Y,1)$$

- si l'on se trouve dans le cas $\underline{f}(Y,a)$ et $\overline{f}(Y)$ on prendra

$$\underline{g} = \underline{f}(Y,0)\ \underline{f}(Y,1)$$

et $\overline{g} = \overline{f}(Y)$

toute solution compatible avec $(\underline{g},\overline{g})$ ne le sera pas forcément avec $(\underline{f},\overline{f})$.

Prenons

$$f = abcd + a'b'c'd + \emptyset(c'de'+a'b'd+a'b'c'e'+de+abc)$$

$$X = \{a,b,c,d,e\} \qquad Y = \{a,b,c,d\}$$

$$\overline{f}(X) = c'de' + a'b'd + a'b'c'e' + de + abc$$

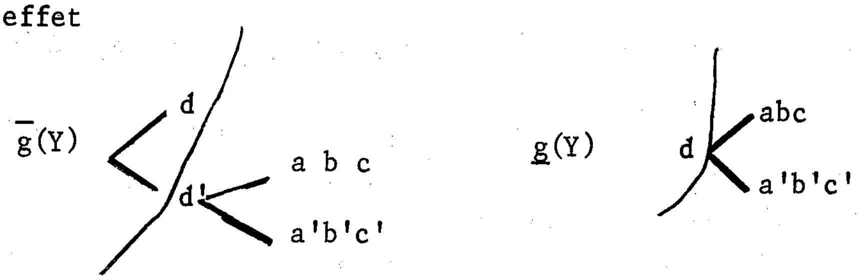
$$\underline{f}(X) = abcd + a'b'c'd = \underline{g}(Y)$$

$$\overline{g}(Y) = \overline{f}(Y,0) + \overline{f}(Y,1) = c'd + a'b'd + a'b'c' + d + a\ b\ c$$
$$= d + a\ b\ c + a'b'c'$$

une forme lexicographique optimale pour $(\underline{g}, \overline{g})$ est

d

en effet



or d n'est pas compatible avec $(\underline{f}, \overline{f})$

on n'a que $d > \underline{f}$

mais pas $d \leq \overline{f}$

<u>Remarque</u>

On a

$$\underline{g}(Y) \leq \underline{f}(X) \leq f \leq \overline{f}(X) \leq \overline{g}(Y)$$

une fonction φ vérifiant $\underline{g} \leq \varphi \leq \overline{g}$

ne vérifiera pas toujours $\underline{f} \leq \varphi \leq \overline{f}$

4.b.1.b. - il nous faudra donc prendre

$$\underline{g}(Y) = \underline{f}(X,0) + \underline{f}(X,1)$$

ou

$$\overline{g}(Y) = \overline{f}(X,0) \ . \ \overline{f}(X.1)$$

puisque c'est le seul cas qui tient compte du fait que les monômes de $\overline{f}$ qui ne sont diviseurs d'aucun monôme de $\underline{f}$ sont inutiles.

On a alors

$$\underline{f} \leqslant \underline{g} \leqslant \overline{g} \leqslant \overline{f}$$

toute solution compatible avec $(\underline{g}, \overline{g})$ le sera avec $(\underline{f}, \overline{f})$.

Sur l'exemple précédent

$$\overline{g}(Y) = a'b'd + a\ b\ c$$

$$\underline{g}(Y) = a\ b\ c\ d + a'b'c'd$$

où la solution est

$$b'd\ a' + b\ c\ a$$

qui est bien compatible avec f.

4.b.2. - <u>Généralisation dans le cas où les bornes ne dépendent pas des mêmes lettres</u>.

Supposons que nous ayons

$$\underline{f}(X,Y) \text{ et } \overline{f}(X,Y)$$

nous remplaçons $\underline{f}$ par la fonction obtenue en mettant à 1 (resp. 0) toutes les variables directes (resp. accentuées) de Y.

et $\overline{f}$ en mettant a 0 (resp. 1) toutes les variables directes (resp. accentuées) de Z.

<u>Remarque</u>. Pour ce qui concerne $\underline{f}$ on prend la fermeture supérieure indépendante des lettres de Y notée

$$\lceil \underline{f}(X\ Y) \rceil^{Y}$$

ce n'est pas exact pour $\overline{f}$ si elle n'est pas donnée sous forme de base complète. Dans le cas où elle l'est effectivement on prendra

$$\lfloor \overline{f}(X\ Z) \rfloor_{Z}$$

<u>Exemple</u> $X = \{a,b,c,d\}$

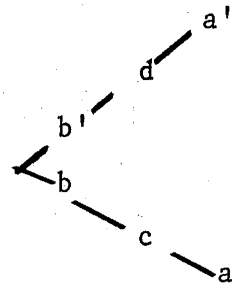
$$\underline{f} = abcdg + a'b'c'd$$

$$\overline{f} = c'de'f + a'b'd + a'b'c'e'f' + c\ d\ e\ f + a\ b\ c$$

$$\underline{g}(X) = \underline{f}^{\{g\}} = a\,b\,c\,d + a'b'c'd$$

$$\bar{g}(X) = \underbrace{\bar{f}}_{\{e,f\}} = a'b'\,d + a\,b\,c$$

d'où la solution



Ces simplifications sont très intéressantes au point de vue temps de calcul, en voici une idée sur l'exemple cité en b.1.a et b.1.b

Pour ($\underline{f}$, $\bar{f}$) au bout de 11' on n'a pas terminé complètement le programme mais au pas 3 la solution a été élaborée.

Pour ($\underline{g}$, $\bar{g}$) on a mis en tout 9"

THEOREME

> Si une fonction Ø booléenne est donnée par $\underline{f}(X\,Y)$ et $\bar{f}(X\,Z)$ on peut remplacer f par la fonction ($\underline{g}$, $\bar{g}$) où
>
> $$\underline{g}(X) = \overbrace{\underline{f}(X\,Y)}^{Y} \quad \text{et} \quad \bar{g}(X) = \underbrace{\bar{f}(X\,Z)}_{Z}$$
>
> et toute écriture lexicographique locale irréductible de ($\underline{g}$, $\bar{g}$) le sera pour ($\underline{f}$, $\bar{f}$).

4.c. - <u>Simplification due aux monômes de $\underline{f}$ et $\bar{f}$</u>

THEOREME

> Si dans une fonction Ø-booléenne ($\underline{f}(X)$. $\bar{f}(X)$), $\bar{f}$ contient un monôme qui a un produit nul avec tous les monômes de $\underline{f}$ on peut le supprimer dans $\bar{f}$ sans rien changer à la solution.

Soit $\underline{f} = \sum_{i \in I} \nu_i \quad \overline{f} = \sum_{j \in J} \mu_j$

Si $\exists k \in J \qquad \forall i \in I \quad \nu_i \mu_k = 0$

On restreint $\overline{f}$ à $\sum_{j \in J_1} \mu_j$ où $J_1 = J - \{k \in J, \quad i \in I \quad \nu_i \mu_k = 0\}$

En effet, ce monôme est surabondant dans $\overline{f}$ tous les points couverts par lui ne se retrouvent pas dans $\underline{f}$ donc aucun de ses multiples ne sera dans $\underline{E}$ on peut donc le supprimer.

4.d. - Etant données une fonction $\emptyset$-booléenne $(\underline{f}, \overline{f})$ et son écriture lexicographique Σ_0 optimale qu'arrive-t-il si on ajoute un monôme seulement à $\overline{f}$?

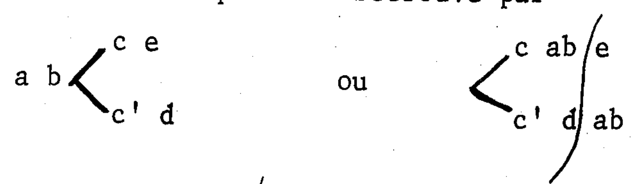
Prenons un exemple.

$$\left.\begin{aligned} \underline{f} &= d\,a\,b\,c' + a\,b\,c\,e \\ \overline{f} &= d + a\,b\,c \end{aligned}\right\} \Sigma_0 = c\,a\,b + c'd$$

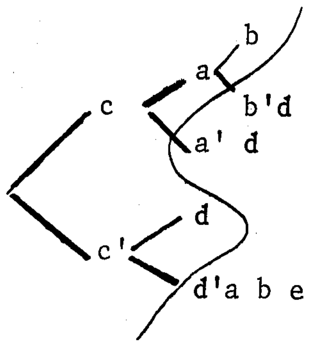
Ajoutons à $\overline{f}$ le monôme $a\,b\,e$.

Ce monôme couvre un monôme de $\underline{f}$ déjà couvert par un monôme de $\overline{f}$, a b c. Cette adjonction d'un monôme à $\overline{f}$ ne change pas la solution on a toujours $\Sigma_0 = c\,a\,b + c'd$ que l'on retrouve par

$\underline{E}$



$\overline{E}$



Voici le problème qui se pose :
Si un monôme de $\underline{f}$ a plusieurs diviseurs dans $\overline{f}$ un seul servira plus tard Seulement dans toutes les écritures $\underline{E}$ de $\underline{f}$ et $\overline{E}$ de $\overline{f}$ les monômes sont transformés et on ne sait plus quel diviseur va servir.

5 - EXEMPLE TRAITE PAR PROGRAMME

$$\overline{f} = b'd' + ab' + ce + a'de$$

$$\underline{f} = ab'd' + b'e + ace + a'de + ab'c$$

dont les bases complètes nécessaires sont

$$\overline{f} = b'd' + ab' + ce + a'de + b'e$$

$$\underline{f}' = a'e' + be' + abc' + a'bd' + bc'd' + c'de'$$

E1

PAS	ECRITURES G(Y)	COUT	MONOMES M DES G(Y)
	0	0	A'E'+BE'+ABC'+A'BD'+BC'D'+ C'DE'
	1	0	B'D'+AB'+CE+A'DE+B'E
1	A	1	B'E'+BCD'E
	A'	1	BC'DE
	B'	1	D'E'+A'D'+C'D'+AE'+AC'
	C	1	A'B'D'E'+AB'DE'+ABE+BD'E
	D	1	A'BE
	D'	1	A'B'E'+B'C'E'
	E	1	A'D+A'C+BC+A'B'+B'C'D
2	AB+B'	3	CD'E
	AB'	2	E'
	AC	2	B'DE'+BD'E
	AC+C'	3	B'D'E'
	AD+D'	3	B'E'
	AD'	2	B'C'E'
	AD'+D	3	BCE
	AE	2	BCD'
	AE'+E	3	B'
	A'B+B'	3	C'DE
	A'C'+C	3	BDE
	A'D	2	BC'E

E2

PAS	ECRITURES G(Y)	COUT	MONOMES M DES G(Y)
2	A'E	2	BC'D
	B'C	2	A'D'E' + ADE'
	B'C'	2	A'D'E'
	B'C' + C	3	D'E + AE
	B'D'	2	A'E' + C'E'
	B'D' + D	3	A'E
	B'E	2	A'D' + AC'D
	B'E' + E	3	CD' + AC
	CD + D'	3	AB'E'
	CD'	2	A'B'E'
	CD' + D	3	A'BE
	CE	2	AB + BD'
	CE' + E	3	A'B'D' + AB'D
	DA' + CA	4	BE
	DE	2	A'B
	D'C' + AC	4	B'E'
	D'E' + E	3	A'B' + B'C'
	EA' + B'A	4	C'D
	EC + C'	3	A'B'D'
	EC + B'C'	4	A'D'
	ED + B'D'	4	A'

E3

PAS	ECRITURES G(Y)		COUT	NONOMES M
3	ABC + B'C + B'C'	OPT	7	D'E
	ABD' + B'D' + D		6	CE
	ABE + B'E + B'E'	OPT	7	CD'
	AB'C		3	DE'
	AB'C + B'C'		5	D'E'
	AB'D + B'D'		5	E'
	AB'D'		3	C'E'
	AB'E' + E		4	C
	AB'E' + B'E		5	C'D'
	AC B + B'		4	D'E
	ACE		3	B'D'
	ACE' + E		4	B'D
	ACE' + C'E' + E		6	B'D'
	ADE' + D'E' + E		6	B'
	AD'E' + E		4	B'C'
	AD'E + DE		5	BC
	AEB + B'		4	CD'
	A'BC' + B'C' + C		6	DE
	A'BD + B'D + B'D'	OPT	7	C'E
	A'C'E + CE		5	BD
	A'DB + B'		4	C'E

OPT : Fonction supprimée en cours d'optimisation par une g(Y) de la même classe (au même pas).

E4

PAS	ECRITURES G(Y)		COUT	MONOMES M
3	A'DE		3	BC'
	B'CD + B'D'		5	AE'
	B'CD'		3	A'E'
	B'C'D' + CD' + D		6	A'E
	B'C'E + CE + B'E'	OPT	7	D' + A
	B'C'E + CE + AB'E'		8	D'
	B'CE + CE + B'CE'		8	A'D' + AD
	B'D'C' + AB'C		6	E'
	B'D'E' + E		4	A'C
	B'D'E' + B'E		5	AC'
	B'D'A + DA' + B'C'A + CA	OPT	10	E
	CD'E + D'E' + E		6	AB'
	CD'E' + E		4	A'B'
	CD'E + DE		5	A'B
	CEB + B'		4	A + D'
	DA'B + CAB + B'		7	E
	DA'E + CAE		6	B
	D'C'E' + ACE' + E		7	B'
	EA'D + B'AD + B'D'		8	C'
	ECD' + B'C'D' + ED		8	A'
	EDA + B'D'A' + B'D'A + EA		11	C

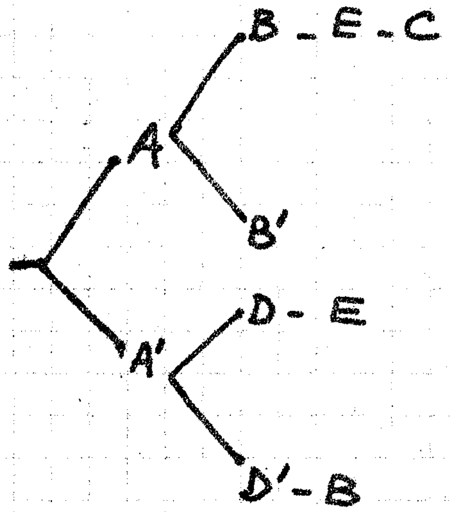
ES

PAS	ECRITURES G(Y)	COUT	MONOMES M
4	$AB'DE' + B'D'E' + E$	8	C
	$AB'E'C + EC + EA'C' + B'AC'$ OPT	12	D
	$ACEB + B'$	5	D'
	$A'DBC' + B'C' + C$	7	E
	$B'CDE' + B'D'E + B'C'E + CE$	12	4
	$CEBA + B'A + EA'$	8	D
	$CEBA + B'A + EDA' + BD'A'$	12	1
5	rien à ce pas on ne cherche que les solutions éventuelles, meilleures que celle déjà trouvée.		

SOLUTION DE COUT EN LETTRES OPIMAL 12

$$CEBA + B'A + EDA' + BD'A'$$

dont le réseau est :



et la forme habituelle :

$$ABEC + AB' + A'DE + A'D'B$$

BORNES FSUP ET FINF DE LA FCNCTICN A TRAITER

B'D' + AB' + CE + A'DE
AB'D' + B'E + ACE + A'DE + AB'C

COMPLEMENT DE FINF
A'E' + BE' + ABC' + A'BD' + BC'D' + C'DE'

DONT LES BASES CCMPLETES SCNT
B'D' + AB' + CE + A'DE + B'E

A'E' + BE' + ABC' + A'BD' + BC'D' + C'DE'
.............6 DIXIEMES DE SECCNDE

PRESENCE

.............11 DIXIEMES DE SECCNDE

BIFURCATION AU PAS 2

CCMPT=8   IND=31
ECRITURES SELECTICNNEES A LA FIN DU PAS 2

AB + B'
AB'
AC
AC + C'
AD + D'
AD'
AD' + D
AE
AE' + E
A'B + B'
A'C' + C
A'D
A'E
B'C
B'C'
B'C' + C
B'D'

B'D' + D
B'E
B'E' + E
CD + D'
CD'
CD' + D
CE
CE' + E
DA' + CA
DE
D'C' + AC
D'E' + E
EA' + B'A
EC + C'
EC + B'C'
ED + B'D'

.............70 DIXIEMES DE SECCNDE

BIFURCATION AU PAS 3

COMPT=61 IND=73
ECRITURES SELECTIONNEES A LA FIN DU PAS 3

ABD' + B'D' + D
AB'C
AB'C + B'C'
AB'D + B'D'
AB'D'
AB'E' + E
AB'E' + B'E
ACB + B'
ACE
ACE' + E
ACE' + C'E' + E
ADE' + D'E' + E
AD'E' + E
AD'E + DE
AEB + B'
A'BC' + B'C' + C
A'C'E + CE
A'DB + B'
A'DE
B'CD + B'D'
B'CD'
B'C'D' + CD' + D
B'C'E + CE + AB'E'
B'C'E + CE + B'CE'
B'D'C' + AB'C

```
B'D'E' + E
B'D'E' + B'E
CDE' + D'E' + E
CD'E' + E
CD'E + DE
CEB + B'
DA'B + CAB + B'
DA'E + CAE
D'C'E' + ACE' + E
EA'D + B'AD + B'D'
ECD' + B'C'D' + ED
EDA' + B'D'A' + B'E'A + EA

.............376 DIXIEMES DE SECCNDE


 --------------------
 BIFLRCATION AL PAS 4
 --------------------


                   CCMPT=160    IND=118
 SOLLTION  DE CCLT 12
CEBA + B'A + EDA' + B'D'A'
 ECRITLRES SELECTICNNEES A LA FIN CU PAS 4
 -----------------------------------------


AB'DE' + B'D'E' + E
ACEB + B'
A'DBC' + B'C' + C
B'CDE' + B'D'E' + B'C'E + CE
CEBA + B'A + EA'

.............661 DIXIEMES DE SECCNDE


 --------------------
 BIFLRCATION AL PAS 5
 --------------------


                   CCMPT=179    IND=124
 ECRITLRES SELECTICNNEES A LA FIN CU PAS 5
 -----------------------------------------



.............674 DIXIEMES DE SECCNDE



 -+-+-+-+-+-+-+-+-+-+-+-+-+-+-
 SOLLTION OPTIMALE  DE CCLT  12
```

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

CEBA + B'A + EDA' + B'D'A'

..............676 DIXIEMES DE SECONDE

ON A TRAVAILLE AVEC LE COUT EN LETTRES

CHAPITRE IV

DIVERS PROGRAMMES ET RESULTATS MACHINE - COMPARAISON DES OPTIMISATIONS DES ECRITURES D'UNE MEME FONCTION

1. - Programmes et Résultats Machine
2. - Comparaison des optimisations
3. - Formes lexicographiques de f et f'

1 - PROGRAMMES ET RESULTATS MACHINE

1.a. - Divers programmes

Nous avons programmé la méthode exposée au chapitre III en Algol du fait de la présence à l'I.M.A.G. de procédures MAP-IBM-7044 et de procédures Algol de traitement et de calcul sur les fonctions booléennes. Un autre langage aurait peut être donné des résultats en des temps plus rapides mais le rapport des temps de deux programmes serait sensiblement le même.

Le programme LEXICO dont on trouvera un listing en appendice est la toute dernière version qui réduit au maximum le nombre des g(Y) essayées et ne construit aucune bifurcations a $g_1(Y) + a' \ g_2(Y)$ avant d'avoir effectué le test de validité par le produit des monômes associés. Ces monômes sont gardés en mémoire sous deux formes, le vecteur de vérité de la fonction qu'ils représentent calculé sur le nombre effectif de variables dont la g(Y) dépend, et la forme polynomiale résultant du produit effectué. Le vecteur de vérité est nécessaire dans l'optimisation pour savoir si deux g(Y) de coûts différents ont les mêmes monômes m. Ce programme Algol fait au total 7618 unités syntaxiques. La complexité du programme est accrue par la représentation machine des fonctions booléennes avec ordination quelconque des variables ainsi que les divers renseignements que nous devons garder en mémoire sur chaque g(Y). Ce programme existe en 3 variantes où l'optimisation est réalisée sur les 3 coûts L, A et $\mathcal{A}$. Nous avons en outre écrit d'autres variantes où l'optimisation s'opère suivant deux critères de coût dont le premier est prioritaire sur l'autre. Le second optimisant là où le premier conclu à un litige. Nous les avons appelées

L + A et A + L ,   L + $\mathcal{A}$ E et $\mathcal{A}$ + L

il ne nous a pas paru utile d'écrire les variantes A + $\mathcal{A}$ et $\mathcal{A}$ + A.

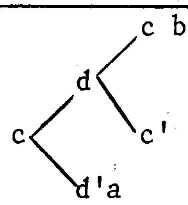
1.b. - Résultats Machine

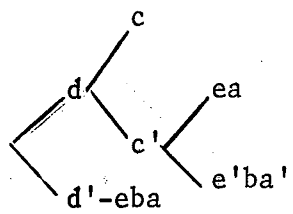
Référence de la fonction f(X)	Référence du complément f'	Fonctions Etudiées f(X)
1-4		abd + cd + ab'c
1-5		abce + c'de + ad'e
2-5		abe + cd + bde' + ade
3-5		abde + abc' + ab'de' + a'c'd' + a'bcd + a'cde'
4-5		ace + bcd + d'e
5-5	(7-5)	ab + bcd + d'e
6-5		ae' + b'cd + d'e
7-5	(5-5)	b'd + a'c'd + b'e' + a'c'e' + a'd'e'
1-6	(6-6)	abc + b'd'e + ad'f + a'bd' + cd'f
2-6	(8-6)	aef' + ab' + ad' + a'cf'
3-6	(4-6)	ab + af + cde + cd'f'
4-6	(3-6)	a'c' + b'c'f' + a'de' + b'de'f' + a'd'f
5-6	(11-6)	a'd + b'd + c'd + b'e'f' + abcf' + ac'e'f' + a'b'c'e'
6-6	(1-6)	a'd + b'd + c'd + b'e'f' + abc'f' + ac'e'f' + a'b'c'e'
7-6	(9-6)	acf' + bd + ef + b'ce
8-6		bdf + abde' + a'c' + bc'de' + a'f
9-6	(7-6)	a'b'e' + b'e'f' + a'd'e' + d'e'f + b'c'f + c'd'f'+a'bd'f'
10-6	(12-6)	a'd + b'd + c'd + b'e'f' + abc'f' + a'c'ef' + a'b'c'e'
11-6	(5-6)	a'bd' + b'd'e + c'de' + abcf + ad'f + cd'f
12-6		abc + bcd' + cd'e + ab'd'e + a'bd'e' + ad'f + bd'f + cd'f + d'ef
13-6		a'd + b'd + c'd + b'e'f' + abc'f' + a'c'e'f' + a'b'c'e'
14-6		a'b'de' + cd'eaf + b'd'f + ac'f + abd'ce + a'bcdef
15-6		c'df + a'b'd + abc + a'b'c'e'f' + cdef
16-6		a'bde' + b'd'f + ac'f + ad'ef + bc'def

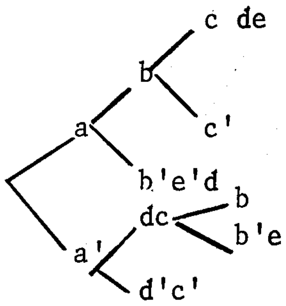
Référence (suite)		
17-6		def + bc'e' + ac'f + c'ef' + ac'd
18-6		cf' + ce' + cd' + a'b'e' + a'def + a'b'd'f + b'c'e'f'
19-6		bdf + c'd'e' + abe' + b'de' + ae'f
20-6		b'e' + a'd + d'b'a'ef' + a'c' + bd
21-6		abcd' + cdef' + efa'b'
22-6		abc'd' + cde'f' + efa'b'
23-6		abd + cd + ab'c + e'f
24-6		
25-6		d'e'f' + ad'ef + bc'def + a'b'de' + b'd'f + ac'f
26-6		bcd' + be + c'ef
27-6		c'df + a'b'd + abc + a'b'c'd'e'f' + cdef.
1-7		
2-7		
3-7		ac'eg + d'f'g + ab'c'g' + b'd'f'
4-7		abd + cd + ab'c + ef' + f + g

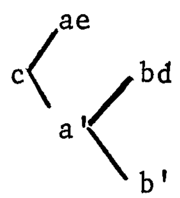
Réf	SOLUTIONS L-OPTIMALES
1-6	bac + bac'fd' + ba'd' + b'd'e + b'd'e'fa + b'd'e'fa'c
2-6	bd + bd'fc + bd'f'ca + b'ec + b'ec'f + b'e'f'ca
3-6	af + af' + af'b'cd' + af'b'cde + a'cde + a'cd'f'
4-6	a'c' + a'cd'f + a'cde' + af'b'c' + af'b'ce'd
5-6	cba'd + cbaf' + cb'd + cb'd'f'e' + c'd + c'd'e'a'b' + c'd'e'af'
6-6	dc' + dcb' + dcba' + d'f'b'e' + d'f'bc'a + d'fe'c'b'a'
7-6	fe + fe'db + f'b'ce + f'b'ce'a + f'bd + f'bd'ca
8-6	adbf + adbf'e' + a'f + a'f'c'
9-6	fe'db' + fe'd' + f'ca'bd' + f'ca'b'e' + f'c'd' + f'c'db'
10-6	dcba' + dcb' + dc' + d'fe'c'b'a' + d'f'abc' + d'f'ab'e' + d'f'a'e'b' + d'f'a'ec'
11-6	cabf + cb'd'f + cb'd'f'e + c'd'e'af + c'd'e'a'b + c'd'e
12-6	cbd' + cbda + cb'd'f + cb'd'f'e + c'd'e'a'b + c'd'e'af + c'd'ef + c'd'ef'b'a
13-6	dc' + dcb' + dcba' + d'bf'c'a + d'bf'c'ae' + d'b'e'f' + d'b'e'fc'a'
14-6	ac'f + acd'be + acd'b'f + a'b'de' + a'b'd'f + a'bfedc'
15-6	c'df + c'df'b'a' + c'd'f'e'b'a' + cb'da' + cb'dafe + cba + cba'fed
16-6	bfedc' + bfed'a + bfe'c'a + b'a'f + b'da'e' + b'dafc'
17-6	efd + efd'c'a + ef'c' + e'c'b + e'c'b'af + e'c'b'a'f'd
18-6	d'c + d'c'b'fa' + d'c'b'f'e' + defa' + def'c + de'c + de'c'b'a'
19-6	dbf + dbf'e'a + db'e' + d'e'c + d'e'c'af + d'e'c'af'b
20-6	bd + bd'c'a' + b'ea'd + b'ea'd'fc' + b'ea'd'f' + b'e'
21-6	bcdf'e + bcd'a + b'efa' + b'ef'dc
22-6	fbd'c'a + fb'ea' + f'de'c + f'd'c'ba
23-6	abd + abd'fe' + ab'c + ab'c'fe' + a'fedc + a'fe' + a'f'dc
25-6	
26-6	be + be'd'c + b'fec' + b'f'e'
27-6	fcde + fcde'ba + fcde'b'a' + fcd'ba + fc'd + f'bca + f'b'a'd + f'b'a'd'e'c'
3-7	fc'age + fc'ag'b + f'dc'age + f'dc'ag'b' + f'd'g + f'd'g'b'
4-7	gf + gf'e + gf'e'abd + gf'e'ab'c + gf'e'a'dc + g'

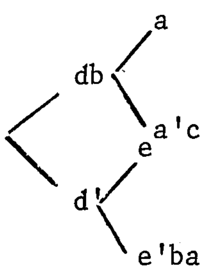
Quelques réseaux solutions.

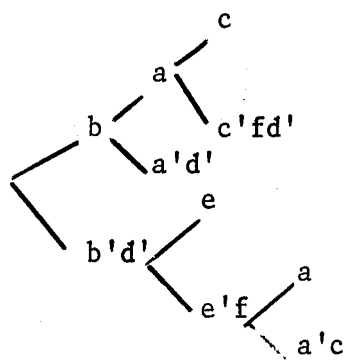
1-5 

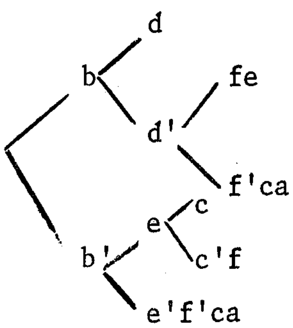
2-5 

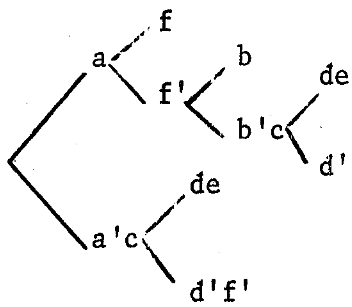
3-5 

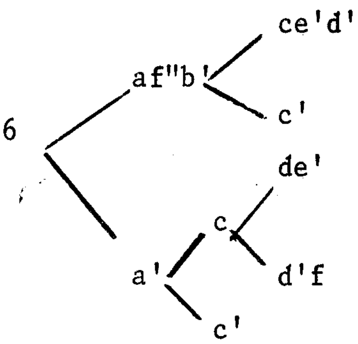
4-5 

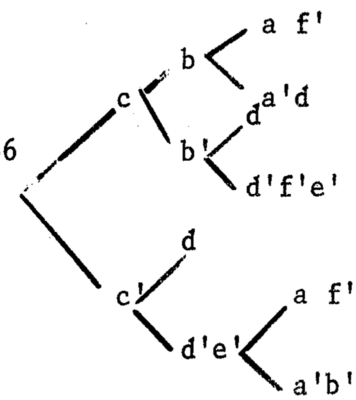
5-5 

1-6 

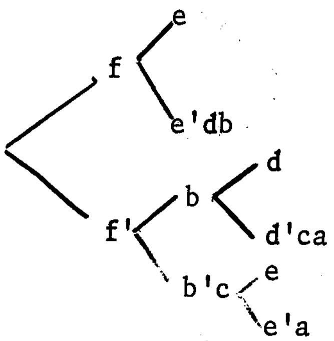
2-6 

3-6 

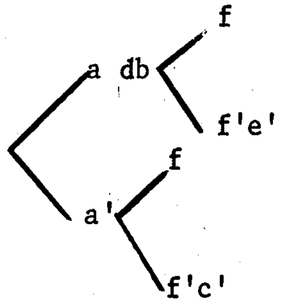
4-6 

5-6 

7-6



8-6



Réf^ce	Temps L	Temps U-A	Temps U - A	Lexico pas écriture	lexico avec écriture
1-4	15"				
1-5	20"				
2-5	1'20"				
3-5	1'10"	1'10"			
5-5	30"				
6-5	30"				
7-5	30"				
1-6	4'40"	4'50"	4'40		3'10
2-6	1'30"	1'30"	1'40"		
3-6	3'	3'	3'30"		
4-6	3'				
5-6	5'30"	5'30	5'30"		
6-6	3'50	4'	4'10"	2'37"	3'
7-6	5'	5'			
8-6	1'20"				
9-6	3'40				
10-6	6'				
11-6	5'				
12-6	5'				
13-6	5'	5'			
14-6		8'			
15-6	6'	6'50	6'50		
16-6	6'10				
17-6	4'40				
18-6		5'30			
19-6	4'40"				
20-6	3'10"				
21-6	2'50"	2'50"			
22-6	2'20	2'20			
23-6	3'10"				
24-6					
25-6	> 7'				

26-6	1'50"			
27-6	6'50	6'55		
11-7	> 12'			
2-7	> 8'			
3-7	11'	11'		7'46" pour le dernier
4-7	9'			

1.c. - <u>Elimination heuristique</u> cf [4] p.18.

Nous avons essayé de supprimer certaines écritures g(Y) au moment de leur formation parce qu'elles avaient une "distance" à la fonction f(X) trop élevée. Nous ne sommes pas toujours assurés d'arriver à une solution. Il faudrait traiter un grand nombre d'exemples pour avoir une valeur convenable de cette distance minimum afin de parvenir à une écriture optimale.

<u>Distance</u>

Nous prenons comme distance de deux fonctions le nombre de composantes de leur vecteur de vérité qui diffèrent. Ces vecteurs ayant tous 2^n composantes si X dépend de n variables.

Nous n'avons pas eu de résultats concluants

----

2. COMPARAISON DES OPTIMISATIONS DES ECRITURES D'UNE MEME FONCTION VIS A VIS DES DIFFERENTS COUTS

2.a. - <u>Position du problème</u>

Nous voulons savoir si les écritures lexicographiques locales irréductibles d'une même fonction, optimales vis à vis des différents coûts L, A et $\mathcal{A}$ peuvent être les mêmes où si elles sont toutes différentes.

Les programmes

L A $\mathcal{A}$

donnent respectivement les coûts

$$L : \begin{cases} L_0 \\ A_1 \\ \mathcal{A}_1 \end{cases} \quad A : \begin{cases} L_2 \\ A_0 \\ \mathcal{A}_2 \end{cases} \quad \mathcal{A} : \begin{cases} L_3 \\ A_3 \\ \mathcal{A}_0 \end{cases}$$

où L_0, A_0 et $\mathcal{A}_0$ sont les coûts optimaux.

Appelons $\mathcal{L}$, A et $\mathcal{A}$ l'ensemble des écritures ayant respectivement pour coût L_0, A_0 et $\mathcal{A}_0$.

$L_0 = \text{Min } \{L_E, E \in \mathcal{L}\}$

$\mathcal{L} = \{E, L_E = L_0\}$

$\forall E \quad L_E \geq L_0$ est la propriété du coût optimal.

Ici nous avons

$$L_0 \leq L_2 \qquad A_1 \geq A_0 \qquad \mathcal{A}_1 \geq \mathcal{A}_0$$
$$L_0 \leq L_3 \qquad A_3 \geq A_0 \qquad \mathcal{A}_2 \geq \mathcal{A}_0$$

Si nous voulons connaitre les coûts L optimaux des écriture U et $\mathcal{U}$ arborescentes optimales nous utilisons les programmes A + L ou $\mathcal{A}$ + L qui donnent

$$A + L \quad \begin{cases} L_2' \\ A_0 \\ \mathcal{A}_2' \end{cases} \qquad \qquad \mathcal{A} + L \quad \begin{cases} L_3' \\ A_3' \\ \mathcal{A}_0 \end{cases}$$

où $L_2' \leq L_2$ et $L_3' \leq L_3$

2.b. <u>Comparaison vis à vis des coûts L et $\mathcal{A}$</u>.

Prenons l'exemple réf. 15-6

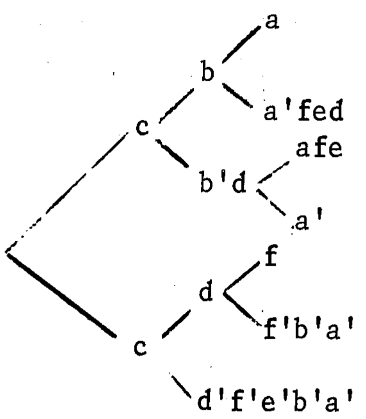
$$f = c'df + a'b'd + abc + a'b'c'e'f' + cdef$$

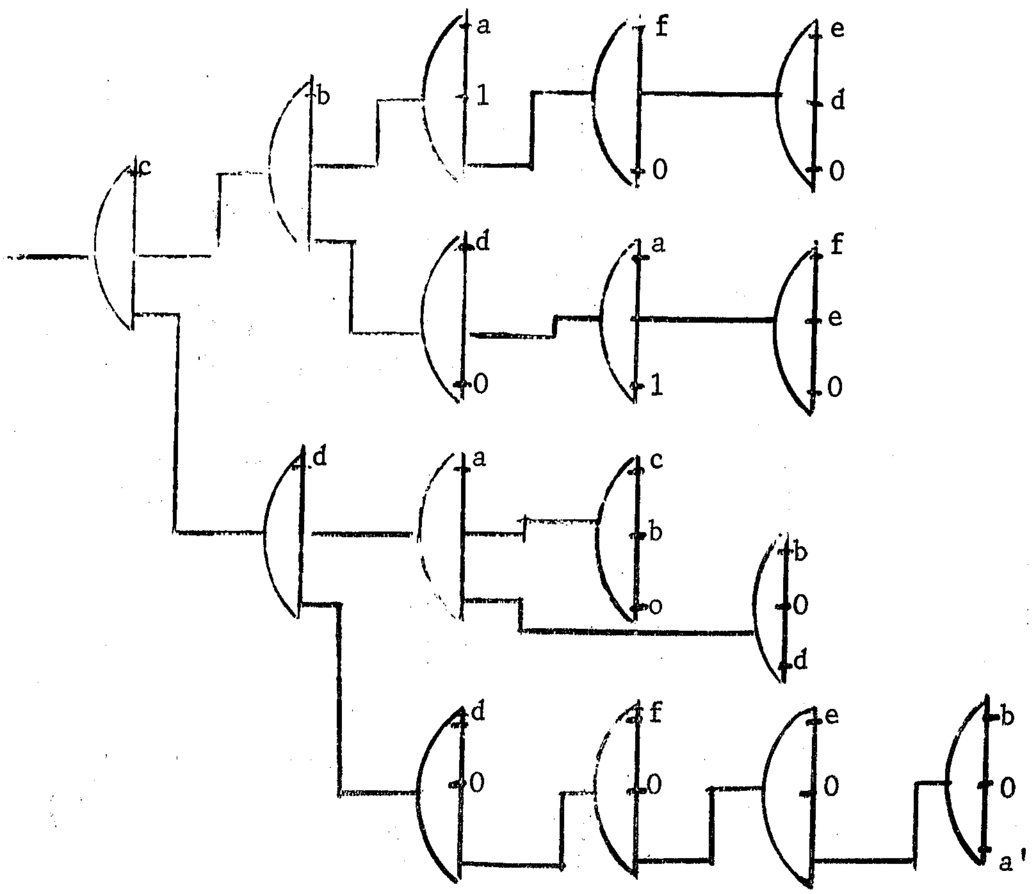
le programme L donne pour solution :

$$cba + cba'fed + cb'afe + cb'da' + cd'f + c'df'b'a' + c'd'f'e'b'a'$$

$$L_0 = 33 \qquad \mathcal{A}_1 = 16$$

les réseaux lexicographiques et $\mathcal{U}$ arborescents sont :



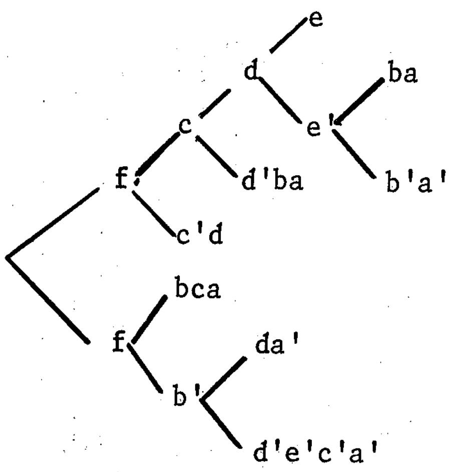


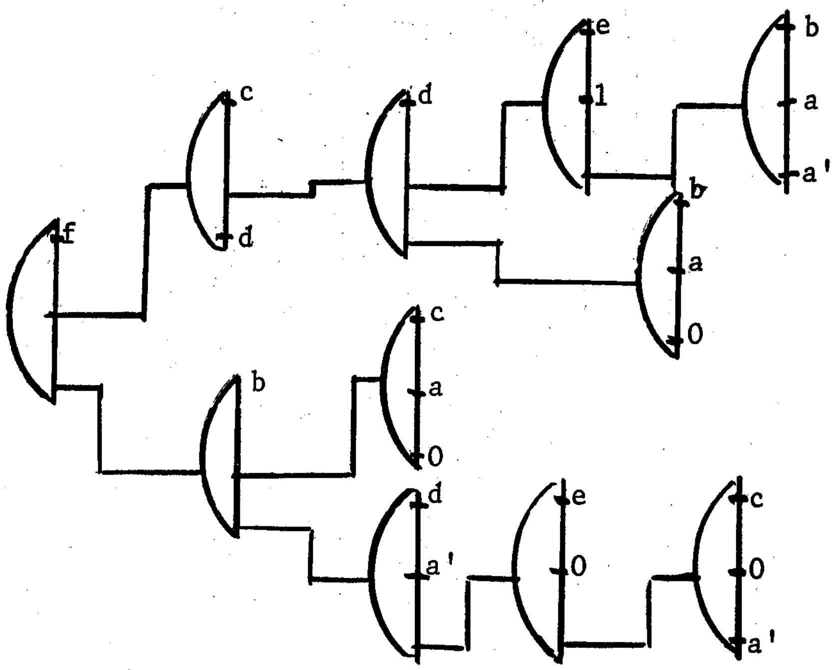
le programme $\mathcal{A}$ donne

fcde + fcde' ba + fcde'b'a' + fcd'ba + fc'd + f'bca + f'b'da' + f'b'd'e'c'a'.

où L_2 = 38 $\mathcal{A}_0$ = 11

nous ne pouvons conclure car le programme ne donne pas forcément $L'_2 = 38$
Le programme $\mathcal{A}$+ L donne la même solution que le programme $\mathcal{C}$ dont les réseaux sont :





Conclusion

Il n'y a pas parmi les écritures $\mathcal{A}$-optimales d'écritures L-optimales.

$$\exists f ; \qquad \mathcal{L}_f \cap \mathcal{A}_f = \emptyset$$

ce n'est pas vrai pour toute fonction f

Si f = a on a $\mathcal{L}_f \equiv \mathcal{A}_f$

Nous pouvons énoncer le résultat suivant :

Résultat

Quelle que soit la fonction f, ses deux ensembles de solutions L-optimales et $\mathcal{A}$-optimales ne sont pas comparables :

— il existe des fonctions telles que des écritures lexicographiques locales réalisent le minimum des coûts, soit $\mathcal{A}_0$ et L_0.

— il existe une fonction pour laquelle il n'y a aucune écriture lexicographique locale réalisant $\mathcal{A}_0$ et L_0 à la fois.

2.c. - Comparaison vis à vis de A et L

Nous n'avons pu à l'opposé du 2.b., conclure d'une façon nette, sur la comparaison de A et $\mathcal{L}$.

Montrons avec des exemples les deux propriétés.

Propriété 1

Il existe deux écritures lexicographiques d'une même fonction E_1 et E_2 dont les coûts respectifs A_1, L_1 et A_2, L_2 vérifient

$$L_1 = L_2 \quad \text{et} \quad A_1 < A_2$$

Exemple

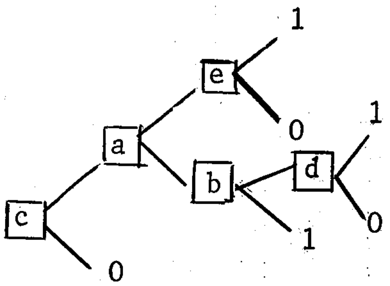
$f(X) = ace + a'cd + a'b'c$

$E_1 = cae + ca'bd + ca'b'$

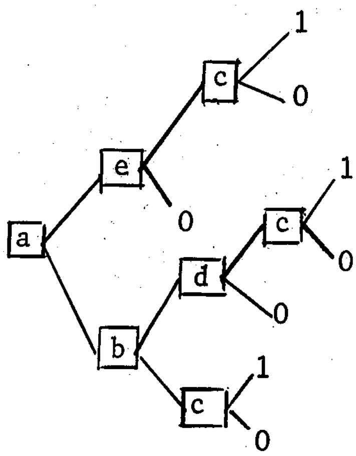
$E_2 = aec + a'bdc + a'b'c$

$L_1 = L_2 = 10$

Les réseaux U - arborescents suivants donnent A_1 et A_2



$A_1 = 5$



$A_1 = 5$

$A_2 = 7$

autrement dit :

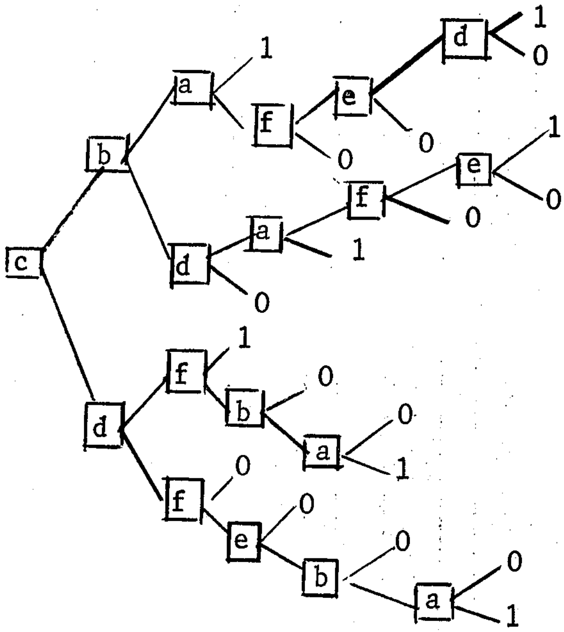
Parmi les solutions L-optimales certaines peuvent ne pas être A-optimales

ou $\exists f \mid \mathcal{L}_f \neq A_f$

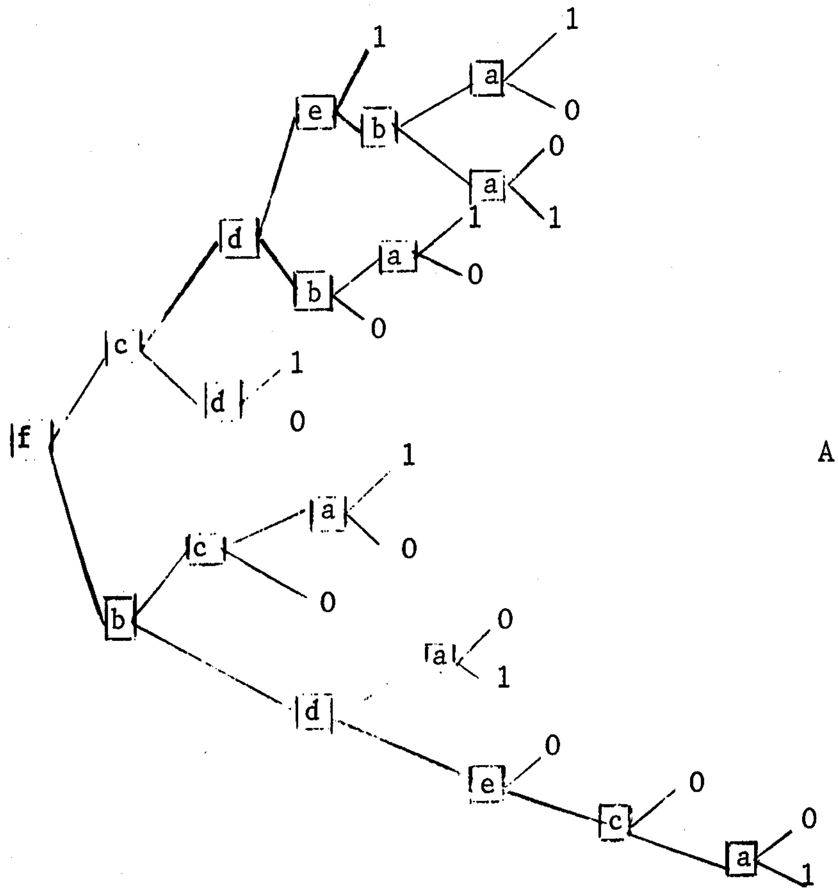
où $A = 16$ $L = 33$

où 33 est bien le coût en lettres ce qui ne permet pas de conclure.

Voyons les réseaux U des autres solutions L et $\mathcal{A}$ + L optimales



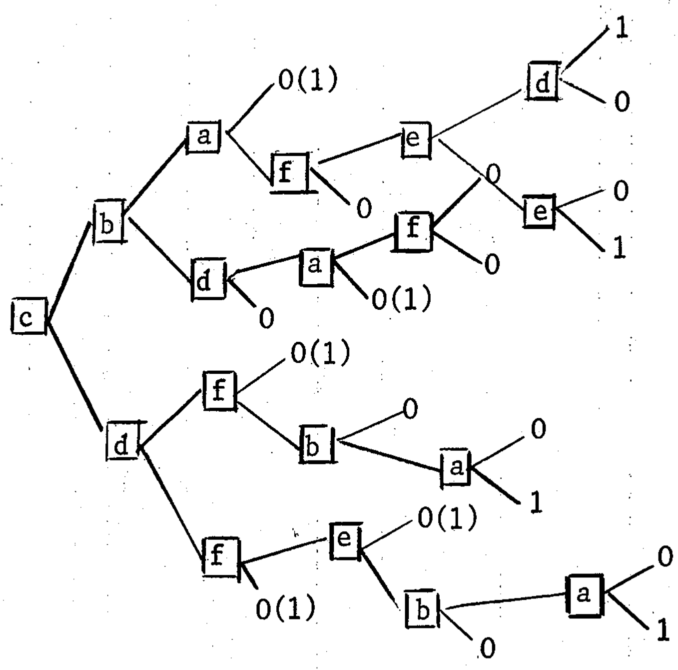
$A = 18$ $L = 33$



A = 17 L = 38

Il y a des cas pour lesquels L est défavorisé : lorsque beaucoup de 1 apparaissent dans les entrées des opérateurs U et peu de 0.

Exemple. Voyons la différence en prenant le réseau normalement dessiné ci-dessous et celui qui prend comme entrées les chiffres entre parenthèses lorsqu'il y en a et les autres chiffres quand il n'y en a pas.



$A = A' = 18$

$f = cba'fed + cb'dafe' + c'df'b'a' + c'd'fe'ba'$

$f' = cba + cba'fed + cb'dafe' + cb'da' + c'df + c'df'b'a' + c'd'fe$

$+ c'd'fe'ba' + c'd'f'$

$L = 23 \qquad L' = L + 17 = 40$

Malheureusement la fonction est modifiée, il y a en général des solutions de coût meilleur.

Propriété 2

Il existe deux écritures lexicographiques d'une même fonction E_1 et E_2 dont les coûts A_1, L_1 et A_2, L_2 vérifient

$$A_1 = A_2 \qquad \text{et} \qquad L_1 < L_2$$

Exemple

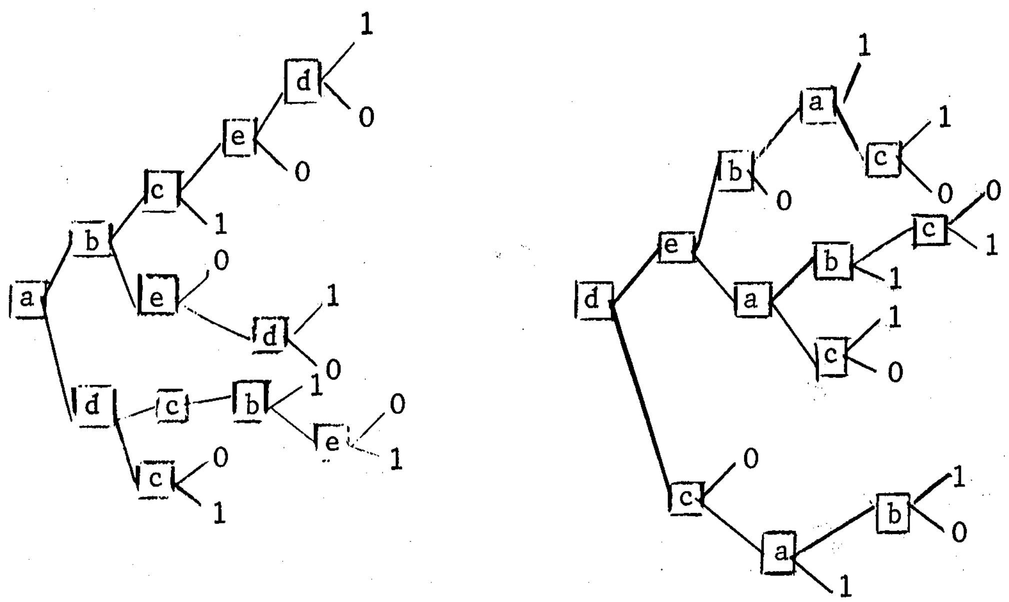
Réf 3-5 $f = abde + abc' + ab'de' + a'c'd' + a'cbd + a'cde'$

$E_1 = abcde + abc' + ab'e'd + a'dcb + a'dcb'e' + a'd'c'$

$E_2 = deba + deba'c + de'ab' + de'abc' + de'a'c + d'c'a + d'c'a' + d'c'ab$

$L_1 = 24$ $\qquad$ $L_2 = 29$

et les réseaux



où encore

Parmi les solutions A-optimales certaines peuvent ne pas être L-optimales

ou $\exists f \qquad A_f \not\equiv \mathcal{L}_f$

Propriété 3

Il existe une fonction pour laquelle ces solutions A-optimales et L-optimales sont toutes les mêmes

$$\exists f \qquad A_f \equiv \mathcal{L}_f$$

il n'y a qu'à prendre $f = a$

La seule chose que nous pouvons conclure au point où nous sommes est :

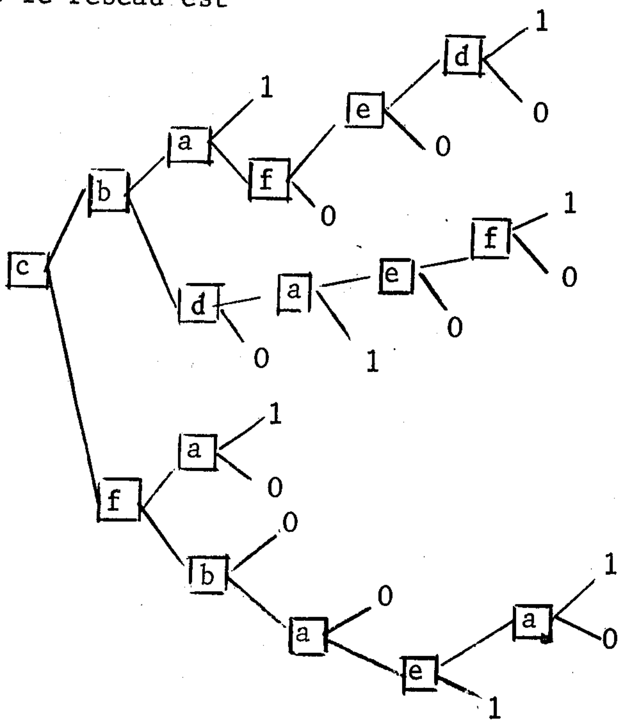
$$\boxed{\exists f \quad \text{pour laquelle} \quad A_f \wedge \mathscr{L}_f \neq \emptyset}$$

<u>Reste à montrer</u> :

$$\boxed{\forall f \quad \text{on a} \quad A_f \wedge \mathscr{L}_f \neq \emptyset}$$

où quelle que soit f il y a une solution réalisant le minimum des 2 coûts.

Reprenons l'exemple du paragraphe 2.b. le programme A + L donne l'écriture dont le réseau est

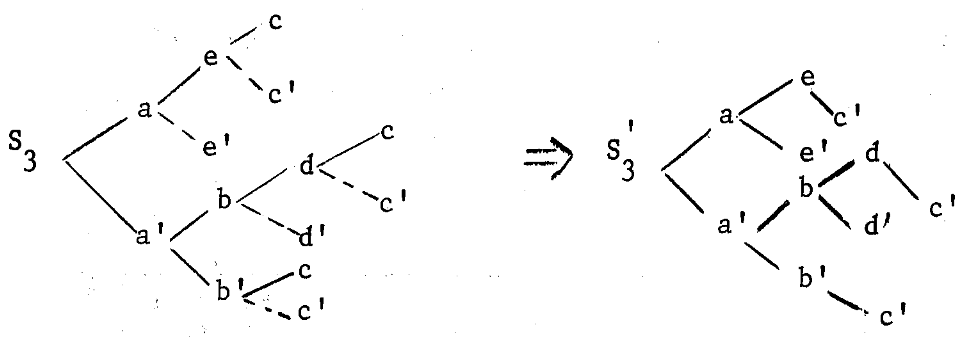


3 - FORMES LEXICOGRAPHIQUES DE F ET F'

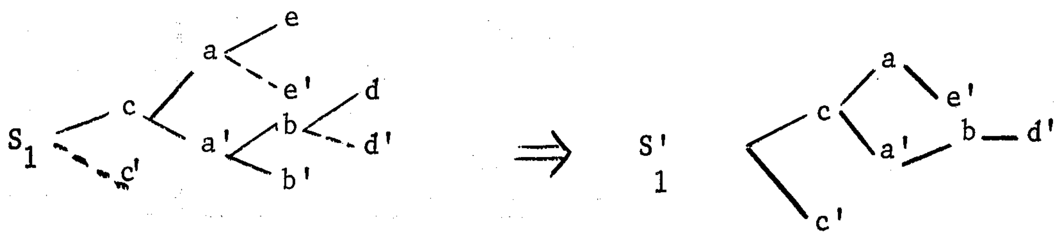
3.a. Ordres lexicographiques optimaux de f et de f'

De la propriété 1 il résulte que si l'on part d'une écriture lexicographique locale L-optimale mais pas A-optimale (ni $\mathcal{U}$) les ordres lexicographiques locaux optimaux de f et de f' ne se correspondent pas en appliquant l'algorithme § 3.a(chap.I)

Exemple précédent



$f' = aec' + ae' + a'bdc' + a'bd' + a'b'c'$ L = 15



$f' = cae' + ca'bd' + c'$ L = 8

S_1 est en effet A-optimale (ou $\mathcal{A}$), S_3 ne l'est pas et au § 3.b propriété(ch.I) on dit que f et f' ont des représentations A(ou $\mathcal{A}$) optimales de même coût d'où le résultat.

Les ordres lexicographiques locaux des formes lexicographiques locales irréductibles de f et de son complément f' qui ont des représentations U-arborescentes optimales sont les mêmes.

Il n'en est pas le cas généralement pour les formes optimales pour le coût en lettres puisque le complément n'est pas nécessairement optimal.

On ne peut pas dire que f et f' ont des ordres lexicographiques optimaux pour toute fonction f.

3.b. - Etude comparée de diverses fonctions et de leur complément.

Les temps d'obtention des solutions L-optimales relatives à f et f' ne sont pas toujours les mêmes ou voisins. Le nombre des g(Y) n'est pas le même non plus. Le tableau suivant montre l'évolution successive à chaque pas, dans chaque case on trouve le temps passé à ce pas en haut et à gauche, l'indice ou nombre total de monômes de tous les g(Y) sélectionnés pour avoir une idée des divers encombrements.

Il n'est cependant pas possible de savoir pour quelles raisons il est plus rapide de considérer une fonction plutôt que son complément.

Pas / Réf^{ce}	1	2	3	4	5	6	L	poids vecteur vérité	Temps total	optimale
1-6	1 8	9 70	66 198	131 307	63 367	14 367	25	29	4'45"	
6-6	1 8	11 63	64 185	109 251	46 284	8 284	24	35	4'	←
2-6	1 7	6 15	29 119	40 189	17 201	2 201	13	34	1'36"	
8-6	1 7	6 13	28 108	39 142	15 148	2 148	14	30	1'32"	←
7-6	1 8	9 66	62 220	150 374	34 426	12 426	23	37	5'	
9-6	1 8	8 51	51 146	101 263	64 299	7 299	24	27	3'52"	←

CHAPITRE V

QUELQUES PROPRIETES RELATIVES AUX FORMES LEXICOGRAPHIQUES IRREDUCTIBLES DES FONCTIONS COMPLETES.

1. - Fonction de la forme $x\,f(Y) + x'g(Z)$

2. - Propriétés dues aux monômes premiers

3. - Encadrement du nombre de monômes et du nombre de lettres d'une écriture lexicographique

4. - Somme de monômes disjoints et forme lexicographique

5. - Recherche de la lexicographicité d'une fonction donnée sous forme polynomiale

1. FONCTION DE LA FORME $x\, f(Y) + x'\, g(Z)$

où Y et Z ne dépendent d'aucun argument commun, on démontre le théorème cf. [9] pp.21.22

1.a. Théorème

> Si une fonction non égale à une constante est de la forme
> $$F = x\, f(Y) + x'\, g(Z)$$
> Y et Z ne dépendant d'aucun argument commun ni de x en particulier f ou g peut être égale à 0 ou 1. Il existe un ordre lexicographique local ayant pour première lettre x et optimal par rapport à F au point de vue des coûts U-arborescents.

La démonstration se trouve à la référence ci-dessus. Voici le principe: on part de $E = x\, E_f + x'\, E_g$ où E_f et E_g sont des écritures lexicographiques U-arborescentes optimales de f et de g. On considère une écriture E' quelconque de F, et en faisant $x = 1$ (resp. $x = 0$) on montre que son coût ne peut pas être inférieur à celui de E_f (resp E_g) donc que le coût de E' est supérieur ou égal à celui de E_f augmenté de celui de E_g et de 1 car Y et Z ne dépendent d'aucun argument commun ni de x.

1.b. Remarque

Nous noterons qu'il est alors suffisant de chercher pour F les écritures lexicographiques locales optimales du point de vue arborescent de f et de g.

1.c. Exemple

$$F = e(abd + cd + ab'c) + e'\ (f+gh)$$

cherchons une écriture A-optimale de F.

Ici $f(Y) = abd + cd + ab'c$

dont $E_f = abd + a'cd + ab'c$ de coût 6

et $g(Z) = f + gh$ dont $E_g = f + f'\ gh$ de coût 3

$E_F = e(abd + ab'c + a'cd) + e'(f + f'gh)$ de coût 10.

2. PROPRIETES DUES AUX MONOMES PREMIERS

2.a. Théorème.

Tout monôme de toute écriture lexicographique E de f est multiple au sens large d'au moins un monôme premier de f.

Soit un monôme lexicographique quelconque de E

$$\mu_K = a^* \; b^* \ldots \; \ell^*$$

μ_k est le seul monôme de E à couvrir l'hyperplan $a^* \; b^* \ldots \; \ell^*$ car tous les autres sont disjoints avec lui. Supposons qu'il n'existe pas de monôme premier ayant une partie (ou tout) des x^* de μ_k dans un ordre quelconque (c'est-à-dire un diviseur de μ_k). Les monômes premiers sont par définition inférieurs à f et tels qu'aucun de leurs diviseurs ne le soit.

Il en résulte qu'il n'est pas possible de trouver un monôme premier, multiple de μ_k. Les fonctions f et E ne coïncident pas sur l'hyperplan $a^* \; b^* \ldots \; ^*$ en conséquence on a le théorème .

2.b. Théorème

Si tous les monômes premiers de f ont tous la lettre a, tous les monômes lexicographiques ont la lettre a.

d'après le théorème précédent (§2.a) nous savons que tout monôme lexicographique est multiple d'au moins un monôme premier. Puisqu'ils ont tous la lettre a, tous les monômes lexicographiques l'ont.

La réciproque est naturellement fausse. Toutes les formes canoniques ont toutes les lettres dans leurs monômes.

2.c. Théorème

Si tous les monômes premiers de f ont tous la lettre a, il y a un ordre lexicographique optimal pour f commençant par a.

Montrons qu'à tout ordre 0 ne commençant pas par a nous pouvons faire correspondre un ordre 0_1 commençant par la lettre a, qui n'utilise pas plus de monômes et pas plus de lettres que 0.

D'après le théorème précédent (§2.b) nous savons que tous les monômes lexicographiques quelconques ont p variables identiques ($p \geq 0$) et dans le même ordre et la même p+1ème lettre.

Si la variable a^* fait partie des p premières il n'y a pas de problème à faire passer a^* en première position. Nous n'augmentons pas plus de nombre de monômes ni le nombre de lettres.

Si par contre a^* n'apparaît pas dans les p premières, elle ne peut qu'être à la p+1ème et figure sous les deux formes a et a'. La faire passer en première position n'accroît aucun des nombres précités.

Il est possible que ces deux nombres diminuent comme le montre l'exemple suivant

---

<u>Exemple</u>

f = abc + ab'e + a'b' de

dont la base complète est :

abc + ab'e + ace + a'b'de

un ordre 0 donnerait

eac + eab' + e'bac + e'b'a'd

tandis que 0_1 commençant par a donne

abc + ab'e + a'e'b'd

<u>Conséquence</u>

Si nous avons f(X) telle que tous ses monômes premiers ont la lettre a nous posons :

$$f(X) = f_1(a,Y)$$

et pour chercher l'écriture lexicographique optimale de f il suffira de chercher celles de

$$f_1(1,Y) \quad \text{et} \quad f_1(0,Y)$$

<u>Remarque</u>

Dans le théorème du §2.c. l'optimisation est valable quelque soit le coût (L,A ou $\mathcal{L}$).

3. ENCADREMENT DU NOMBRE DE MONOMES ET DU NOMBRE DE LETTRES DES SOLUTIONS L-OPTIMALES D'UNE FONCTION COMPLETE DE n LETTRES.

3.a. Encadrement

Soit une fonction f complète de n-lettres.

Soit E une écriture lexicographique (locale ou non) L-optimale de f et soient φ et Ψ respectivement une forme canonique et une base première irredondante de f ayant un nombre minimum de monômes et un coût en lettres minimum. Désignons par M_φ le nombre de monômes de φ, L_φ le coût en lettres de φ il vient :

(1) $\qquad M_\Psi \leq M_E \leq M_\varphi$

(2) $\qquad L_\Psi \leq L_E \leq L_\varphi$

en effet

- φ est une écriture lexicographique (non locale) particulière, il n'y en a pas de plus coûteuse tant pour M que pour L. Elle peut cependant être optimale.

- Ψ est une écriture de f qui comporte un nombre M et un coût L minima. Ψ est choisie à cette fin. Ψ n'est en général pas une écriture lexicographique.

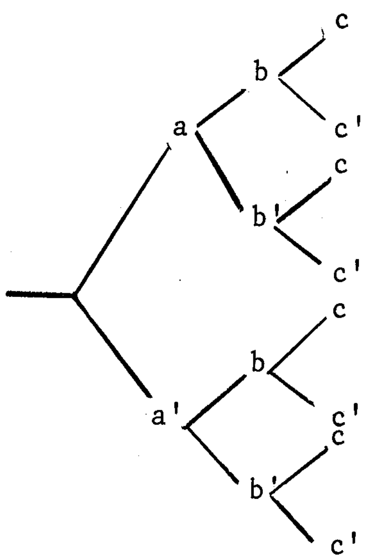
Exemple

$$f = abc + a'cd + a'b'c = \Psi$$
$$E = cba + cb'a' + cba'd$$
$$\varphi = abcd + abcd' + a'bcd + a'b'cd + a'b'cd'$$

(1) et (2) donnent $\quad 3 \leq 3 \leq 5 \quad$ et $\quad 9 \leq 10 \leq 20$

3.b. Détermination des bornes supérieures en fonction de n

φ ne peut pas avoir plus de 2^n monômes comme le montre l'exemple suivant pour n = 3



il faut que cette écriture lexicographique n'ait pas deux monômes ayant sur leurs n variables n-1 variables identiques et la dernière sous une forme chez l'un sous la forme complémentée chez l'autre, puisque nous voulons des bornes de formes irréductibles .

$$M_1 = 2^{n-1} \qquad \text{et} \qquad L_1 = n.2^{n-1}$$

pour n =4 $\quad M_1 = 8$ et $L_1 = 32$

Ceci peut ne pas paraître avantageux. Cependant les fonctions clés de parité et clés d'imparité atteignent effectivement ces bornes.

$$\varphi_1 = x_1 \otimes x_2 \otimes x_3 \dots \otimes x_n$$

égale d'ailleurs comme son nom l'indique à toutes les fonctions obtenues en complémentant un nombre pair de variables (un nombre impair correspond au complément de φ_1).

<u>Exemple</u>

$n = 4$

$f = a \otimes b \otimes c \otimes d$

$= abcd + a'bcd + ab'c'd + a'bc'd + a'bcd' + ab'cd' + abc'd' + a'b'c'd'$

où $f = E = \Psi = \varphi$

$M_1 = M_E = 8$ $\qquad$ $L_1 = L_E = 32$

Voici quelques valeurs de M_1 et L_1

n =	1	2	3	4	5	6	7
M_1 =	1	2	4	8	16	32	64
L_1 =	1	4	12	32	80	192	448

de (1) et de (2) on peut donner

$$\boxed{\begin{array}{ll} (1') & M_\Psi \leq M_E \leq \mathrm{Min}(M_\varphi, M_1) \\ (2') & L_\Psi \leq L_E \leq \mathrm{Min}(L_\varphi, L_1) \end{array}}$$

3.c. Exemple montrant que la borne supérieure est en général trop élevée.

$n = 6$

$\Psi = f = abc + b'd'e + ad'f + a'bd' + cd'f$ (réf 1-6)

$E = bac + bac'fd' + ba'd' + b'd'e + b'd'e'fa + b'd'e'fa'c$

$\varphi = abcdef + abcdef' + abcde'f + abcde'f' + abcd'ef + abcd'ef' +$
$abcd'e'f + abcd'e'f' + abc'd'e'f + ab'cde'f + ab'cd'ef + ab'cd'ef' +$
$ab'cd'e'f + ab'c'de'f + ab'c'd'ef + ab'c'd'ef' + ab'c'd'e'f +$
$a'bcd'ef' + a'bcd'e'f + a'bc'de'f + a'bc'd'ef + a'bc'd'ef' +$
$a'bc'd'e'f + a'bc'd'e'f' + a'b'cd'ef + a'b'cd'ef' + a'b'cd'e'f +$
$a'b'cd'e'f' + a'b'c'd'ef + a'b'c'd'ef'.$

pour cette fonction nous avons

$M_\Psi = 5$ $\qquad$ $M_E = 6$ $\qquad$ $M_\varphi = 30$ $\qquad$ $M_1 = 32$

$L_\Psi = 15$ $\qquad$ $L_E = 25$ $\qquad$ $L_\varphi = 180$ $\qquad$ $L_1 = 192$

Remarque. On peut abaisser L_φ et M_φ en remplaçant φ par φ_1 obtenue en appliquant à φ les réductions du type $A\alpha + A\alpha' = A$ jusqu'à ce que çà ne soit plus possible.

$$\varphi_1 = abc + ab'cde'f + ab'cde'f + ab'cd'e + ab'cd'e'f + ab'c'de'f + abc'd'e + ab'c'd'e'f + a'bcd'ef' + a'bcd'e'f + a'bc'de'f + a'bc'd' + a'b'cd' + a'b'c'd'e$$

où $M_{\varphi_1} = 14$ et $L_{\varphi_1} = 74$

$$\boxed{\begin{array}{ll} (1'') & M_\Psi \le M_E \le \mathrm{Min}(M_\varphi, M_{\varphi_1}, M_1) \\ (2'') & L_\Psi \le L_E \le \mathrm{Min}(L_\varphi, L_{\varphi_1}, L_1) \end{array}}$$

donnent

$5 \le 6 \le 14$ et $15 \le 25 \le 74$

4. SOMME DE MONOMES DISJOINTS ET FORME LEXICOGRAPHIQUE

4.a. Définition et propriétés

On dit que deux monômes sont disjoints lorsque leur produit est nul.

- On peut toujours écrire toute fonction sous la forme d'une somme de monômes disjoints, la forme canonique ou de Lagrange, la forme lexicographique locale irréductible, existant toujours, sont deux telles sommes.

- toute forme lexicographique de f est la somme de monômes disjoints égale à la fonction f. La réciproque n'est pas vraie comme le montre l'exemple suivant :

  $$f = ab' + bc' + ca'$$

  tous les monômes sont disjoints mais cette expression n'est pas lexicographique.

  En effet toutes les lettres a, b et c jouent le même rôle

  $$E = ab' + abc' + a'c + a'c'b$$

- toute somme de monômes disjoints n'est pas nécessairement lexicographique.
- parmi toutes les sommes de monômes disjoints il y a au moins une forme lexicographique locale irréductible.

4.b. Applications avec des vecteurs de vérité

A chaque monôme μ_i de l'écriture lexicographique $E = \sum_{i \in I} \mu_i$ on affecte w_i, vecteur de vérité à 2^n composantes si n est le nombre de lettres dont E dépend; on appelle de plus V celui de E

on a

- $V = \sum_{i \in I} w_i$
- $\forall i \in I \quad \forall j \in J \quad i \neq j \qquad w_i \cdot w_j = \emptyset$

où les sommations et produits sont ici les opérations entre deux vecteurs dont on fait les sommes et produits booléens composantes à composantes.

Vecteur de vérité. C'est un vecteur qui a autant de composantes que l'hypercube associé à la fonction a de sommets; chaque composante est égale à 1 si la fonction vaut 1 sur le sommet de l'hypercube correspondant, 0 autrement.

Exemple

$f(X) = abd + cd + ab'c \qquad \text{card}(X) = 4$

V(f) =	0	
	0	
	0	
	0	
	1	correspond à ab'cd'
	0	
	0	
	0	
	0	
	0	
	1	.... abc'd
	1	.... a'b'cd
	1	.... ab'cd
	1	.... a'bcd
	1	.... abcd

on peut lui faire correspondre l'ensemble des indices des composantes non nulles

$$\mathcal{E}(f) = \{5, 11, 12, 13, 14, 15\}$$

$$\mathcal{E}(abd) = \{11, 15\}$$

$$\mathcal{E}(ab'c) = \{5, 13\}$$

$$\mathcal{E}(a'cd) = \{12, 14\}$$

on peut obtenir les écritures des fonctions ayant peu de monômes en donnant tous les multiples d'un monôme m de la base complète de f et en les rangeant avec m dans une colonne appelée classe de m. A chacun de ces monômes μ_i on affecte l'ensemble $\mathcal{E}(\mu_i)$ précédemment défini. Nous devons maintenant faire avec ces monômes ou ces ensembles $\mathcal{E}(\mu_i)$ des partitions de f ou de l'ensemble $\mathcal{E}(f)$. Nous n'aurons pas nécessairement une écriture lexicographique, et devrons tester la lexicographicité de l'écriture polynomiale obtenue (cf. § 5). Reprenons l'exemple précédent

dc	12.13.14.15	dba	11.15	cb'a	5.13
dca	13.15	dcba	15	dcb'a	13
dca'	12.14	dc'ba	11	d'cb'a	5
dcb'	14.15				
dcb'	12.13				
dcba	15				
dcba'	14				
dcb'a	13				
dcb'a'	12				

pour réaliser les partitions nous découvrons la notion de composante indispensable ou sommet indispensable.

<u>Définition</u>.

On dit qu'un monôme m de la base complète de f(X) a un sommet indispensable s'il est le seul monôme à couvrir ce point sommet de l'hypercube à N dimensions si card(X) = N.

Ici, dc a 12 et 14 comme sommets indispensables
dba a 11 , cb'a a 15
on dira qu'un monôme de f est indispensable si tous les points qu'il couvre sont des sommets indispensables.

Propriété.

Un monôme indispensable est disjoint de tous les autres monômes de f.

Exemple

$f(X) = ab' + bc' + a'c$ a ses 3 monômes indispensables

ab'	1.5	bc'	2.3	a'c	4.6
ab'c	5	abc'	3	a'bc	6
ab'c'	1	a'bc'	2	a'b'c	4

la partition (1.5)(2.3)(4.6) de (1,2,3,4,5,6) ne donne pas une écriture lexicographique comme nous l'avons vu plus haut (§ 1) par contre

$b'a + bc' + bca' + b'a'c$ en est une.

5. RECHERCHE DE LA LEXICOGRAPHICITE D'UNE FONCTION φ DONNEE SOUS FORME POLYNOMIALE

5.a. Méthode

Soit $\varphi = \sum_{i \in I} \mu_i$ la fonction considérée.

le problème n'est possible que si l'on a

$$\forall i \in I \quad \forall j \in J \qquad i \neq j \qquad \mu_i \cdot \mu_j = 0$$

Sinon φ n'a même pas ses monômes disjoints, elle ne sera pas non plus lexicographique.

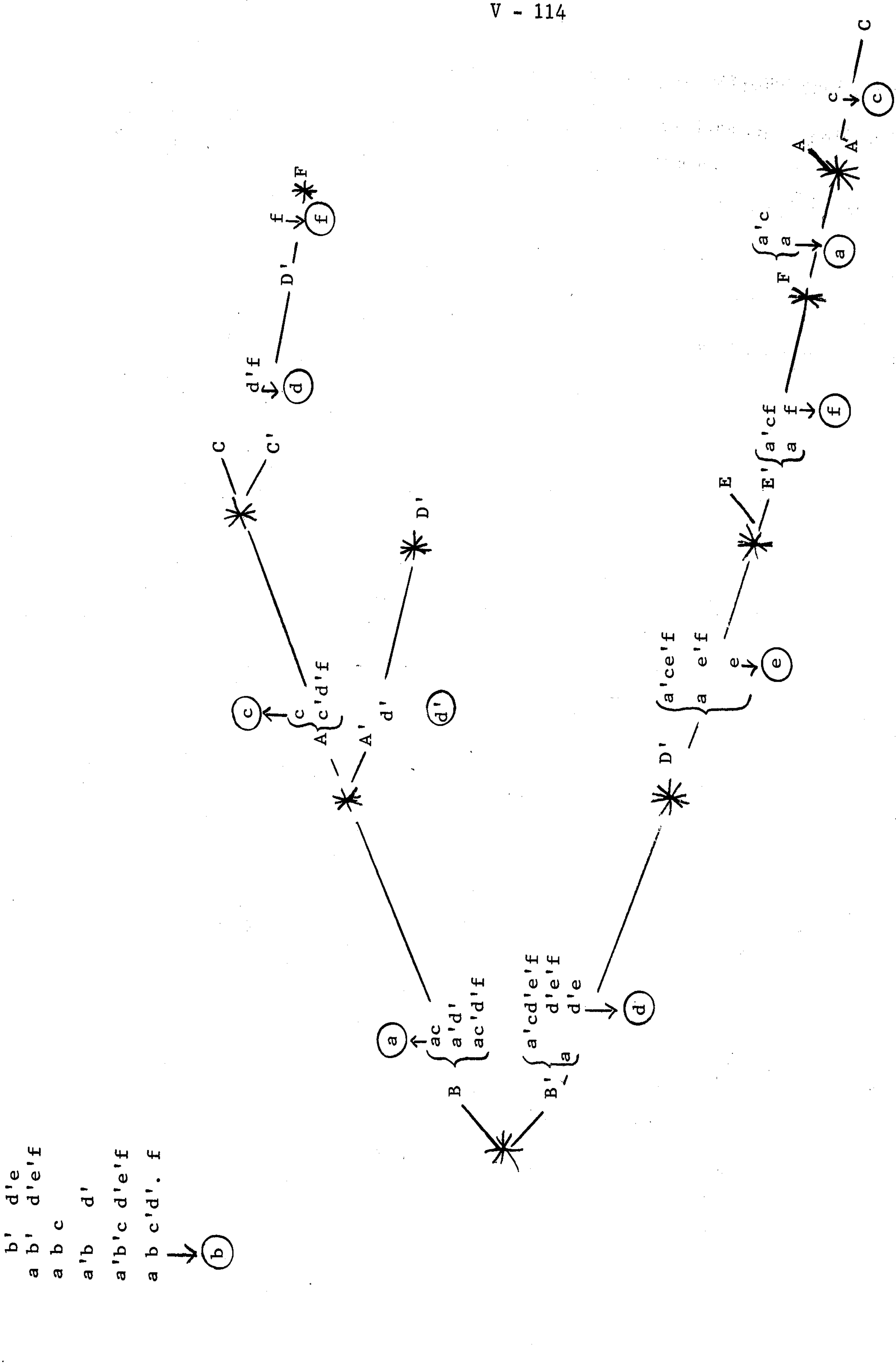
La recherche suivante consiste à donner essentiellement un ordre lexicographique s'il y en a ou aucun (on pourrait si l'on veut les chercher tous).

Ecrire φ sous forme lexicographique revient à déterminer les bifurcations éventuelles, donc la présence d'une lettre dans les monômes de l'arbre ou de sous-arbre considéré. Cette méthode est récursive et très facilement programmée. Elle consiste à chaque pas à chercher les racines possibles, à en envisager une jusqu'à ce qu'elle donne une solution sinon l'abandonner et reprendre à cet embranchement jusqu'à ce qu'on obtienne une partition de tous les monômes.

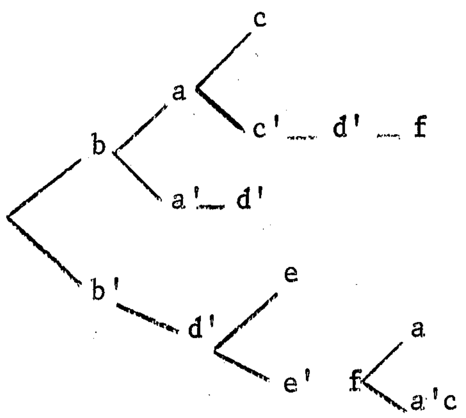
5.b. Exemple

$$\varphi = b'd'e + ab'd'e'f + abc + a'bd' + a'b'cd'e'f + abc'd'f$$

La solution s'extrait en lisant l'arbre formé par les majuscules



Solution



ou bac + bac'd'f + ba'd' +
b'd'e + b'd'e'fa + b'd'e'fa'c

Remarque

Lorsqu'il y a le choix entre une bifurcation et une adjonction nous prenons l'adjonction.

A N N E X E

LISTING DU PROGRAMME ALGOL

```
'DEBUT'
 'ENTIER' M,N,P,T,S,ALPHA,PAS,K ,I,J,K1,COMPT,IND,ICANON,TOPADR,TOPRES,
CPREM,AA,PDEF,NBSYNT,PDEG,CARSYNT,TOPMCA,TOPMON,N0,M0,P0,FACT,L0,
TOPFACT,  MALET,MALONG,NONMALONG,MADR,NONMADR,MANB,NONMANB,MACOUT,
TOPSYNT,TRACE,COMPTSOL,LONGSYNT,TAN ::
      'ENTIER' 'TABLEAU' RANG,NOM,MASQUE.(1:36).,F,G,H.(1:50).,
  MASKPOS.(1:7).,MAN1,MAN2 ,MAN3.(1:50).,NONMASK.(1:15). ::
   'BOOLEEN' ESSAI,OC,ARRET ::

'COMMENTAIRE' CODAGE A 01, A'10 , PAS DE A 00 , LE MONOME NUL  CONTIENT
 EN PARTICULIER  11  ::

         LIRE(N,TOPFACT,TOPMCA,TOPMON,TOPRES,TOPSYNT) ::
                 N0:= TG(262143,18) ::          P0:=2 ::
'POUR'I:=2 'PAS' 1 'A' N 'FAIRE' P0:=P0*2 ::
    L0:=(P0-1)'/'36+1 ::
   'SI' N 'SUPER' 7 'ALORS' 'DEBUT' FACT:=2 :: M0:=7 :: 'FIN'
                 'SINON' 'DEBUT' FACT:=1 :: M0:=N 'FIN' ::
'DEBUT' 'ENTIER''TABLEAU' FACTEUR.(1:2*N,0:TOPFACT).,MCANON.(1:L0,
 1:TOPMCA).,SYNTHESE.(1:FACT,0:TOPSYNT).,RESUL.(-1:FACT,0:TOPRES).,
 MON.(1:TOPMON).,  KUB,KUBDEF,KUBDEG.(1:L0). ,ADRESSE.(1:TOPMCA). ::





'PROCEDURE' RCH ::
'COMMENTAIRE' RETOUR A LA LIGNE ::
                                                                  'CODE'

'FCODE' ::

'PROCEDURE' LIGNE(A,B) :: 'VALEUR'A,B :: 'ENTIER'A,B ::
'COMMENTAIRE' DEFINI DEUX MARGES ENTRE LESQUELLES TOUS LES TEXTES
SUIVANTS SERONT ECRITS .
SI LA PROCEDURE N'EST PAS APPELEE ON ECRIT SUR TOUTE LA LARGEUR DE
LA FEUILLE ::
                                                                  'CODE'

'FCODE' ::

'PROCEDURE' SORCHAINE(I,J,X) :: 'VALEUR' I,J :: 'ENTIER' I,J,X ::
'COMMENTAIRE' ECRIT UN TEXTE DE I SYMBOLES DONT LE PREMIER EST DANS
LA MEMOIRE X EN POSITION J    . LE TEXTE S'IMPRIME A LA SUITE DES
TEXTES PRECEDENTS , S'IL Y A SUFFISEMENT DE PLACE. AUCUN ESPACE N'EST
PREVU .    ::
```

```
794   'FCODE' ::
796
796   'PROCEDURE' SORTEXTE(X) :: 'CHAINE' X ::                                  'CODE'
```

```
1058  'FCODE' ::
1060
1060  'PROCEDURE' SORENTIER(X) :: 'VALEUR'X:: 'ENTIER' X ::
1072                                                                             'CODE'
```

```
1265  'FCODE' ::
1267
1267
1267  'ENTIER' 'PROCEDURE' REUNION(F,M) :: 'VALEUR' M ::
1279     'ENTIER' M :: 'ENTIER' 'TABLEAU' F ::
1286  'COMMENTAIRE ' CETTE PROCEDURE PRENDS L'UNION LOGIQUE DES
1286  M PREMIERES MEMOIRES DU TABLEAUF ::
1286                                                                             'CODE'
```

```
1419  'FCODE' ::
1421
1421
1421
1421
1421  'PROCEDURE' ECRIRDCB(A) :: 'ENTIER' A ::
1430  SORCHAINE(1,6,A) ::
1439
1439
1439    'PROCEDURE' PREPARE1(N) :: 'VALEUR' N :: 'ENTIER' N ::
1451  'COMMENTAIRE'
1451  CETTE PROCEDURE FABRIQUE LE TABLEAU DES MASQUES .
1451  ELLE PREPARE LA MACHINE A TRAITER DES PROBLEMES DE N VARIABLES
1451  ELLE LIT PAR EDONNEE 2*N DONNEES :
1451  CHAQUA GROUPE DE 2 CHIFFRES CORRESPONDAS A UNE VARIABLE :
1451  LE PREMIER CHIFFRE EST LE RANG BINAIRE (COMPRIS ENTRE 1 ET 18) DE LA
1451  VARIABLE, LE DEUXIEME CHIFFRE EST LE CODE D.C.B DU NOM DE LA VARIABLE ::
1451      'DEBUT' 'ENTIER' I,S ::
```

```
1535     'FIN' PREPARE1 ::
1537
1537  'PROCEDURE' LECT1(F,M,N) :: 'VALEUR' N :: 'ENTIER' M,N ::
1555     'ENTIER' 'TABLEAU' F :: 'COMMENTAIRE' CETTE PROCEDURE LIT
1559                     UNE EXPRESSION POLYNOMIALE  ET LA RANGE DANS LE
1559  TABLEAU F .LES UNIONS SONT REPRESENTEES PAR LE SYMBOLE +  ,
1559  LES COMPLEMENTS PAR LE SYMBOLE ' SUIVANT IMMEDIATEMENT UNE LETTRE,
1559  LES INTERSECTIONS NE SONT PAS REPRESENTEES .
1559  L'EXPRESSION LUE NE DOIT PAS COMMENCER PAR + , ET DOIT SE TERMINER
1559  PAR LE SYMBOLE .  ::
1559  'DEBUT' 'ENTIER' T,I,J,G :: 'ENTIER' 'TABLEAU' C.(1:73). ::

1811  'FIN' LECT1 ::
1813
1813  'PROCEDURE' EXTRF1(F,M,N,LIGNE) ::
1825     'VALEUR' M,N,LIGNE :: 'ENTIER' M,N,LIGNE ::
1839     'ENTIER' 'TABLEAU' F ::
1843  'DEBUT' 'ENTIER' CAR,LIBRE,TAU,I,J,A,B,LA  ::
1861  'COMMENTAIRE'   IMPRIME EN LITTERAL  ENTRE LES POSITIONS 1 ET LIGNE ,
1861  LE POLYNOME DE N VARIABLE ET DE M MONOMES  SITUE DANS LE TABLEAU F  ::
1861     CAR:=0 :: LIBRE:=LIGNE-2*N ::
1873  RCH ::

2076  STOP : 'FIN' EXTRF1 ::
2080
2080
2080
2080  'PROCEDURE' INCLURE1(F,J,A) ::
2090     'VALEUR' A :: 'ENTIER' A,J :: 'ENTIER' 'TABLEAU' F ::
2102  'COMMENTAIRE'
2102  AJOUTE AU POLYNOME DE J MONOMES SITUE EN F , LE MONOME A .
2102  SI LE POLYNOME ETAIT IRREDONDANT, ON OBTIENT ENCORE UN POLYNOME
2102  IRREDONDANT   ::
2102     'DEBUT' 'ENTIER' B,C,I,K :: 'BOOLEEN' RECUP ::

2272     STOP: 'FIN' INCLURE1 ::
2276     'PROCEDURE' REDUC1(F,M) :: 'ENTIER' M :: 'ENTIER' 'TABLEAU' F ::
2291  'COMMENTAIRE'   SUPPRIME LES MONOMES REDONDANTS DU POLYNOME  DE M
2291  MONOMES SITUE DANS LE TABLEAU F  ::
```

```
    'DEBUT' 'ENTIER' I,J ::
       J:=1 ::
       'POUR' I:=2 'PAS' 1 'JUSQUA' M 'FAIRE'
    INCLURE1(F,J,F.(I).) :: M:=J
    'FIN' REDUC1 ::


'PROCEDURE' COMPLEMENT(F,  M,N,G,P) ::
'VALEUR'   M,N :: 'ENTIER '   M,N,P ::

'FIN' COMPLEMENT ::



'PROCEDURE' DUALE1(F,M,N,G,P,DC,MAX) ::
'VALEUR' M,N,MAX ::
'ENTIER'M,N,P,MAX :: 'BOOLEEN' DC  ::
'ENTIER ' 'TABLEAU ' F,G ::
'COMMENTAIRE'
DUALISE LA FONCTION F DE N VARIABLES ET M MONOMES . G EST LE DUAL , P LE
NOMBRE DE MONOMES. LE BOOLEEN  DC SERA FAUX SI EN COURS DE CALCUL IL Y
A DEPASSEMENT DE LA CAPACITE MAX DU TABLEAU G  ::
   'DEBUT' 'ENTIER' S,I,Q,P1,P2,J,K,R,L ::

'FIN' DUALE1 ::



'PROCEDURE' PRODUIT(F1,M1,F2,M2,G,M) ::
                        'ENTIER' M1,M2,M ::
  'ENTIER''TABLEAU' F1,F2,G ::
'COMMENTAIRE' FAIT LE PTODUIT DE M1 MONOMES DE F1 AVEC LES M2 MONOMES DE
'COMMENTAIRE' FAIT LE PRODUIT DE M1 MONOMES DE F1 AVEC LES M2 MONOMES DE
 F2 ET LES RANGE DANS DANS G SUR M MONOMES ::
    'DEBUT' 'ENTIER' I,J,K,A,B,C ::

  'FIN' PRODUIT ::

'PROCEDURE' COMPREM1(F,M,N,MAX,DC) :: 'VALEUR' N,MAX ::
'ENTIER' M,N,MAX ::
'ENTIER' 'TABLEAU' F ::
'BOOLEEN' DC ::
```

```
3038  'COMMENTAIRE'
3038  CALCUL TOUS LES COMPOSANTS PREMIERS DE LA FONCTION DE M MONOMES
3038  ET DE N VARIABLES SITUEE DANS LE TABLEAU F .
3038       LE BOOLEEN DC SERA VRAI SI LA CAPACITE MAXIMUM MAX  DU TABLEAU F
3038  EST DEPASSEE .
3038     EN CAS DE DEPASSEMENT DE CAPACITE , ON OBTIENT DANS F UNE BASE
3038  QUELCONQUE .
3038     ATTENTION   UN TABLEAU AUXILIAIRE DE MAX MEMOIRES EST UTILISE DANS
3038  LA PROCEDURÉ .  ::
3038     'DEBUT' 'ENTIER' I,J,S,I1,I2,P,Q,R,N1,N2 ::
3060        'ENTIER' 'TABLEAU' AUX.(1:MAX). ::

3324     STOP:  'FIN' COMPREM1 ::
3328
3328
3328
3328  'ENTIER''PROCEDURE' VALIRE(RESUL,INDICE,POSITION) ::
3339  'VALEUR' INDICE,POSITION ::
3344  'ENTIER' INDICE,POSITION :: 'ENTIER' 'TABLEAU' RESUL ::
3353  'COMMENTAIRE' LIT LA MEMOIRE DE LA POSITION 1 A DROITE A 7 DE 5 BITS
3353   CHACUNE DANS LE TABLEAU RESUL ::
3353  'DEBUT' 'ENTIER' MASK ,POS,UN ::
3361                                          'SI' POSITION 'INFEG' 7 'ALORS'
3366      'DEBUT' POS:=POSITION :: UN:=1 :: 'FIN' 'SINON' 'DEBUT' POS:=
3380      POSITION  - 7 ::UN:=2 :: 'FIN' ::
3390    DERDEDER : MASK:= MASKPOS.(POS). ::
3399       MASK := ET(RESUL.(UN,INDICE).,MASK) ::
3413   VALIRE := TD(MASK,5*(POS-1)) ::
3428  'FIN' VALIRE ::
3430
3430
3430  'PROCEDURE' VAPLACER(RESUL,INDICE,POSITION,NB) ::
3442  'VALEUR' INDICE,POSITION ::        'ENTIER''TABLEAU'
3449     RESUL :: 'ENTIER' INDICE ,POSITION,NB ::
3458  'DEBUT''ENTIER' MASK,POS,UN ::
3466                                               'SI' POSITION 'INFEG'7
3469   'ALORS''DEBUT' POS:=POSITION ::UN:=1 :: 'FIN' 'SINON' 'DEBUT'
3483    POS:=POSITION-7 :: UN:=2 :: 'FIN' ::
3495   DERNIERE : MASK:=MASKPOS.(POS). ::
3504  RESUL.(UN,INDICE).:=OU(RESUL.(UN,INDICE).,TG(NB,5*(POS-1))) ::
3534  'FIN' VAPLACER ::
3536
3536
3536  'PROCEDURE' SORTPERM1(RESUL,PAS,PLACE,LIGNE) :: 'VALEUR' PAS,
3551  PLACE,LIGNE ::'ENTIER''TABLEAU' RESUL :: 'ENTIER' PLACE,LIGNE ,
3564  'COMMENTAIRE' SORT LES G(Y) DU TABLEAU RESUL,CHAQUE MONOME AYANT SON
3564  ORDRE EFFECTIF ET NON L ORDRE CANONIQUE .IL EST A NOTER QUE CETTE
3564  ECRITURE EST L INVERSE DE CELLE HABITUELLE , LA DERNIERE LETTRE DE
3564  CHAQUE MONOME EST LA RACINE DE L ARBORESCENCE ASSOCIEE ::
3564    PAS ::
```

```
'DEBUT' 'ENTIER' CAR,LIBRE,LA,A,B,I,J,K,NOMB ::
      'SI' RESUL.(1,PLACE). =0 'ALORS'
                 'ALLERA' STOP ::
RCH::'POUR' K:=0 'PAS' 1 'A' LIGNE'FAIRE' 'DEBUT'
             A:= RESUL.(1,K+PLACE).:: 'SI' A=0 'ALORS' 'DEBUT'
  ECRIRDCB(1) :: 'ALLERA' STOP :: 'FIN' ::
      'POUR'J:=1 'PAS' 1 'A' PAS 'FAIRE'
'DEBUT' B:=VALIRE(RESUL,PLACE+K,J) ::
     'SI' B=0 'ALORS' 'ALLERA' PASSA ::
     LA:= ET(B,15) :: B:=ET(B,16) :: LA:=NOM.(LA). ::
      ECRIRDCB(LA) ::
      'SI' B=16 'ALORS'        ECRIRDCB(12) ::
      PASSA :
  'FIN' :: 'SI' K 'N=' LIGNE'ALORS' 'DEBUT' ECRIRDCB(48) ::
      ECRIRDCB(16) ::
      ECRIRDCB(48) :: 'FIN' ::
   'FIN' ::
STOP : RCH :: 'FIN' SORTPERM 1 ::

 'PROCEDURE' PLACERNB ( ADR,NB,NOMB,PLACE,RESUL) :: 'VALEUR'  ADR, NB,
NOMB,PLACE :: 'ENTIER' 'TABLEAU' RESUL ::
'ENTIER' NOMB,NB,PLACE,ADR ::
'DEBUT' 'ENTIER' A,B ::
     A:=TG(NOMB,15)  :: B:=ET(RESUL.(0,PLACE).,MALET) ::
     A:= OU(TG(NB,20),A) ::    A:=OU(TG(ADR,25),A) ::
     RESUL.(0,PLACE). := OU(A,B) ::
'FIN' PLACERNB  ::

'PROCEDURE' PLACERNB1(NOMB,PLACE) :: 'VALEUR' NOMB,PLACE ::
'ENTIER' NOMB,PLACE ::
'DEBUT' 'ENTIER' A,B ::
A:=TG(NOMB,15) ::B:=ET(RESUL.(0,PLACE).,NONMALONG) ::
  RESUL.(0,PLACE).:=OU(A,B) ::
'FIN' PLACERNB 1 ::

'PROCEDURE' PLACERNB2(NOMB,PLACE) :: 'VALEUR' NOMB,PLACE ::
'ENTIER' NOMB,PLACE ::
'DEBUT' 'ENTIER' A,B ::
 A:= TG(NOMB,20) :: B:=ET(RESUL.(0,PLACE).,NONMANB) ::
     RESUL.(0,PLACE). := OU(A,B) ::
'FIN' PLACERNB1 ::

'PROCEDURE' PLACERADR(NOMB,PLACE) :: 'VALEUR' NOMB,PLACE ::
'ENTIER' NOMB,PLACE ::
'DEBUT' 'ENTIER' A,B ::
 A:= TG(NOMB,25) :: B:=ET(RESUL.(0,PLACE).,NONMADR) ::
     RESUL.(0,PLACE). := OU(A,B) ::
'FIN' PLACERADR ::

'ENTIER''PROCEDURE' LIRENB1(PLACE) :: 'VALEUR' PLACE ::
'ENTIER' PLACE ::
LIRENB1:=TD(ET(RESUL.(0,PLACE).,MALONG),15) ::

'ENTIER''PROCEDURE' LIRENB2(PLACE) :: 'VALEUR' PLACE ::
'ENTIER' PLACE ::
```

```
4087  LIRENB2:=TD(ET(RESUL.(0,PLACE).,MANB ),20) ::
4106
4106  'ENTIER''PROCEDURE' LIREADR(PLACE) :: 'VALEUR' PLACE ::
4116  'ENTIER' PLACE ::
4119  LIREADR:=TD(ET(RESUL.(0,PLACE).,MADR ),25) ::
4138
4138
4138
4138
4138  'PROCEDURE' VERITE2(FONC,UN,M,N,P,KUB,ABS) ::
4156    'VALEUR' ABS,UN,M,N ::
4165  'ENTIER' 'TABLEAU' FONC,KUB:: 'ENTIER' M,N,UN,P,ABS ::
4182  'COMMENTAIRE' CHERCHE LES POINTS DE L'HYPERCUBE BOOLEEN COUVERTS PAR LA
4182  FONCTION FONC DE M+1 MONOMES. LE RESULTAT EST CODE DE 1 A 2
4182  PUISSANCE N LE ZERO ETANT CADRE A DROITE ...
4182     ENSUITE .. EXTRACTION DU VECTEUR DE VERITE LE TABLEAU KUB.LES POIDS
4182   FAIBLES DANS KUB(1) 0 A DROITE LES POIDS FORTS DANS KUB(LO)LE PLUS
4182    FORT A GAUCHE,ABS 2 FOIS LES LETTRES PERMISES ::
4182  'DEBUT' 'ENTIER' I,J,A,B,C,D ::
4196          D:=TG( ABS ,18) ::
4205          'POUR' J:=1 'PAS' 1 'A' LO 'FAIRE' KUB.(J).:=0 ::
4221            'POUR' J:=0 'PAS' 1 'A' TRACE 'FAIRE'
4230          'SI' ET(J,ABS) =J 'ALORS'
4240          'DEBUT'  A:=OU(J,DJ(TG(J,18),D)) ::
4260           'POUR' I:=0 'PAS' 1 'A' M 'FAIRE' 'SI' OU(A,FONC.(I+UN).)
4281      =A 'ALORS' 'DEBUT' B:=J'/'36+1 ::  C:=J-36*(B-1)+1 ::
4307          KUB.(B).:=OU(KUB.(B).,MASQUE.(C).) ::
4325          'ALLERA' TOSSA
4326        'FIN' ::
4329   TOSSA : 'FIN' ::
4333        P:=0 :: 'POUR' J:=1 'PAS' 1 'A' LO 'FAIRE'
4346        P:= P+ POIDS(KUB.(J).) ::
4358  'FIN' VERITE1 ::
4360
4360
4360
4360
4360
4360  'PROCEDURE' PRESENCE(F,M,N,K1,INDMON,G,P,INDCAN) ::
4380  'VALEUR' M,N,P ::
4387  'ENTIER' 'TABLEAU' F , G ::
4393  'ENTIER' M,N,K1,INDMON,P,INDCAN ::
4406  'COMMENTAIRE' DONNE DE 1 A K1 DANS LE TABLEAU RESUL LES G(Y) D UNE
4406  LETTRE. CE SONT D AILLEURS  LES VARIABLES FIGURANT EFFECTIVEMENT
4406  DANS LA FONCTION ::
4406  'DEBUT' 'ENTIER' S,I,J,K,R,Q,L,T,U,RAN,A,IK ::
4432  'BOOLEEN' RIEN,V ::
4437        SORTEXTE('' PRESENCE '') ::  RCH ::
4447              S:=REUNION(F,M) ::
4456     'POUR' J:=1 'PAS' 1 'A' N 'FAIRE'
4465        'DEBUT' R:=NONMASK.(J). ::
4473        'POUR' I:=0,16 'FAIRE'
4480           'DEBUT' K:= 'SI' I=0 'ALORS' 0 'SINON' 18 ::
4492                         Q:=MASQUE.(K+J). ::
4501         A:=ROT(Q,18) ::
```

```
            'SI' ET(S,Q)=Q 'ALORS'
               'DEBUT'
'COMMENTAIRE' ON PEUT CHANGER LE COUT ::
  MONOBIFUR(1,0,Q, A    ,INDMON,R,K1,0,M-1,P+1,P-1,1,1,INDCAN,RIEN,V) :
          'SI' RIEN 'ALORS' 'DEBUT'
             RESUL.(0,K1).:=OU(RESUL.(0,K1).,R) ::
                                       RESUL.(1,K1).:=   OU(J,I) ::
          'POUR' IK :=2 'PAS' 1 'A' FACT 'FAIRE'  RESUL.(IK,K1).:=0 ::
                                  'FIN' ::
          'SI' V 'ALORS' 'DEBUT' LONGSYNT:=0 ::
                SYNTHESE.(1,0).:=OU(J,I) ::
                               'FIN' ::
             'FIN' ::
           'FIN' ::
        'FIN' ::
'FIN' PRESENCE ::



'PROCEDURE' DERIVATION(F,M,N,IND1,G,P,LET,ABS) ::
'VALEUR'       ABS,LET,M,N,IND1 ::
'ENTIER' 'TABLEAU' F,G :: 'ENTIER' M,N,IND1,P,ABS, LET ::
'COMMENTAIRE' DERIVE DANS LE TBBLEAU F DEPUIS IND1 M+1 MONOMES PAR
RAPPORT A LET ET RANGE DS LE TABL G LES P+1 MONOMES OBTENUS.NOTER
QUE LET EST UN MASQUE  ::
'DEBUT' 'ENTIER' A,B,I,U,Q,K,J,L,GA ::
              G.(1). :=0 ::
B:=NG(LET) :: Q:=ROT(LET,18) ::
   U:=OU(LET,Q) :: J:= 0 ::
   'POUR' I:=0 'PAS' 1 'A' M 'FAIRE'
      'DEBUT' K:=IND1+I ::
         L:=F.(K). :: A:=ET(L,U) :: 'SI' A=U 'OU' A=Q
         'ALORS' 'ALLERA' PALUI ::
             GA:= 'SI' A=LET 'ALORS' ET(L,B) 'SINON' L ::
                                 'SI' ET(GA,ABS) 'N=' GA
               'ALORS' 'ALLERA' PALUI ::
           'SI' GA=0 'ALORS' 'DEBUT' P:=1 :: G.(1).:=0 ::
              'ALLERA' CETERMINE ::
                                 'FIN' ::
      'SI' J=0 'ALORS' 'DEBUT' G.(1).:=GA :: J:=1 ::
                                 'ALLERA' PALUI ::
                         'FIN' ::
         INCLURE1(G,J,GA) ::
          PALUI:
       'FIN' ::
  P:=J ::
CETERMINE  :
'FIN' DERIVATION ::


'PROCEDURE' MONOBIFUR(IA,IB,LET,COMPL,IND,ABS,INDICE,NOMB,
  B,A,B1,A1,COUT,ICANON,CEBON,ILYASOL) ::
'VALEUR' IA,IB,LET,COMPL,ABS,NOMB,B,A,A1,B1,COUT::
                          'ENTIER' IA,IB,LET,COMPL,IND,ABS,
INDICE,NOMB,B,A,B1,A1,COUT,ICANON ::
```

```
4983  'BOOLEEN' CEBON,ILYASOL ::
4988  'COMMENTAIRE' ON A FAIT UNE BIFURCATION SUR LES INDICES IA ET IB AVEC LA
4988  LETTRE DT LE MASQUE EST LETET SON COMPLEMENT COMPL . ON CHERCHE LES MONO
4988  RELATIFS A CETTE BIFUR N AYANT QUE DES LETTRES DS ABS(ON A ENLEVE
4988  LET A ABS) . LE TOUT EST RANGE A PARTIR DE MON(IND+1) ::
4988  'DEBUT' 'ENTIER' C,D,E,F,I,W ::
5002  'ENTIER' F1,IC,G1,A2,J,C1,G,D1,E1,U,V,GARIND,K ::
5029  'ENTIER' 'TABLEAU' MC,MD.(1:LO). ::
5040                                              E:=OU(ABS,TG(ABS,18)) ::
5054            MAN3.(1).:=MAN2.(1).:=0 ::
5066     DERIVATION(MON,B,N,A,MAN3,F,  LET,E) ::
5085     DERIVATION(MON,B1,N,A1,MAN2,D,  COMPL,E) ::
5104              ILYASOL := 'FAUX' ::
5108  OPTISOL1 :
5110    'SI' 'NON'(MAN3.(1). =0 'ET' MAN2.(1).=0 'ET' F=1 'ET' D=1)'ALORS'
5136              'ALLERA' RECHMULT ::
5139            'SI' CARSYNT 'SUPER' COUT 'ALORS'
5144             'DEBUT'  CARSYNT := COUT ::
5149                           RCH ::   SORTEXTE('' SOLUTION '') ::
5169    SORTEXTE('' DE COUT '') :: SORENTIER(COUT) ::
5172              ILYASOL := 'VRAI' ::
5176             'FIN' ::
5178             'ALLERA' NUL ::
5181  RECHMULT  :
5183            'SI' PAS=N 'ALORS' 'ALLERA' NUL ::
5191       'SI' MAN3.(1).=0 'ET' F=0 'OU' MAN2.(1).=0 'ET' D=0 'ALORS'
5214             'ALLERA' NUL ::
5217     PRODUIT(MAN3,F  ,MAN2,D  ,MAN1,IC) ::
5232        'SI' MAN1.(1).=0 'ET' IC=0 'ALORS' 'ALLERA' NUL ::
5247        VERITE2(MAN1,1,IC-1,N,C1,MC,ABS) ::
5266  OPTISYNT1  :
5268
5268             'POUR' I:=1 'PAS' 1 'A' ICANON 'FAIRE'
5277             'DEBUT' C:=ADRESSE.(I). ::  F:=ET(C,TRACE) ::
5294                  G:=TD(ET(C,MACOUT),13) ::
5308                  D:=TD(ET(C,MANB),20) ::
5322                  'SI' F=ABS 'ET' D=C1   'ALORS'   'DEBUT'
5332                      'POUR' J:=1 'PAS' 1 'A'LO 'FAIRE'
5341         'SI' MCANON.(J,I). 'N=' MC.(J). 'ALORS' 'ALLERA' OTRI ::
5357           'SI' G 'SUPER' COUT 'ALORS'
5362                  'DEBUT'
5363              U:=TD(C,25) ::
5372             D1:=LIRENB1(U) ::
5379             'POUR' J:=-1 ,1 'PAS' 1 'A' FACT 'FAIRE'
5391              'POUR' K:=U 'PAS' 1 'A' U+D1'FAIRE'
5402                   RESUL.(J,K). := 0 ::
5411                        'ALLERA' BON1::
5414                 'FIN' ::
5416             'ALLERA'  NUL ::
5419                                                    'FIN' ::
5421  OTRI  :
5423              'FIN' ::
5425              'ALLERA' BON1::
5428  NUL: CEBON := 'FAUX' ::
5434              'ALLERA' MOVAI1::
```

```
5437 BON1 :
5439       INDICE:=INDICE+1 :: CEBON := 'VRAI' ::
5449         ICANON := ICANON+1 ::
5455          W:= ABS ::
5459         W := OU(W,TG(  COUT,13)) ::
5473         W := OU(W,TG(     C1,25)) ::
5487         W := OU(W,TG(INDICE,25)) ::
5501         ADRESSE.(ICANON).:= W ::
5508    PLACER NB(IND+1,IC-1,NOMB,INDICE,RESUL) ::
5525          'POUR' I:=1 'PAS' 1 'A' IC 'FAIRE' 'DEBUT'
5535         IND:=IND+1 ::
5541            MON.(IND).:=MAN1.(I). ::'FIN' ::
5553         'POUR' J:=1 'PAS' 1 'A' LD 'FAIRE'
5562         MCANON.(J,ICANON). := MC.(J). ::
5574         RESUL.(-1,INDICE).:=OU(  TG(ABS,17),COUT) ::
5594 MOVAI1 :
5596 'FIN' MONOBIFUR ::
5598
5598
5598
5598
5598 'PROCEDURE' BIFURCATION(PAS,INDINF,INDSUP,N,COMPT,IND) ::
5614 'VALEUR' PAS,INDINF,INDSUP,N ::
5623 'ENTIER' PAS,INDINF,INDSUP,N,COMPT,IND ::
5636 'COMMENTAIRE' CONSTRUIT UNE BIFURCATION DU TYPE A.. + A'... AVEC LES
5636 INDICES IA ET IB CF A LA FOIS ADJONCTION ET MONO BIFUR ::
5636 'DEBUT' 'ENTIER' UN,DE,I,A,COMT,J,C,B,JJ,KK,H,L,K,P,J1,S,IL,COUT,
5674         A1,H1,L1,NOMB,NC,NB, COIND,IA,COUTOTAL,MAX,INT,IK ::
5698 'BOOLEEN' OMENTE,CETUNESOL ::
5703         RCH  ::
5705      SORTEXTE('' ----------- '') :: RCH ::
5716      SORTEXTE('' BIFURCATION '') :: SORENTIER(COMPT) ::
5730  SORTEXTE('' '') :: SORENTIER(IND) :: RCH ::
5744      SORTEXTE('' ----------- '') :: RCH ::
5755    I:=INDINF ::
5759 BOUCLE:
5761      COMT:=LIRENB1(I) ::
5768  'SI' I'INFEG' 1 'ALORS' 'ALLERA' CONTINUE ::
5776        'SI' RESUL.(1,I).=0 'ALORS' 'ALLERA' PACELECI ::
5789 CONTINUE  :
5791    'DEBUT' 'ENTIER' 'TABLEAU' ATTENT.(0:FACT,0:COMT+1). ::
5807        A:=RESUL.(-1,I). ::  COUT:=ET(A,127  ) :: A:=TD(A,17) ::
5835      'POUR' J:= 1   'PAS' 1 'A' N 'FAIRE'
5844      'DEBUT' C:=MASQUE.(J). :: B:=MASQUE.(J+18). ::
5861         'SI' ET(A,C) 'N=' C 'ALORS' 'ALLERA' NICELECI ::
5874        NB:=NONMASK.(J). :: NC:=ET(A,NB) ::
5890                MAX := OU(B,C) ::
5899         'POUR' JJ:=C,B 'FAIRE'
5906         'DEBUT' 'SI' JJ=C 'ALORS'
5912            'DEBUT' KK:=B :: J1:=0
5919            'FIN' 'SINON'
5922            'DEBUT' KK:=C :: J1:=16  ::
5931            'FIN' ::
5933            H:=LIRENB2(I) ::IL:=LIREADR(I) ::
5947            'POUR' K:=0 'PAS' 1 'A' H 'FAIRE'
```

```
5956              'DEBUT' P:=IL+K::  INT:=ET(MAX,MON.(P).) ::
5975           'SI' INT 'N=' KK 'ET' INT 'N=' MAX
5982              'ALORS' 'ALLERA' CETEXA ::
5987              'FIN' ::
5989              'ALLERA' PACELELA ::
5992  CETEXA: 'POUR' K:=0 'PAS' 1 'A' COMT 'FAIRE'
6003        'DEBUT' 'POUR' L:=1 'PAS' 1 'A' FACT 'FAIRE'
6013        'DEBUT' ATTENT.(L,K).:=RESUL.(L,K+I). ::
6030              VAPLACER(ATTENT,K,PAS,J+J1) ::
6043        'FIN' :: ATTENT.(0,K).:=ET(RESUL.(0,K+I).,NB) ::
6066        'FIN' :: L:=0 ::
6072  REBOUCLE:                  S := LIRENB1(L) ::
6081      'SI' L=I 'ALORS' 'ALLERA' NICELELA ::
6089         'SI' L 'INFEG' 1 'ALORS' 'ALLERA' CETUN ::
6097        'SI' RESUL.(1,L).=0 'ALORS' 'ALLERA' NICELELA ::
6110  CETUN  :
6112          DE:=RESUL.(-1,L). ::  A1 := TD(DE,17) ::
6131           'SI' ET(A1,C)=C 'ALORS'
6141              'DEBUT' H1:=LIRENB2(L) ::
6149                L1:=LIREADR(L) ::
6156                'POUR' K:=0 'PAS' 1 'A' H1 'FAIRE'
6165                'DEBUT' P:=L1+K :: INT:=ET(MAX,MON.(P). ) ::
6184         'SI' INT 'N='MAX 'ET' INT 'N=' JJ
6191                'ALORS' 'ALLERA' CEJUSTE ::
6196                'FIN' ::
6198                'ALLERA' NICELELA ::
6201  CEJUSTE:
6203              IA := ET( NC,A1) ::
6212         'SI' L=0 'ALORS' 'DEBUT' NOMB:=COMT ::
6222  'COMMENTAIRE' ON PEUT CHANGER LE COUT ::
6222          COUTOTAL := COUT+1+COMT ::
6230             'ALLERA' CEZERO 'FIN' ::
6234          COUTOTAL:=ET(DE,127) ::
6243       NOMB:=COMT+S+1 ::
6251  'COMMENTAIRE' ON PEUT CHANGER LE COUT ::
6251          COUTOTAL := COUTOTAL + COUT + 1+NOMB ::
6261  CEZERO  :
6263       'SI' COUTOTAL 'SUPEG' CARSYNT 'ALORS' 'ALLERA' NICELELA ::
6271  MONOBIFUR(I,L,JJ,KK,IND,IA,COMPT,NOMB,H,IL,H1,L1,COUTOTAL,
6299  ICANON,OMENTE,CETUNESOL) ::
6306              'SI' OMENTE      'ALORS' 'DEBUT'
6310        'POUR' K:=0 'PAS' 1 'A' COMT 'FAIRE'
6319    'DEBUT'  'SI' K 'N=' 0 'ALORS' COMPT:=COMPT+1 ::
6331        'POUR' IK:=0 'PAS' 1 'A' FACT 'FAIRE'
6340       'SI' K=0 'ET' IK=0 'ALORS'
6349       RESUL.(0,COMPT).:=OU(RESUL.(0,COMPT).,ATTENT.(0,0).)  'SINON'
6373        RESUL.(IK,COMPT    ).:=ATTENT.(IK,K). ::
6387    'FIN' ::
6389       'SI' L=0 'ALORS' 'ALLERA' CELEZERO ::
6397        'POUR' K:=0 'PAS' 1 'A' S 'FAIRE'
6406    'DEBUT'         COMPT:=COMPT+1 ::
6413          'POUR' IK:=1 'PAS' 1 'A'  FACT 'FAIRE'
6422        'DEBUT'
6423        RESUL.(IK,COMPT    ).:=RESUL.(IK,L+K). ::
6439        VAPLACER(RESUL,COMPT     ,PAS,J+16-J1) ::
```

```
            'FIN' ::
            RESUL.(D,COMPT    ).:=ET(RESUL.(J,L+K).,NB) ::
     'FIN' ::
                                              'FIN' ::
 'SI' CETUNESOL 'ALORS'  'DEBUT'
             'POUR' K:=0 'PAS' 1 'A' COMT 'FAIRE'
          'POUR' IK:=1 'PAS' 1 'A' FACT 'FAIRE'
          SYNTHESE.(IK,K).:=ATTENT.(IK,K). ::    COIND:=COMT  ::
          'SI' L=0 'ALORS' 'ALLERA' CELEZERO ::
             'POUR' K:=0 'PAS' 1 'A' S    'FAIRE'
        'DEBUT'   COIND:=COIND+1 ::
          'POUR' IK:=1 'PAS' 1 'A' FACT 'FAIRE'
               'DEBUT'  SYNTHESE.(IK,COIND).:=RESUL.(IK,L+K). ::
               'FIN' ::
VAPLACER(SYNTHESE,COIND,PAS,J+16-J1) ::
       'FIN' ::
            LONGSYNT:= NOMB ::   SORTPERM1(SYNTHESE,N,0,LONGSYNT) ::
                             'FIN' ::
CELEZERO :
               'FIN' ::
NICELELA:
               L := L+1+S ::
    'SI' L'INFER'I       'ALORS' 'ALLERA' REBOUCLE ::
PACELELA  :
           'FIN' ::
NICELECI  :
        'FIN' ::
    'FIN' ::
PACELECI:
               I := 1+COMT+I ::
    'SI' I'INFEG'INDSUP 'ALORS' 'ALLERA' BOUCLE ::
'FIN' BIFURCATION ::



    SAUTLIGNE ::SAUTLIGNE ::
     PREPARE1(N) ::
     LIGNE(25,100) ::
     TRACE := P0-1 ::
     'POUR' I:= 1 'PAS' 1 'A'  N 'FAIRE' NONMASK.(I).:=ET(NG(MASQUE.(I).
     ),TRACE) ::
                                                     MASKPOS.(1).:=31 ::
     'POUR' I:=2 'PAS' 1 'A' 7  'FAIRE' MASKPOS.(I).:=TG(MASKPOS.(I-1).
     ,5 ) ::
MALET:=32767 :: MALONG:=MASKPOS.(4). :: NONMALONG:=NG(MALONG) ::
               MACOUT := TG(127,13) ::

  MANB:=MASKPOS.(5). :: NONMANB:=NG(MANB) :: MADR:=TG(MALET,25) ::
   NONMADR:= NG(MADR) ::
       SAUTLIGNE ::

 PAS:=NBSYNT := ICANON :=0 ::      CARSYNT:=P0 ::
        ALPHA := TEMPS ::
SORTEXTE('' ----------------------------------------- '') :: RCH ::
SORTEXTE('' BORNES FSUP ET FINF DE LA FONCTION A TRAITER '') :: RCH ::
```

```
6848  SORTEXTE('' ------------------------------------------------ '') :: RCH ::
6864   SAUTLIGNE :: SAUTLIGNE :: SAUTLIGNE ::
6870           LECT1(F,PDEF,N) :: EXTRF1(F,PDEF,N,75) :: RCH ::
6892          LECT1(H,P,N) ::
6901          EXTRF1(H,P,N,75) :: RCH ::       SAUTLIGNE ::
6916   RCH :: SORTEXTE('' COMPLEMENT DE FINF '') ::
6928          COMPLEMENT(H,P,N,G,PDEG) ::
6941        EXTRF1(G,PDEG,N,75) :: RCH ::
6954          SAUTLIGNE ::
6956      RCH ::  SORTEXTE('' DONT LES BASES COMPLETES SONT '') ::
6970            COMPREM1(F,PDEF,N,100,DC) ::
6983        'SI' DC 'ALORS' SORTEXTE('' DEPASS CAPA DE F '') ::
6996          EXTRF1(F,PDEF,N,75) :: RCH ::
7009          SAUTLIGNE :: SAUTLIGNE ::
7013          EXTRF1(G,PDEG,N,75) :: RCH ::
7026           RESUL.(0,0).:=RESUL.(0,1).:=TRACE ::
7042          PLACERNB2(PDEF-1,1) :: PLACERNB2(PDEG-1,0) ::
7060            PLACERADR(1,0) ::     PLACERADR(PDEG+1,1) ::
7076        RESUL.(1,0).:=RESUL.(1,1).:=0 ::
7092         RESUL.(-1,1). :=
7100            RESUL.(-1,0). := TG (TRACE,17) ::
7115        IND  := PDEF+PDEG ::
7121        'POUR' I:=1 'PAS' 1 'A' PDEG 'FAIRE' MON.(I).:=G.(I). ::
7140        'POUR' I:=1 'PAS' 1 'A' PDEF 'FAIRE'
7149      'DEBUT' J:=I+PDEG :: MON.(J).:=F.(I). 'FIN' ::
7167                     COMPT :=1 ::
7171   RCH :: SORTEXTE(''............'') ::SORENTIER( TEMPS-ALPHA) ::
7189   SORTEXTE('' DIXIEMES DE SECONDE'') ::
7199          RCH :: SAUTLIGNE :: SAUTLIGNE ::
7205       PRESENCE(F,PDEF,N,COMPT,IND,G,PDEG,ICANON) ::
7224          T:=1 ::
7228        SAUTLIGNE ::
7230   RCH :: SORTEXTE(''............'') ::SORENTIER( TEMPS-ALPHA) ::
7248   SORTEXTE('' DIXIEMES DE SECONDE'') ::
7258        'POUR' PAS:=2 'PAS' 1 'A' N 'FAIRE'
7267       'DEBUT' S:=T+1 ::  J:=T:=COMPT ::
7280          RCH :: SAUTLIGNE :: SAUTLIGNE ::
7286        BIFURCATION(PAS,S,T,N,COMPT,IND) ::
7301    SAUTLIGNE :: SAUTLIGNE ::
7305       'SI' J = COMPT 'ALORS' 'ALLERA' PASORBIF :: RCH ::
7315   SORBIF  : J:=J+1 :: K:=LIRENB1(J) :: SORTPERM1(RESUL,PAS,J,K) ::
7341       J:=J+K :: 'SI' J 'INFER' COMPT
7350                                 'ALORS' 'ALLERA' SORBIF :: RCH ::
7357  PASORBIF : SAUTLIGNE ::
7361   RCH :: SORTEXTE(''............'') ::SORENTIER( TEMPS-ALPHA) ::
7379   SORTEXTE('' DIXIEMES DE SECONDE'') ::
7389        'FIN' ::
7391  ILYAPLURIEN  :
7393        SAUTLIGNE :: SAUTLIGNE ::
7397        SAUTLIGNE :: SAUTLIGNE ::    RCH ::
7403   SORTEXTE('' -+-+-+-+-+-+-+-+-+-+-+-+-+-'') :: RCH ::
7417       SORTEXTE( '' SOLUTION OPTIMALE  DE COUT  '') ::
7428                                SORENTIER( CARSYNT) ::    RCH ::
7435   SORTEXTE('' -+-+-+-+-+-+-+-+-+-+-+-+-+-'') :: RCH ::
7449        SAUTLIGNE :: SAUTLIGNE ::
```

```
      SORTPERM1(SYNTHESE,N,D,LONGSYNT) ::
     SAUTLIGNE :: SAUTLIGNE ::
 RCH :: SORTEXTE(''.............'') ::SORENTIER( TEMPS-ALPHA) ::
 SORTEXTE('' DIXIEMES DE SECONDE'') ::
        RCH :: SAUTLIGNE :: SAUTLIGNE ::
SORTEXTE(''  ON A TRAVAILLE AVEC LE COUT EN LETTRES '') ::
 SAUTPAGE ::
   'FIN' ::
'FIN' PROGRAMME ::
FALGOL
```

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