



HAL
open science

Etude des décompositions d'un réseau

Michel Chein

► **To cite this version:**

Michel Chein. Etude des décompositions d'un réseau. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1967. Français. NNT: . tel-00280704

HAL Id: tel-00280704

<https://theses.hal.science/tel-00280704v1>

Submitted on 19 May 2008

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

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

THESES

présentées à

LA FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE

pour obtenir

LE GRADE DE DOCTEUR INGENIEUR

par

Michel CHEIN

Ingénieur I. M. A. G.



Première thèse :

**Etude des décompositions d'un réseau.
Application à l'écriture des Fonctions
Booléennes en sommes et produits**

Deuxième thèse :

PROPOSITIONS DONNEES PAR LA FACULTE



Thèses soutenues le 11 octobre 1967, devant la Commission d'examen :

Monsieur J. KUNTZMANN Président

Messieurs B. VAUQUOIS Examineurs
N. GASTINEL

FACULTE DES SCIENCES

LISTE DES PROFESSEURS

DOYENS HONORAIRES :

M. MORET

M. WEIL

DOYEN :

M. BONNIER E.

PROFESSEURS TITULAIRES :

MM. NEEL <i>Louis</i>	Chaire de Physique Expérimentale
HEILMANN <i>René</i>	Chaire de Chimie
KRAVTCHENKO <i>Julien</i>	Chaire de Mécanique Rationnelle
CHABAUTY <i>Claude</i>	Chaire de calcul différentiel et intégral
BENOIT <i>Jean</i>	Chaire de Radioélectricité
CHENE <i>Marcel</i>	Chaire de Chimie Papetière
WEIL <i>Louis</i>	Chaire de Thermodynamique
FELICI <i>Noël</i>	Chaire d'Electrostatique
KUNTZMANN <i>Jean</i>	Chaire de Mathématiques Appliquées
BARBIER <i>Reynold</i>	Chaire de Géologie Appliquée
SANTON <i>Lucien</i>	Chaire de Mécanique des Fluides
OZENDA <i>Paul</i>	Chaire de Botanique
FALLOT <i>Maurice</i>	Chaire de Physique Industrielle
KOSZUL <i>Jean-Louis</i>	Chaire de Mathématiques M.P.C.
GALVANI <i>O.</i>	Mathématiques
MOUSSA <i>André</i>	Chaire de Chimie Nucléaire
TRAYNARD <i>Philippe</i>	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
Melle LUTZ Elisabeth	Mathématiques
MM. 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
DESSAUX Georges	Chaire de Physiologie Animale
PILLET E.	Chaire de Physique Industrielle et Electrotechnique
VOCCOZ Jean	Chaire de Physique Nucléaire Théorique
DEBELMAS Jacques	Chaire de Géologie Générale
GERBER R.	Mathématiques
PAUTHENET R.	Electrotechnique
VAUQUOIS B.	Chaire de Calcul Electronique
BARJON R.	Physique Nucléaire
BARBIER Jean-Claude	Chaire de Physique
SILBER R.	Mécanique des Fluides
BUYLE-BODIN Maurice	Chaire d'Electronique
DREYFUS B.	Thermodynamique
KLEIN J.	Mathématiques
VAILLANT F.	Zoologie et Hydrobiologie
ARNOUD Paul	Chaire de Chimie M.P.C.
SENGEL P.	Chaire de Zoologie
BARNOUD F.	Chaire de Biosynthèse de la Cellulose
BRISSONNEAU P.	Physique
GAGNAIRE Didier	Chaire de Chimie Physique

Mme KÖFLER L.	Botanique
MM. DEGRANGE Charles	Zoologie
PEBAY-PEROULA J.C.	Physique
RASSAT A.	Chaire de Chimie Systématique

PROFESSEURS SANS CHAIRE :

MM. GIDON P.	Géologie et Minéralogie
GIRAUD P.	Géologie
PERRET R.	Servomécanismes
Mme BARBIER M.J.	Electrochimie
Mme SOUTIF J.	Physique
MM. COHEN J.	Electrotechnique
DEPASSEL R.	Mécanique des Fluides
GASTINEL N.	Mathématiques Appliquées
ANGLES-d'AURIAC P.	Mécanique des Fluides
DUCCROS P.	Minéralogie et Cristallographie
GLENAT R.	Chimie
LACAZE A.	Thermodynamique
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

MAITRES DE CONFERENCES :

MM. DODU J.	Mécanique des Fluides
LANCIA Roland	Physique Automatique
Mme KAHANE J.	Physique
MM. DEPORTES C.	Chimie
Mme BOUCHE L.	Mathématiques
MM. SARROT-RAYNAUD J.	Géologie Propédeutique
Mme BONNIER M.J.	Chimie
MM. KAHANE A.	Physique Générale
DOLIQUE J.M.	Electronique
BRIERE G.	Physique M.P.C.
DESPRE P.	Chimie S.P.C.N.
LAJZEROWICZ 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.
MM. LONGEQUEUE J.P.	Physique
NIVAT M.	Mathématiques Appliquées
SOHM J.C.	Electrochimie
ZADWORNY F.	Electronique
DURAND F.	Chimie Physique
CARLIER 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
BLOCH D.	Electrotechnique
BRUGEL L.	I.U.T.
SIBILLE R.	I.U.T.

Je tiens à exprimer ici ma profonde reconnaissance à Monsieur le Professeur J. KUNTZMANN, Directeur du Laboratoire de Calcul de l'Université de Grenoble, qui a dirigé ce travail et qui par son aide et ses conseils précieux 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 Messieurs les Professeurs N. GASTINEL et B. VAUQUOIS d'avoir bien voulu accepter de faire partie du Jury.

Je tiens aussi à remercier tous ceux, en particulier l'Equipe de logique du Laboratoire de Calcul, qui m'ont aidé dans ce travail, ainsi que le Secrétariat et le Service Tirage qui ont assuré la réalisation matérielle de cette thèse.

TABLE DES MATIERES

HAPITRE - I : ETUDE DES FRACTURES ET DES HYPERCOUPURES D'UN RESEAU

- DEFINITIONS

- 1.1. Réseau d'articulations
- 1.2. Réseau de noeuds et d'étoiles
- 1.3. Décomposition en k sous-réseaux connexes.

- HYPERCOUPURE D'UN ENSEMBLE FINI MUNI D'UNE RELATION BINAIRE

- 2.1. Définition
- 2.2. Groupe des hypercoupures : G_H
- 2.3. Systèmes générateurs de G_H

- DECOMPOSITION EN 2 SOUS RESEAUX CONNEXES

- 3.1. Cas d'un réseau d'articulations.
 - 3.1.1. Définitions
 - 3.1.2. Algorithme
- 3.2. Cas d'un réseau de noeuds et d'étoiles
 - 3.2.1. Définitions
 - 3.2.2. Représentation d'un réseau
 - 3.2.3. Algorithmes
 - 3.2.4. Programmes.
- 3.3. Recherche des composantes connexes d'un réseau.

- RECHERCHE DES FRACTURES MINIMALES D'UN RESEAU

- 4.1. Définitions. Algorithme de Malgrange.
- 4.2. Simplification de l'algorithme de Malgrange
 - 4.2.1. Treillis T
 - 4.2.2. Cas particulier T_1
 - 4.2.3. Simplification.

HAPITRE II: DECOMPOSITION D'UN RESEAU D'ARTICULATIONS EN k SOUS-RESEAUX CONNEXES

- RESEAU REGULIEREMENT DECOMPOSABLE.

- 1.1. Définitions
- 1.2. Une CNS pour qu'un réseau connexe soit régulièrement décomposable.
- 1.3. Propriété des charnières d'un réseau régulièrement décomposable.
- 1.4. Algorithme de reconnaissance d'un réseau régulièrement décomposable
- 1.5. Réseau sans charnière
- 1.6. Quelques propriétés.
- 1.7. Cas d'un réseau de noeuds et d'étoiles.

- DECOMPOSITION EN k SOUS-RESEAUX CONNEXES.

- 2.1. Borne supérieure du nombre de décompositions
- 2.2. Propriété générale.
- 2.3. Réseau condensé régulièrement décomposable
- 2.4. Décomposition en sous-réseaux connexes de même rayon.
 - 2.4.1. Position du problème
 - 2.4.2. Algorithme.

CHAPITRE III : RESULTATS GENERAUX SUR LES ECRITURES DES FONCTIONS BOOLEENNES.

1 - PRINCIPE DES METHODES.

- 1.1. Définitions. Position du problème.
- 1.2. Représentation d'une somme de monômes par un réseau de noeuds et d'étoiles.
- 1.3. Principe des méthodes.
 - 1.3.1. Méthode par fractionnement
 - 1.3.2. Fonction complète
 - 1.3.3. Fonction incomplète

2 - RESULTATS SUR LES ECRITURES MINIMALES

- 2.1. Décomposition d'une fonction booléenne
 - 2.1.1. Connexité. C-connexité
 - 2.1.2. Une condition nécessaire de décomposabilité par rapport au produit.
 - 2.1.3. Algorithme pour obtenir n_1, n_2 connaissant $n = n_1 + n_2$
- 2.2. Décomposabilité d'une écriture minimale.
- 2.3. Une borne inférieure du coût minimum.
- 2.4. Fermeture indépendante.
- 2.5. Ecriture minimale d'une fx incomplète dont l'une des bornes a une base première irredondante ayant au plus 2 monômes.
 - 2.5.1. Cas de 1 monôme.
 - 2.5.2. Fonction complète de 2 monômes.
 - 2.5.3. Cas de 2 monômes.

CHAPITRE IV : ALGORITHMES D'ECRITURE DES FONCTIONS BOOLEENNES EN S ET P.

1 - FONCTION COMPLETE CROISSANTE.

- 1.1. Algorithmes
 - 1.1.1. Respect des composantes connexes
 - 1.1.2. Critère de choix. Convergence
 - 1.1.3. Critères de décomposition
 - 1.1.4. Introduction de monômes redondants.
- 1.2. Programmes
- 1.3. Résultats.

2 - FONCTION COMPLETE QUELCONQUE.

- 2.1. Algorithmes
 - 2.1.1. Choix de la base de départ
 - 2.1.2. Respect des composantes connexes
 - 2.1.3. Critères de décomposition
 - 2.1.4. Introduction de monômes redondants
 - 2.1.5. Convergence des algorithmes.
- 2.2. Programmes
- 2.3. Résultats

3 - FONCTION INCOMPLETE.

- 3.1. Algorithmes
- 3.2. Programmes
- 3.3. Résultats.

4 - REMARQUES AU SUJET DE LA SYNTHÈSE EN OPÉRATEURS NI.

4.1. Remarques au sujet des algorithmes

4.2. Passage d'un réseau S-P à un réseau NI.

BIBLIOGRAPHIE.

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

CHAPITRES 1 et 2.

- [1] BERGE, C., Théorie des graphes et ses applications.
Dunod 1958
- [2] BIRKHOFF, G., Lattice Theory,
AMS Colloq. Publi. Vol XXV, 1948
- [3] HARARY, F., NORMAN, R.Z., CARTWRIGHT, D., Structural Models
John Wiley & Sons 1965
- [4] KUNTZMANN, J., Théorie des relations et des réseaux
Cours Grenoble 1967
- [5] BULL, (Publication) Utilisation des calculateurs électroniques en recherche
opérationnelle.

CHAPITRE 3 et 4.

- [1] ACKERS, S.B., A diagrammatic approach to multilevel logic synthesis
I.E.E.E., Trans. on EC, 14, 174-181, février 1965
- [2] ACKERS, S.B., ROBBINS T.C. Synthesis of combinational logic using 3-input
majority gates
Computing Review, 5,5, 1964
- [3] BARTEE, T.C., LEBOW, I.L., REED, I.S., Theory and Design of digital machines
Mc. Graw Hill Book Comp., 1962
- [4] CURTIS, H.A., A new approach to the design of switching circuits
D. Van Nostrand Co, Princeton (N.J.) 1962
- [5] DESCHIZEAUX, P., Synthèse de fonctions booléennes générales
Thèse de Docteur-Ingénieur, Grenoble 1967
- [6] KUNTZMANN, J., Algèbre de Boole
Dunod 1965
- [7] KUNTZMANN, J., Méthodes générales de réalisations de fonctions booléennes
par des organes technologiques donnés
Automatisme, X, 58-60, février 1965
- [8] LAPSCHER, F., Thèse à paraître - Grenoble.
- [9] LAWLER, E.L., An approach to boolean minimization
Journal A.C.M., 11, 283-295, 1964

- [10] LUSTMAN, F., Réalisation de fonctions booléennes avec l'opérateur majorité.
Thèse de Docteur-Ingénieur - Grenoble, 1966
- [11] MACHERAS, J., Synthèse de fonctions booléennes par l'opérateur U
Thèse de 3ème Cycle, Grenoble 1966
- [12] MEO, A.R., On the minimal third order expression of a boolean function
A.I.E.E. Symp. on Switching Theory, Chicago, 1962 pp. 5-25
- [13] PICHAT, E., Décompositions des fonctions booléennes
Thèse de 3ème Cycle, Grenoble 1966
- [14] PICHAT, E., Décompositions disjointes simples des fonctions booléennes
Séminaire de Logique, Grenoble 1967 (non publié)
- [15] SESHU, S., REED, M.B., Linear graphs and electrical networks
Addison Wesley Public. Comp., 1961
- [16] TISON, P., Théorie des consensus
Thèse de Docteur-Ingénieur - Grenoble 1965

CHAPITRE I

ETUDE DES FRACTURES ET DES HYPERCOUPURES D'UN RESEAU

I- DEFINITIONS

I.I. Réseau d'articulations.

L'articulation est l'élément constitutif des réseaux, il peut communiquer avec l'extérieur en un nombre fini de points appelés connexions.

Un réseau est constitué par deux ensembles :

- un ensemble d'articulations $M = \{ a_1, a_2, \dots, a_m \}$
- un ensemble \mathcal{C} de couples de connexions tel que :

toute connexion apparaît au plus une fois dans \mathcal{C} .

Deux articulations, a_1 et a_2 , seront dites jointives s'il existe au moins un couple de connexions $(c_1, c_2) \in \mathcal{C}$ c_1 étant une connexion de a_1 et c_2 une de a_2 , la relation sur $M \times M$ ainsi définie s'appelle relation de jointivité $\mathcal{J}(a_1, a_2)$ du réseau.

Les classes d'équivalence définies dans M par la fermeture transitive forte de \mathcal{J} sont appelées composantes connexes du réseau, un réseau est connexe s'il a une seule composante connexe.

Si toute connexion est élément d'un $c \in \mathcal{C}$ le réseau est dit fermé on peut alors le considérer comme un graphe topologique.

Une chaîne est un ensemble d'articulations (a_1, a_2, \dots, a_n) : a_i étant jointive à a_{i+1} quel que soit $i = 1, 2, \dots, n - 1$, si tous les a_i sont distincts elle est dite élémentaire, si $a_1 = a_n$ c'est un cycle qui est dit élémentaire si tous les a_i sont distincts.

1.2. Réseau de noeuds et d'étoiles.

C'est un réseau d'articulations pour lequel M est partagé en deux classes disjointes : les étoiles et les noeuds telles que si deux articulations sont jointives l'une est un noeud, l'autre une étoile.

1.3. Décomposition en k sous-réseaux connexes.

Soit R un réseau d'articulations connexe ayant M pour ensemble d'articulations et \mathcal{J} pour relation de jointivité, le sous-réseau engendré par $M' \subset M$ a M' pour ensemble d'articulations et \mathcal{J}' pour relation de jointivité avec :

$$\forall x', y' \in M' \quad \mathcal{J}'(x', y') = \mathcal{J}(x, y). \quad (\text{restriction de } \mathcal{J} \text{ à } M').$$

Si $(R_i)_{i \in I}$ sont des sous-réseaux engendrés par une partition $(M_i)_{i \in I}$ de M en posant :

$$\bigcup_{i \in I' \subset I} R_i = R' \left(\bigcup_{i \in I'} M_i, \bigcup_{i \in I'} \mathcal{J}_i \right)$$

R est égal à l'union $\bigcup_{i \in I} R_i$ nous disons que R est décomposé en I sous-réseaux.

Nous nous proposons d'étudier les décompositions d'un réseau connexe en k sous-réseaux connexes c'est-à-dire d'étudier les ensembles de k sous-réseaux connexes ayant R pour union.

On appelle réseau condensé associé à la décomposition le réseau ayant R_i pour articulations R_i étant lié à R_j si au moins une articulation de R_i étant liée à R_j .

2 - HYPERCOUPURE D'UN ENSEMBLE FINI MUNI D'UNE RELATION BINAIRE

2.1. Définition.

Soit un ensemble E fini, de m éléments, muni d'une relation interne $R(x, y)$ que nous supposerons symétrique (si elle n'est pas symétrique nous la remplacerons par sa symétrisée $S(x, y) = R(x, y) + R(y, x)$).

Une partie $A \subset E$ est dite connexe par rapport à la relation R si la fermeture transitive forte de la restriction de R à A ne définit qu'une seule classe d'équivalence. Nous supposerons par la suite que E est connexe (par rapport à R).

On appelle hypercoupure de E toute bipartition de $E : (A ; \bar{A})$. Une hypercoupure est dite élémentaire si son 1^{er} membre est réduit à un seul élément, simple si les parties A et \bar{A} de E sont connexes par rapport à R .

2.2. Groupe des hypercoupures : G_H .

L'ensemble des hypercoupures d'un ensemble E , muni de la loi \oplus (disjonction) définie par :

$$(A ; \bar{A}) \oplus (B ; \bar{B}) = ((A \cap \bar{B}) \cup (\bar{A} \cap B) ; (A \cap B) \cup (\bar{A} \cap \bar{B}))$$

a une structure de groupe abélien. En effet :

- c'est une loi interne car le premier membre du résultat est égal à la différence symétrique des premiers membres et le second à son complément.

- elle est associative et commutative du fait de l'associativité et de la commutativité de la différence et de la somme symétriques.

- $(\emptyset ; E)$ est élément neutre :

$$(A ; \bar{A}) \oplus (\emptyset ; E) = \{(A \cap E) \cup (\bar{A} \cap \emptyset) ; (A \cap \emptyset) \cup (\bar{A} \cap E)\} = (A ; \bar{A})$$

- tout élément est son propre inverse :

$$(A ; \bar{A}) \oplus (A ; \bar{A}) = \{(A \cap \bar{A}) \cup (\bar{A} \cap A) ; (A \cap A) \cup (\bar{A} \cap E)\} = (\emptyset ; E).$$

Pour des raisons de commodité d'écriture nous représenterons souvent une hypercoupure seulement par son premier membre par exemple :

$$A \oplus B = (A \cap \bar{B}) \cup (\bar{A} \cap B).$$

Quelques propriétés de \oplus :

$$A, B \in G_H$$

$$\overline{(A \oplus B)} = \bar{A} \oplus B = A \oplus \bar{B}$$

$$A \oplus B = \bar{A} \oplus \bar{B}$$

$$A \oplus E = \bar{A}$$

$$A \oplus \bar{A} = E$$

si $A \cap B = \emptyset$ alors $A \oplus B = A \cup B$

Les hypercoupures A et \bar{A} seront dites complémentaires.

2.3. Bases du groupe des hypercoupures.

Le groupe G_H étant abélien de type fini nous savons que tous les systèmes d'éléments linéairement indépendants maximaux sont des systèmes générateurs irréductibles ayant le même nombre d'éléments r rang de G_H .

Théorème 1 :

L'ensemble des hypercoupures élémentaires d'un ensemble E fini est une base du groupe des hypercoupures.

En effet soit $(A ; \bar{A}) = (a_1 a_2 \dots a_p ; b_1 b_2 \dots b_q)$ une hypercoupure quelconque de E :

$(A ; \bar{A}) = a_1 \oplus a_2 \oplus \dots \oplus a_p$ donc l'ensemble des hypercoupures élémentaires est un système générateur de G_H qui est irréductible car si on enlevait une hypercoupure élémentaire on ne pourrait pas obtenir l'hypercoupure $(E ; \emptyset)$.

Théorème 2 :

Les hypercoupures simples forment un système générateur de G_H .

Démonstration :

Soit $(A ; B)$ une hypercoupure quelconque non simple. On note $A_1 A_2 \dots A_p$ et $B_1 B_2 \dots B_q$ les parties de E qui sont les composantes connexes par rapport à R de A et de B . On a :

$$\bigcup_{i=1}^p A_i = A \qquad \bigcup_{j=1}^q B_j = B \qquad A \cup B = E \qquad A \cap B = \emptyset$$

$$A_i \cap A_j = \emptyset \quad i \neq j \qquad B_i \cap B_j = \emptyset$$

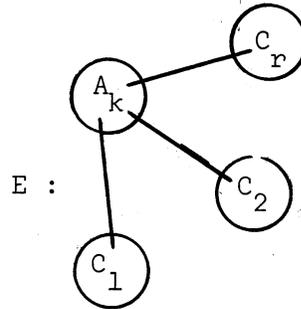
et pour $q \geq 2$ sinon $(A ; B)$ serait simple.

Si les hypercoupures $(A_i ; \bar{A}_i)$ sont simples le théorème est alors démontré car : $(A ; B) = A_1 \oplus A_2 \oplus \dots \oplus A_p$.

Supposons qu'il existe au moins un k tel que $(A_k ; \bar{A}_k)$ ne soit pas simple, \bar{A}_k n'est pas connexe par rapport à R notons C_1, C_2, \dots, C_r ses composantes connexes.

Il existe un élément de A_k en relation avec un élément de C_i sinon un élément de C_i serait en relation avec un de C_j , puisque E est connexe, mais alors C_i et C_j sont dans la même composante connexe de \bar{A}_k . L'hypercoupure $(C_1 ; \bar{C}_1)$ est donc simple puisque $\bar{C}_1 = C_2 \cup \dots \cup C_r \cup A_k$ est connexe un élément de A_k étant relié à un élément de $C_i \quad \forall i = 1, 2, \dots, r$. On voit que :

$C_1 \oplus C_2 \oplus \dots \oplus C_r = (C_1 \ C_2 \ \dots \ C_r ; A_k)$ nous pouvons donc obtenir toutes les hypercoupures $(A_i ; \bar{A}_i)$ non simples par disjonction d'hypercoupures simples il ne reste plus qu'à faire la disjonction des A_i pour obtenir $(A ; B)$.



Remarque :

Si nous considérons $(E ; \emptyset)$ comme une hypercoupure simple il est inutile de considérer deux hypercoupures simples complémentaires puisque : $(A ; \bar{A}) \oplus (E ; \emptyset) = (\bar{A} ; A)$ donc si un ensemble E (muni d'une relation R) à h hypercoupures simples tout ensemble de $\frac{h}{2}$ hypercoupures simples tel que 2 d'entre elles ne soient pas complémentaires est un système générateur de G_H : de tels ensembles sont appelés des ensembles suffisants d'hypercoupures simples il y en a $2 \frac{h}{2}$.

3 - DECOMPOSITION EN 2 SOUS-RESEAUX CONNEXES

3.1. Cas d'un réseau d'articulations.

3.1.1. Définitions.

Une coupure d'un réseau d'articulations connexe $R(M, \mathcal{C})$ est un ensemble $\mathcal{C}' \subset \mathcal{C}$ de couples de connexions tel que :

$R(M, \mathcal{C} - \mathcal{C}')$ soit non connexe, \mathcal{C}' étant minimal par inclusion en ce qui concerne cette propriété.

Une hypercoupure d'un réseau d'articulations connexe sera une hypercoupure de l'ensemble des articulations M muni de la relation de jointivité \mathcal{C} .

Une fracture d'un réseau connexe est un ensemble d'articulations $M' \subset M$ tel que le sous-réseau engendré par $M - M'$ soit non connexe, minimal par inclusion en ce qui concerne cette propriété.

Une charnière est une fracture constituée d'une seule articulation.

Propriétés :

1 - Si toutes les hypercoupures élémentaires d'un réseau sont simples le réseau est sans charnière et réciproquement.

2 - Tout réseau connexe d'au moins 2 articulations a au moins deux hypercoupures élémentaires simples puisqu'il a au moins deux articulations qui ne sont pas charnières ([1] p. 192).

3 - Si un réseau est réduit à une chaîne : tous les ensembles suffisants d'hypercoupures simples sont des bases puisqu'ils ont $m = |M|$ éléments (il y a 2^m hypercoupures simples).

4 - Une coupure définit 2 hypercoupures simples complémentaires et réciproquement 2 hypercoupures simples complémentaires définissent la même coupure, en effet une coupure fait apparaître dans le réseau exactement deux composantes connexes (sinon elle ne serait pas minimale) ayant donc des ensembles d'articulations A et \bar{A} complémentaires ($A ; \bar{A}$) et ($\bar{A} ; A$) sont bien des hypercoupures simples. La réciproque est évidente.

3.1.2. Algorithme.

La propriété 4 de 3.1.1. permet de ramener la recherche des décompositions d'un réseau connexe en 2 sous-réseaux connexes, c'est-à-dire de ses hypercoupures simples, à celle de ses coupures. En utilisant la propriété suivante : (voir [4]) une coupure est un ensemble minimal de couples de connexions contenant un couple de connexions de chaque arbre complet du réseau, on obtient un algorithme simple pour résoudre ce problème.

Un arbre complet d'un réseau d'articulations connexe R est un arbre (réseau connexe n'ayant aucun cycle) ayant pour articulations celles de R et obtenu à partir de R en supprimant certains couples de connexions.

La propriété ci-dessus est alors évidente car un tel ensemble de couples de connexions fait perdre sa connexité au réseau si on le supprime et qu'aucun de ses sous-ensembles ne possède cette propriété.

On représente chaque arbre complet du réseau par une somme de lettres chacune représentant un couple de connexions, on fait le produit des sommes ainsi obtenues pour tous les arbres complets, on développe ce produit en considérant les lettres comme des variables booléennes et en respectant les règles de calcul de l'algèbre de Boole, chaque monôme obtenu définit une coupure, ([4]).

De nombreux algorithmes existent pour résoudre ce problème notamment à partir de la connaissance des cycles du réseau, mais nous sommes plus particulièrement intéressé par l'étude de ce problème dans le cas d'un réseau de noeuds et d'étoiles et il est intéressant de construire un algorithme spécifique.

3.2. Cas d'un réseau de noeuds et d'étoiles.

3.2.1. Définitions.

Considérons un réseau connexe $R(X, M)$ de noeuds X , et d'étoiles M .

Les notions ne faisant intervenir que les couples de connexions dans le cas d'un réseau d'articulations sont inchangées pour un réseau de noeuds et d'étoiles. Par exemple : une coupure d'un réseau de noeuds et d'étoiles est un ensemble minimum de couples de connexions dont la suppression disconnecte le réseau.

Les notions faisant intervenir l'ensemble des articulations peuvent être dédoublées du fait de la bipartition de cet ensemble en noeuds et étoiles. Nous pouvons ainsi définir une fracture-noeud (resp. étoile) comme une fracture du réseau de noeuds et d'étoiles considéré comme un réseau d'articulations et n'ayant que des noeuds (resp. étoile), nous ne considérerons que les fractures-noeud que, pour des raisons de commodité, nous appellerons fracture de réseau de noeuds et d'étoiles. De même une hypercoupure d'un réseau de noeuds et d'étoiles est une hypercoupure de l'ensemble des étoiles muni de la relation : deux étoiles sont en relation s'il existe au moins un noeud adjacent aux deux.

A une hypercoupure simple $(A ; \bar{A})$ de $R(X, M)$ est associée une fracture unique constituée des noeuds adjacents à au moins une étoile de A et une de \bar{A} . La réciproque est fautive car la suppression des noeuds d'une fracture peut décomposer le réseau en plus de 2 sous-réseaux connexes, on peut alors associer un ensemble d'hypercoupures simples à cette fracture. En effet si $Y = \{a, b, \dots, c\}$ est une fracture de $R(X, M)$, notons R_1, R_2, \dots, R_p les sous-réseaux connexes de $R(X, M)$ engendrés par les p composantes connexes de $R(X - Y, M)$. Toute décomposition de $R(X, M)$ en 2 sous-réseaux connexes correspondant à une bipartition des R_i peut être obtenue en dédoublant tous les noeuds de Y . Si, par exemple, on veut décomposer R en

$$\left. \begin{aligned} R'_1 &= R_1 \cup R_2 \cup \dots \cup R_\ell \\ R'_2 &= R_{\ell+1} \cup \dots \cup R_p \end{aligned} \right\} \text{ en supposant } R'_1 \text{ et } R'_2 \text{ connexes.}$$

il suffit de remplacer a, b, \dots, c par a_1, b_1, \dots, c_1 dans R'_1 et par a_2, b_2, \dots, c_2 dans R'_2 .

Remarques :

1 - Les seules décompositions d'un réseau de noeuds et d'étoiles que nous étudierons seront celles obtenues en fractionnant les noeuds d'une fracture.

2 - Tout réseau connexe $R(X, M)$ de noeuds et d'étoiles avec $|M| \geq 2$ a au moins une fracture : celle constituée par tous les noeuds adjacents à une étoile donnée et qui ne sont pas pendants. Un noeud est pendant s'il n'est adjacent qu'à une seule étoile (ceci est faux dans le cas d'un réseau d'articulations : un réseau d'articulations complet, c'est-à-dire tel que 2 articulations quelconques soient adjacentes, n'a pas de fracture).

3.2.2. Représentation d'un réseau de noeuds et d'étoiles.

Un réseau de noeuds et d'étoiles sera donné par :

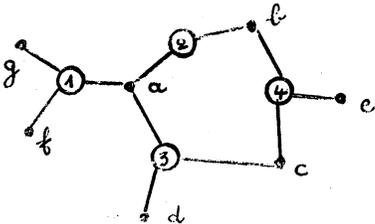
$X = \{a, b, \dots, \ell\}$ ensemble des noeuds

$M = \{1, 2, \dots, m\}$ ensemble des étoiles

$F = \{X_1, X_2, \dots, X_m\}$ X_i étant l'ensemble des noeuds adjacents à l'étoile i .

Il sera représenté en machine (IBM 7044) par un ensemble de m mémoires numérotées $1, 2, \dots, m$ un bit présent en $j^{\text{ième}}$ position de la mémoire i indiquant que le $j^{\text{ième}}$ noeud est adjacent à la $i^{\text{ième}}$ étoile (si $m > 36$ on prendra plusieurs mémoires en parallèle).

Exemple :



$X = \{a, b, c, d, e, f, g\}$

$M = \{1, 2, 3, 4\}$

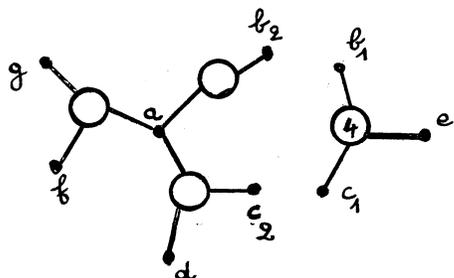
$F = \{afg, ab, acd, bce\}$

F en machine :	1 :															

les positions 8 à 36 étant nulles puisque $|X| = 7$.

3.2.3. Algorithmes pour rechercher les hypercoupures simples.

On supprime tous les noeuds pendants du réseau, F est alors le tableau des hypercoupures élémentaires h et des fractures associées minimales f_h . En effet il faut nécessairement fractionner tous les noeuds de f_h pour obtenir une décomposition du réseau ayant h pour sous-réseau connexe. Le fractionnement minimum est obtenu en dédoublant un noeud n de f_h en : n_1 pour h et n_2 pour \bar{h} . En reprenant l'exemple précédent nous voyons qu'il est nécessaire de dédoubler b et c pour obtenir une décomposition de R ayant un sous-réseau connexe réduit à l'étoile 4 :



1^{er} Algorithme :

On génère toutes les hypercoupures à partir des m hypercoupures simples et on ne conserve que celles qui sont simples. Cette méthode est très longue car elle nécessite le calcul de $2^{\lfloor \frac{m}{2} \rfloor}$ hypercoupures et on doit effectuer $2 \times 2^{\lfloor \frac{m}{2} \rfloor}$ tests de connexité : elle ne sera pas utilisée.

2^{ème} Algorithme :

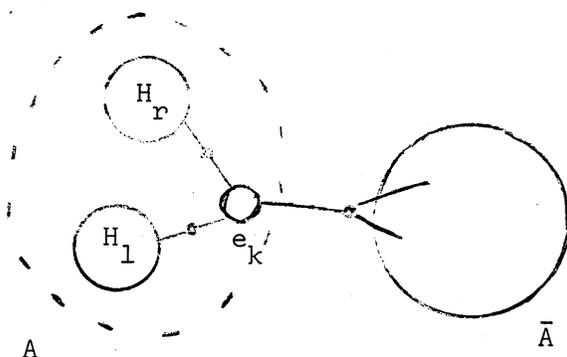
Il est basé sur la propriété suivante :

Propriété :

Toute hypercoupure simple est la réunion d'hypercoupures simples disjointes inférieures et d'une étoile à laquelle elles sont toutes jointives.

En effet soit $(A ; \bar{A})$ une hypercoupure simple de $R(X, M)$ il existe au moins une étoile $e_k \in A$ liée à \bar{A} , la suppression de e_k engendre une partition H_1, H_2, \dots, H_r de A ($r \geq 1$), les H_i sont des hypercoupures simples de $R(X, M)$ disjointes et inférieures à A , c'est-à-dire $H_i \subset A$, et l'on a :

$$A = \left\{ \bigcup_{i=1}^r H_i, e_k \right\}.$$



Si $X_i \cap X_j \neq \emptyset$ il existe une hypercoupure $(ij ; \dots)$ ayant pour fracture associée :

$$X_i \cup X_j - \{ \text{les noeuds adjacents uniquement à } i \text{ ou/et } j \}$$

on obtient ainsi toutes les hypercoupures séparant un sous-réseau connexe de 2 étoiles, on élimine celles qui ne sont pas simples.

Plus généralement pour obtenir les hypercoupures simples ayant k éléments dans leur premier membre on associe q hypercoupures simples disjointes (h_i) déjà obtenues ayant i_1, i_2, \dots, i_q éléments et une hypercoupure élémentaire h_e de telle sorte que le sous-réseau ayant pour étoiles $\left\{ \bigcup_{i=1}^q h_i, h_e \right\}$ soit connexe.

Cette méthode est assez difficile à mettre en oeuvre, nous avons programmé la méthode suivante.

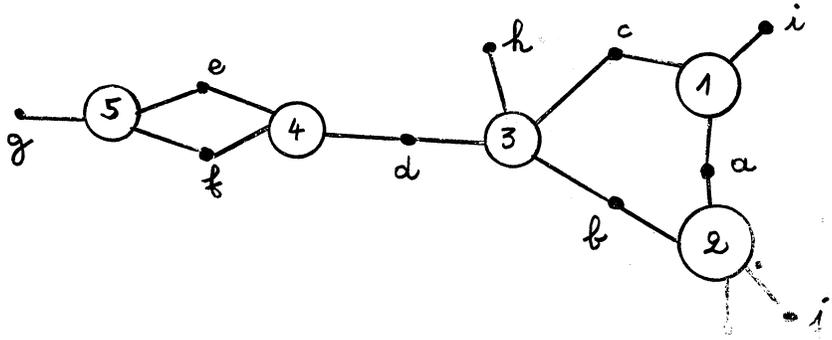
3^{ème} Algorithme :

On complète le tableau F en cherchant comme précédemment les hypercoupures séparant un sous-réseau connexe de 2 étoiles mais on n'élimine pas celles qui ne sont pas simples. A partir de ces hypercoupures ayant 2 étoiles dans leur premier membre et des hypercoupures élémentaires on forme celles ayant 3 étoiles dans leur premier membre et qui engendrent un sous-réseau connexe etc...

si $m = 2p + 1$ on s'arrête lorsque l'on a obtenu les hypercoupures ayant p étoiles dans leur premier membre

si $m = 2p$ on s'arrête lorsque l'on a obtenu les hypercoupures ayant p étoiles dans leur premier membre et contenant une même étoile. Lorsque la liste des hypercoupures et de leur fracture associée est complète il faut éliminer les hypercoupures non simples, c'est-à-dire faire un test de connexité sur \bar{A} si A est dans la liste.

Exemple :



On note M_i l'ensemble des étoiles adjacentes au noeud i , X_i l'ensemble des noeuds adjacents à l'étoile i .

$$F = \{aci, abkj, bcdh, def, efg\}$$

1 - On supprime les noeuds pendants ($|M_i| = 1$) et on forme le tableau des hypercoupures H_i ayant 1 étoile dans leur premier membre et de leur fracture associée F_i à partir de F :

H_i	F_i
1	ac
2	ab
3	bed
4	def
5	ef

T1

H_i	F_i
12	ab c
13	abcd
23	abcd
34	bcdef
45	d e f

T2

2 - Comme $m = 5$ on ne forme que les hypercoupures ayant 2 étoiles dans leur premier membre, on obtiendra les autres en complétant.

Si $F_i \cap F_j \neq \emptyset$ et si $F_{ij} = F_i \cup F_j - L \neq \emptyset$ avec $L = \{l \in F_i \cup F_j ; M_l \subset H_i \cup H_j\}$ alors on ajoute l'hypercoupure $H_i \cup H_j$ ayant pour fracture associée F_{ij} on obtient ainsi le tableau T2.

3 - Pour m quelconque à partir de T_{h-1} et de T_1 on forme T_h en faisant comme en 2. On s'arrête lorsque l'on a obtenu toutes les hypercoupures ayant $\frac{m}{2}$ étoiles et contenant la même étoile dans le cas où m est pair, lorsque l'on a obtenu toutes les hypercoupures ayant $E\left(\frac{m}{2}\right)$ si m est impair.

4 - Si A est une hypercoupure de l'un des T_i on la garde si le réseau engendré par \bar{A} est connexe sinon on la supprime.

En considérant les hypercoupures simples ainsi obtenues et leurs complémentaires on obtient toutes les hypercoupures simples du réseau et leur fracture associée.

Remarque :

Comme nous n'utilisons que T_{h-1} et T_1 pour obtenir T_h si nous cherchons uniquement les hypercoupures simples ayant certaines propriétés, par exemple une fracture associée ayant un nombre minimum de noeuds, après avoir formé T_h on peut éliminer toutes les hypercoupures de T_{h-1} ne satisfaisant pas cette propriété.

3.2.4. Programmes.

Nous avons programmé le 3^{ème} algorithme de 3.2.3. en utilisant la représentation 3.2.2. d'un réseau qui limite à 36 le nombre de noeuds de ce réseau (on peut facilement généraliser tous les programmes en considérant plusieurs mémoires en parallèle).

La procédure ARTI1 donne toutes les hypercoupures simples d'un réseau connexe avec leurs fractures associées, la procédure ARTI2 donne : soit la première hypercoupure simple ayant un nombre maximum d'étoiles parmi celles ayant une fracture associée ayant un nombre minimum de noeuds,

soit la première hypercoupure simple de $E\left(\frac{m}{2}\right)$ étoiles ayant une fracture associée ayant un nombre minimum de noeuds. Les programmes sont écrits en ALGOL et utilisent les procédures en MAP :

entier procedure OU(A,B) ; valeur A,B ; entier A,B ;

commentaire : union logique de A et B ;

entier procedure ET(A,B) ; valeur A,B ; entier A,B ;

commentaire : intersection logique de A et B ;

entier procedure DJ(A,B) ; valeur A,B ; entier A,B ;

commentaire : disjonction de A et B ;

entier procedure TD(A,I) ; valeur A,I ; entier A,I ;

commentaire : translation de I positions vers la droite de A ;

entier procedure TG(A,I) ; valeur A,I ; entier A,I ;

commentaire : translation de I positions vers la gauche de A ;

entier procedure BIT(A,I) ; valeur A,I ; entier A,I ;

commentaire : Iième digit de A ;

entier procedure POIDS(A) ; valeur A ; entier A ;

commentaire : nombre de digits de A ;

procedure NULTAB(A,K) ; valeur K ; entier K ; entier tableau A ;

commentaire : cette procédure annule les K premières mémoires du tableau A ;

Les programmes complets sont donnés en annexe. La procédure ARTI1 a 1.000 unités syntaxiques, ARTI2 en a 1.067.

procedure ARTI1(F,M,N,AD,VA,H,TROP) ; valeur F,M,N ; entier M,N,H ;

entier tableau F,AD,VA ; etiquette TROP ;

commentaire cette procédure range dans le tableau AD de H mémoires les premiers membres des hypercoupures simples et dans VA leur fracture associée du réseau connexe de M étoiles et de N noeuds définit par le tableau F. On va en TROP si $H > 1.000$;

procédure ARTI2(F,M,N,HS,AHS,TEST,TROP) ; valeur F,M,N ; entier M,N,HS,AHS ;
entier tableau F ; booléen TEST ; etiquette TROP ;
commentaire HS contient le premier membre de l'hypercoupure simple ayant un
nombre maximum d'étoiles parmi celles ayant une fracture associée ayant un nom-
bre minimum de noeuds si TEST est vrai, sinon HS contient le premier membre de
l'hypercoupure simple ayant M/2 étoiles et une fracture associée minimum. AHS
contient la fracture associée à HS. On va en TROP si un tableau T_i a plus de
1000 éléments. Le réseau est représenté comme en ARTI1 ;

La procédure DECP décompose un réseau connexe en 2 sous-réseaux connexes en
dédoublant les noeuds de la fracture associée à une hypercoupure simple donnée.
procédure DECP(F,M,N,HS,F1,M1,N1,F2,M2,N2) ; valeur F,M,N,HS ; entier
M,N,M1,N1,M2,N2,HS ; entier tableau F,F1,F2 ;
commentaire le réseau connexe (F,M,N) est décomposé en deux sous-réseaux
connexes (F1,M1,N1) et (F2,M2,N2), F1 ayant pour étoiles celles de HS hyper-
coupure simple du réseau ;

3.3. Recherche des composantes connexes.

$A = (a_{ij})$ étant la matrice associée à un réseau d'articulations :
c'est-à-dire $a_{ij} = 1$ si l'articulation i est adjacente à l'articulation
 j , $a_{ij} = 0$ autrement ($a_{ii} = 1$) on peut soit rechercher la fermeture transitive
de A soit, ce qui est beaucoup plus rapide, procéder par fusion de lignes de A .
Si deux lignes de A ont une intersection non vide, on remplace l'une de ces
deux lignes par leur union et on supprime l'autre etc... Quand le processus
s'arrête il reste des lignes correspondant chacune à une composante connexe,
les éléments de cette composante étant indiqués par les 1 présents.

Nous avons programmé cette dernière méthode dans le cas d'un réseau
de noeuds et d'étoiles avec la représentation usuelle. Le programme écrit en
ALGOL a 265 unités syntaxiques et utilise les procédures en MAP : OU, ET,
NULTAB déjà décrites.

procedure COMCNXE(F,M,IC,NC,SORT1) ; valeur M ; entier M,NC ;
entier tableau F,IC ; etiquette SORT1 ;
commentaire cette procédure range dans le tableau F entre IC(A) et IC(I+1) la Iⁱème composante connexe du réseau (F,M) ayant MC composantes connexes. On va en SORT1 si NC > 100 ou si l'une des composantes connexes a plus de 100 étoiles ;

Dans certain cas, seule la connexité ou la non-connexité d'un réseau donné nous intéresserons, nous avons écrit une procédure prenant la valeur vrai si le réseau est connexe faux autrement et basée sur la méthode précédente.

Ecrite en ALGOL cette procédure a 120 unités syntaxiques, elle utilise les procédures en MAP : ET, OU.

booleen procedure CONEX(F,M) ; valeur F,M ; entier M ; entier tableau F ;
commentaire CONEX est vrai si le réseau (F,M) est connexe, faux autrement ;

4 - RECHERCHE DES FRACTURES MINIMALES D'UN RESEAU

4.1. Définitions. Algorithme de Malgrange (voir [5]).

On considère la matrice de connexion (M, M) du réseau ($m_{ij} = 1$ si les articulations i et j sont adjacentes). Si un sous-tableau de cette matrice ayant A, B pour ensemble de lignes et de colonnes est rempli de 0 la suppression de $\overline{A \cup B}$ disconnecte le réseau donc $\overline{A \cup B}$ est une fracture de R ($A \cup B \neq M$ sinon R ne serait pas connexe) la réciproque est évidente.

Le problème de la recherche des fractures minimales sera simplifié par la connaissance des sous-tableaux maximaux (par rapport à l'inclusion) ou "premiers" n'ayant que des 0 pour éléments, il suffira de considérer parmi ces sous-tableaux ceux ayant un nombre maximum d'éléments.

L'algorithme de Malgrange résout ce problème : il fournit les sous-tableaux premiers remplis du même élément a d'un tableau rectangulaire ayant I, J pour ensembles d'indices de lignes et de colonnes. Soit P l'ensemble des cases du tableau ayant la valeur a et T_1 l'ensemble des sous-tableaux n'ayant que des a.

$(I_1, J_1), (I_2, J_2) \in T_1$ on définit deux opérations :

$$(I_1, J_1) \vee (I_2, J_2) = (I_1 \cup I_2, J_1 \cap J_2) \quad J_1 \cap J_2 \neq \emptyset$$

$$(I_1, J_1) \wedge (I_2, J_2) = (I_1 \cap I_2, J_1 \cup J_2) \quad I_1 \cap I_2 \neq \emptyset$$

On appelle couverture de P un ensemble :

$$\mathcal{C} = \{(X_\ell, Y_\ell) \in T_1\} \text{ tel que } \forall (x_i, y_j) \in P \text{ il existe } \ell \text{ tel que } x_i \in X_\ell \text{ et } y_j \in Y_\ell.$$

Tous les éléments premiers de T_1 s'obtiennent par \vee et \wedge à partir d'une couverture quelconque.

En effet soit $(A, B) \in T_1$ nous allons montrer qu'à partir d'une couverture on peut obtenir $(X_{AB}, Y_{AB}) \in T_1$ avec $X_{AB} \supset A$ et $Y_{AB} \supset B$.
Posons $A = \{x_1, x_2, \dots, x_r\}$ $B = \{y_1, y_2, \dots, y_s\}$ et soit \mathcal{C} une couverture de T_1 .

Il existe $c_{ij} \in \mathcal{C}$ qui couvre (x_i, y_j) $\forall i = 1, 2, \dots, r$
 $\forall j = 1, 2, \dots, s$

en posant $c_1 = c_{11} \wedge c_{12} \wedge \dots \wedge c_{1s}$

c_1 couvre $(x_1, y_1 y_2 \dots y_s)$ de même :

$c_r = c_{r1} \wedge c_{r2} \wedge \dots \wedge c_{rs}$ couvre $(x_r, y_1 y_2 \dots y_s)$ donc

$c_1 \vee c_2 \vee \dots \vee c_r$ couvre (A, B) .

4.2. Simplification de l'algorithme de Malgrange.

4.2.1. Treillis de l'ensemble des parties du produit de 2 ensembles finis.

Soient I et J deux ensembles finis. $\mathcal{P}(I)$ (resp. $\mathcal{P}(J)$), ensemble des parties de I , muni de l'union et de l'intersection forme un treillis de Boole \mathcal{B}_1 (resp. \mathcal{B}_2) avec l'ordre habituel :

$$x, y \in \mathcal{B}_1 \quad x \geq_U y \iff x \cup y = x$$

Le dual \mathcal{B}_2^* est également un treillis de Boole : on notera son ordre \geq_\cap : $x, y \in \mathcal{B}_2^* \quad x \geq_\cap y \iff x \cap y = x$.

Le produit cardinal $(T, \geq_1) = (\mathcal{B}_1, \geq_U) \times (\mathcal{B}_2^*, \geq_\cap)$ est formé des éléments de $\mathcal{P}(I) \times \mathcal{P}(J)$ ordonné par :

$$(x_1, y_1), (x_2, y_2) \in T$$

$$(x_1, y_1) \geq_1 (x_2, y_2) \iff x_1 \geq_U x_2 \text{ et } y_1 \geq_\cap y_2$$

$$\iff x_1 \cup x_2 = x_1 \text{ et } y_1 \cap y_2 = y_1$$

T est un treillis de Boole (cette propriété se conserve par produit) dont les opérations sont :

$$(x_1, y_1) \vee (x_2, y_2) = (x_1 \cup x_2, y_1 \cap y_2)$$

$$(x_1, y_1) \wedge (x_2, y_2) = (x_1 \cap x_2, y_1 \cup y_2)$$

$$\text{avec } (x_1, y_1) \geq_1 (x_2, y_2) \iff (x_1, y_1) \vee (x_2, y_2) = (x_1, y_1)$$

l'élément nul est (\emptyset, J) en effet : $\forall (x, y) \in T :$

$$(\emptyset, J) \wedge (x, y) = (\emptyset \cap x, J \cup y) = (\emptyset, J)$$

l'élément universel est (I, ϕ) en effet : $\forall (x, y) \in T$

$$(I, \phi) \vee (x, y) = (I \cup x, \phi \cap y) = (I, \phi).$$

Si $|I| = p$ et $|J| = q$ T a 2^{p+q} éléments.

L'ensemble S des atomes de T est :

$$S = \{(\phi, J - y_i) ; y_i \in J\} \cup \{(x_i, J) ; x_i \in I\}$$

en effet :

$$(\phi, J - y_i) \geq_1 (X, Y) \iff X = \phi \text{ et } J - y_i \subset Y$$

si on veut un ordre strict $Y = J$ le seul élément inférieur à $(\phi, J - y_i)$ est donc bien l'élément nul de même :

$$(x_i, J) \geq_1 (X, Y) \iff x_i \supset X \text{ et } J \subset Y \text{ pour un ordre strict } X = \phi.$$

Tout atome est élément de S. En effet nous savons que tout élément V-irréductible d'un treillis de Boole est un atome et que :

Théorème : (Th.5 p. 139 [2])

L'ensemble des éléments v-irréductibles d'un treillis distributif de longueur finie n est d'ordre n.

T est de longueur $p + q$ car il contient la chaîne suivante :

$$(\phi, J) \prec (x_1, J) \prec (x_1 x_2, J) \prec \dots \prec (I, J) \prec (I, J - y_1) \dots \prec (I, \phi)$$

or $|S| = p + q$ donc S est bien l'ensemble des atomes de T.

4.2.2. Cas particulier.

Si I et J sont les ensembles d'indices d'un tableau \mathcal{C} , notons T_1 l'ensemble des sous-tableaux de \mathcal{C} remplis d'un élément unique a : T_1 est un sous-treillis de T si on admet que (X, ϕ) et (ϕ, Y) sont des éléments de T_1 . En effet

$$\forall (X_1, Y_1), (X_2, Y_2) \in T_1 :$$

$$(X_1, Y_1) \vee (X_2, Y_2) = (X_1 \cup X_2, Y_1 \cap Y_2) \text{ sous-tableau rempli de } a$$

$$(X_1, Y_1) \wedge (X_2, Y_2) = (X_1 \cap X_2, Y_1 \cup Y_2) \text{ sous-tableau rempli de } a$$

T_1 étant un sous-treillis d'un treillis de Boole est distributif. Les éléments \vee -irréductibles sont les éléments $(\phi, J - y_i)$ $\forall i \in J$ puisque ce sont des atomes et (x_i, Y_i) Y_i étant maximum (lignes de a).

En effet :

$$(x_i, Y_i) = (A, B) \vee (C, D) \text{ entraîne } \begin{cases} A = x_i \text{ ou } \phi \\ C = \phi \text{ ou } x_i \end{cases}$$

$$\text{et } B = D = Y_i$$

Si $A = x_i$ par exemple Y_i étant maximum et $B \supset Y_i$ $B = Y_i$. L'ensemble de ces éléments est d'ordre $p + q$ ce sont donc les seuls éléments \vee -irréductibles d'après le théorème précédent. Nous dirons qu'un élément (X, Y) de T_1 est premier s'il est maximal par rapport à l'inclusion c'est-à-dire s'il n'existe pas d'élément (X', Y') de T_1 avec $X' \supset X$ et $Y' \supset Y$.

4.2.3. Simplification de l'Algorithme de Malgrange.

T_1 est \vee -expressible à partir de ses éléments \vee -irréductibles donc en particulier ses éléments premiers, T_1 étant distributif un élément de T_1 a une expression irrédondante unique sous forme d'union d'éléments \vee -irréductibles, les éléments $(\phi, J - y_i)$ n'interviennent pas dans l'expression irrédondante d'un élément premier de T_1 .

En effet soit (A, B) un élément premier de T_1

$(A, B) = (X_1, Y_1) \vee (\phi, \mathcal{J} - y_{i1}) \vee \dots \vee (\phi, \mathcal{J} - y_{is})$ expression irrédondante

$(X_1, Y_1) = \vee$ (des éléments \vee -irréductibles de la forme (x_i, Y_i)).

$(\phi, \mathcal{J} - y_{i1}) \vee \dots \vee (\phi, \mathcal{J} - y_{is}) = (\phi, \mathcal{J} - y_{i1} \cup y_{i2} \cup \dots \cup y_{is})$

$(X_1, Y_1) \vee (\phi, \mathcal{J} - y_{i1} \cup \dots \cup y_{is}) = (X_1, C)$ donc $X_1 = A$

$C = Y_1 \cap (\mathcal{J} - y_{i1} \cup \dots \cup y_{is}) = B \quad (\mathcal{J} - y_{i1} \cup \dots \cup y_{is}) \supset Y_1$

sinon (A, B) ne serait pas premier puisque inclus dans (A, Y_1) .

Soient $(c_i)_{i=1 \dots p}$ les éléments \vee -irréductibles de T_1 :

$c_i = (x_i, Y_i) \quad x_i \in I$. On forme $c_i \vee c_j$ si $Y_i \cap Y_j \neq \phi$ les éléments premiers sont obtenus par union de ces éléments. (On a le même résultat en considérant les colonnes de $a : (X_j, y_j) \quad X_j$ maximum, et l'intersection \wedge).

Exemple :

$$\mathcal{C} = \begin{array}{cccccc|l} 0 & 1 & 0 & 1 & 1 & 1 & A \\ 1 & 0 & 1 & 0 & 0 & 0 & B \\ 0 & 1 & 0 & 1 & 1 & 0 & C \\ 0 & 1 & 1 & 0 & 0 & 1 & D \\ \hline & a & b & c & d & e & f \end{array}$$

$I = (A, B, C, D)$

$\mathcal{J} = (a, b, c, d, e, f)$

$c_1 = (A, bdef) \times$

$c_2 = (B, ac) \times$

$c_3 = (C, bde)$

$c_4 = (D, bcf) \times$

$c_1 \vee c_3 = (AC, bde) \times$

$c_1 \vee c_4 = (AD, bf) \times$

$c_2 \vee c_4 = (BD, c) \times$

$c_3 \vee c_4 = (CD, b)$

$c_1 \vee c_3 \vee c_4 = (ACD, b) \times$

les éléments premiers sont marqués par une croix.

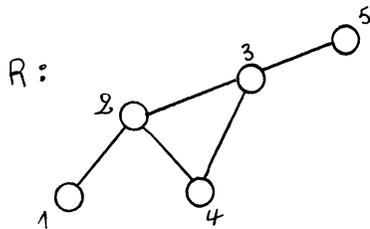
Chaque élément n'est calculé qu'une fois, l'idempotence des opérations évite des calculs inutiles et on n'utilise qu'une opération.

Remarques :

Lorsqu'on utilise l'algorithme de Malgrange la couverture de départ est généralement celle constituée par les lignes (ou les colonnes) de a.

D'autre part si on considère une couverture quelconque en utilisant uniquement l'opération \wedge (resp. \vee) on peut obtenir la couverture constituée des lignes (resp. colonnes) de a, la démonstration de l'algorithme est donc évidente.

Exemple : recherche des fractures minimales du réseau R.



$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 & 1 & 3 \\ 0 & 1 & 1 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 & 1 & 5 \\ 1 & 2 & 3 & 4 & 5 & \end{bmatrix}$$

$$c_1 = (1,345) \times$$

$$c_2 = (2,5)$$

$$c_3 = (4,15) \times$$

$$c_4 = (3,1)$$

$$c_5 = (5,124) \times$$

$$c_1 \vee c_2 = (12,5)$$

$$c_1 \vee c_3 = (14,5)$$

$$c_1 \vee c_5 = (15,4)$$

$$c_2 \vee c_3 = (24,5)$$

$$c_3 \vee c_4 = (34,1)$$

$$c_3 \vee c_5 = (45,1)$$

$$c_4 \vee c_5 = (35,1)$$

$$c_1 \vee c_2 \vee c_3 = (124,5) \times$$

$$c_3 \vee c_4 \vee c_5 = (345,1) \times$$

les éléments premiers sont marqués par une croix, les fractures minimales sont {2} et {3}.

(Il était inutile de calculer les \vee de 3 éléments puisque les \vee de 2 éléments n'avaient qu'un élément comme 2^{ème} membre).

CHAPITRE II

DECOMPOSITION D'UN RESEAU D'ARTICULATIONS EN K SOUS-RESEAUX CONNEXES

1 - RESEAU REGULIEREMENT DECOMPOSABLE

1.1. Définitions.

Un réseau connexe R de m articulations (a_1, a_2, \dots, a_m) est dit régulièrement décomposable si on peut ordonner ses articulations de telle sorte que :

$$\left. \begin{array}{l} (a_1 ; a_2 \dots a_m) \\ (a_1 a_2 ; \dots a_m) \\ \dots\dots\dots\dots\dots\dots \\ (a_1 a_2 \dots a_{m-1} ; a_m) \end{array} \right\}$$

soient des hypercoupures simples de R.

R est totalelement décomposable si quel que soit l'entier k $0 \leq k \leq m$ on peut trouver une décomposition de R en q + 1 sous-réseaux connexes

$$(R_i)_{i=1,2,\dots,q+1} \text{ avec : } \left\{ \begin{array}{ll} m = q \cdot k + r & r < k \\ |R_i| = k & i = 1, 2, \dots, q \\ |R_{q+1}| = r & \end{array} \right.$$

La puissance d'un sous-réseau est égale au nombre de composantes connexes engendrées lorsqu'on le supprime.

1.2. Une condition nécessaire et suffisante pour qu'un réseau d'articulations soit régulièrement décomposable.

Théorème 1 :

Une condition nécessaire et suffisante pour qu'un réseau connexe R soit régulièrement décomposable entre 2 articulations a et b est qu'il existe un couple d'articulations (a, b) tel que :

quel que soit $m \in R$ il existe une chaîne élémentaire (amb).

La condition est nécessaire. En effet supposons que R soit régulièrement décomposable entre les articulations (a, b) et qu'il existe $m \in R$ tel qu'il n'existe pas de chaîne élémentaire (amb).

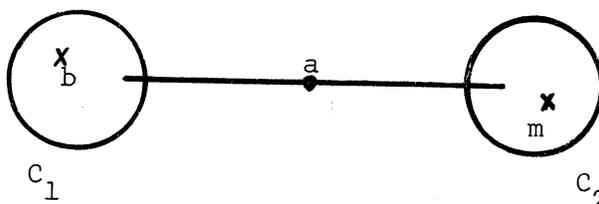
Considérons la liste des HS :

$$\left\{ \begin{array}{l} (a ; \dots b) \\ (ac ; \dots b) \\ \dots \\ (ac \dots m ; \dots d \dots b) = H \\ \dots \\ (ac \dots m \dots ; b) \end{array} \right.$$

le sous-réseau engendré par le 1^{er} membre de H est connexe il existe donc une chaîne élémentaire (am) = μ , soit d un élément du 2^{ème} membre de H adjacent à m il existe une chaîne élémentaire (db) = ν constitué uniquement d'éléments du 2^{ème} membre de H : $\mu\nu$ est une chaîne élémentaire de R joignant a à b en passant par m il y a donc contradiction.

La condition est suffisante : soit (a, b) un couple satisfaisant à la condition.

(1) - $(a ; \dots)$ est une HS sinon : soient C_1 et C_2 deux composantes connexes engendrées par la suppression de a , C_1 contenant b par exemple. Si m est une articulation de C_2 il n'existe pas de chaîne élémentaire (amb) il y a donc contradiction.



(2) - Considérons une hypercoupure simple $H = (\underbrace{a \dots}_{C_1} ; \underbrace{\dots b}_{C_2})$,

Nous allons montrer qu'il existe $\alpha \in C_2$ et adjacent à un élément de C_1 tel que l'on puisse joindre tout élément $c \in C_2$ à b dans C_2 sans passer par α , c'est-à-dire que l'hypercoupure $(C_1 \alpha ; C_2 - \alpha)$ est simple.

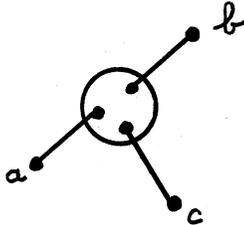
Considérons $\alpha \in C_2$, adjacent à C_1 avec $d_{C_2}(\alpha, b)$ maximum. ($d_{C_2}(x, y)$ représentant la longueur d'une plus courte chaîne de C_2 joignant x à y).

Si toutes les chaînes joignant c à b dans C_2 passent par α il existe une chaîne $\mu = (ac)$ qui ne passe pas par α puisque, par hypothèse, il existe une chaîne élémentaire (acb) . Soit α' la première articulation de C_2 sur $\mu = (ac)$, on note v la sous-chaîne $(\alpha'c)$ de μ . Il existe une chaîne de C_2 $\xi = (\alpha'b)$ qui ne passe pas par α sinon $d_{C_2}(\alpha', b) > d_{C_2}(\alpha, b)$ donc $v\xi$ est une chaîne de C_2 qui joint c à b sans passer par α .

(3) - A partir de l'hypercoupure simple $(a ; \dots b)$ (2) nous permet de construire une liste d'hypercoupures simples satisfaisantes.

Propriété 1 :

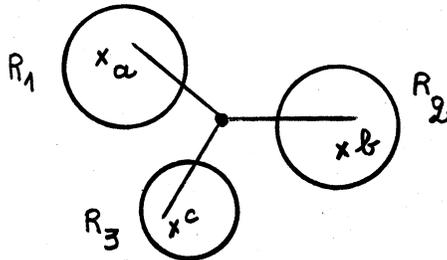
Si un réseau est régulièrement décomposable il a au plus deux articulations pendantes (qui sont les articulations (a, b) entre lesquelles le réseau est régulièrement décomposable). En effet il n'existe pas de chaîne élémentaire (xay) quels que soient $x, y \in R$ $x \neq a$ et $y \neq a$. Si nous supposons que $x = a$ par exemple il n'existe pas de chaîne élémentaire (aby) quel que soit $y \in R$ $y \neq b$ si $y = b$ il n'existe pas de chaîne élémentaire (acb).



chaîne élémentaire (aby) quel que soit $y \in R$ $y \neq b$ si $y = b$ il n'existe pas de chaîne élémentaire (acb).

Propriété 2 :

Si un réseau a une charnière de puissance ≥ 3 il n'est pas régulièrement décomposable.



En effet il n'existe pas de chaîne élémentaire (abc) par exemple :

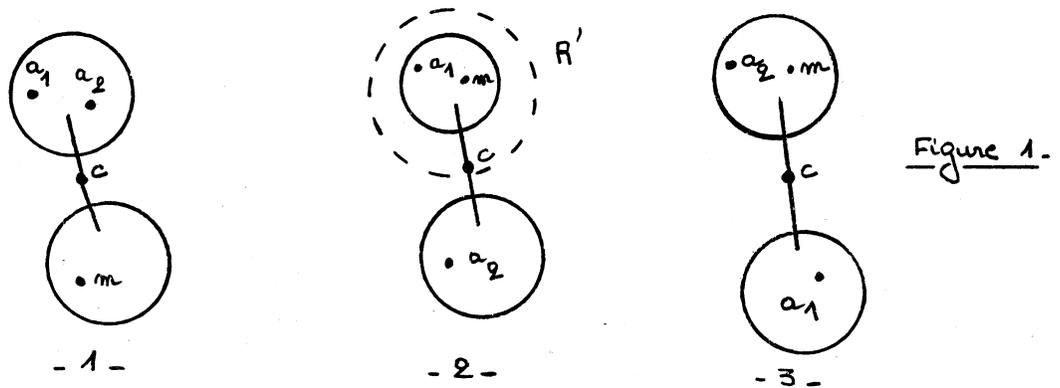
$$\left\{ \begin{array}{l} \forall a \in R_1 \\ \forall b \in R_2 \\ \forall c \in R_3 \end{array} \right.$$

1.3. Propriété des charnières d'un réseau régulièrement décomposable.

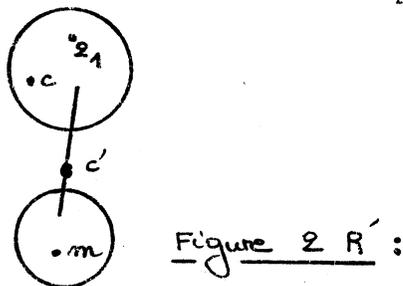
Lemme :

Si trois articulations a_1, a_2, m d'un réseau connexe sont telles qu'il n'existe pas de chaîne élémentaire $(a_1 m a_2)$ alors il existe une charnière c de R séparant (a_1, a_2) et m .

En effet : d'après le théorème 2 de 3.1.3. les articulations a_1, a_2, m ne sont pas dans un même sous-réseau connexe sans charnière. Si R n'a qu'une charnière c elle fractionne donc $(a_1 a_2 m)$ c'est-à-dire que l'on a l'un des 3 cas suivants :



le théorème est alors vrai car dans le cas -1- c sépare $(a_1 a_2)$ et m dans les cas 2 et 3 il existe une chaîne élémentaire $(a_1 m a_2)$ supposons que le théorème soit vrai lorsque R a au plus k charnières et considérons un réseau de $k + 1$ charnières nous avons de nouveau les 3 cas de la figure 1. Dans le cas 1 le théorème est vrai, dans le cas 2 (idem pour 3) s'il n'existe pas de chaîne élémentaire $(a_1 m a_2)$ c'est qu'il n'existe pas de chaîne élémentaire $(a_1 m c)$ or le réseau R' a k charnières on peut donc appliquer l'hypothèse de récurrence : il existe une charnière c' de R' séparant $(a_1 c)$ de m (figure 2) mais alors c' sépare bien $(a_1 a_2)$ et m dans R .



Théorème 2 :

Un réseau connexe sans charnière de puissance ≥ 3 est régulièrement décomposable entre a et b si et seulement si toutes ses charnières séparent a et b.

La condition est nécessaire : supposons qu'il existe une charnière c de R ne séparant pas a et b : il existe pas de chaîne élémentaire (amb) quel que soit m élément de la composante ne contenant ni a ni b (figure 3).

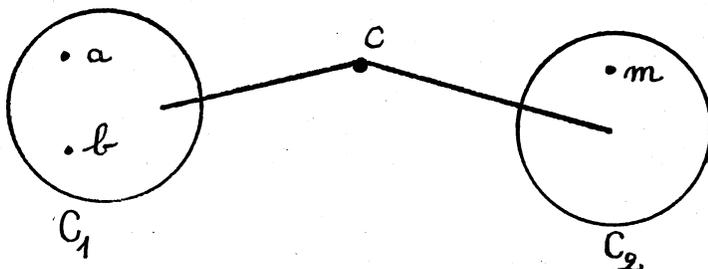


Figure 3 -

La condition est suffisante : nous allons démontrer que si toutes les charnières de R séparent a et b il existe une chaîne élémentaire (amb) quel que soit $m \in R$.
Considérons une charnière c de R :

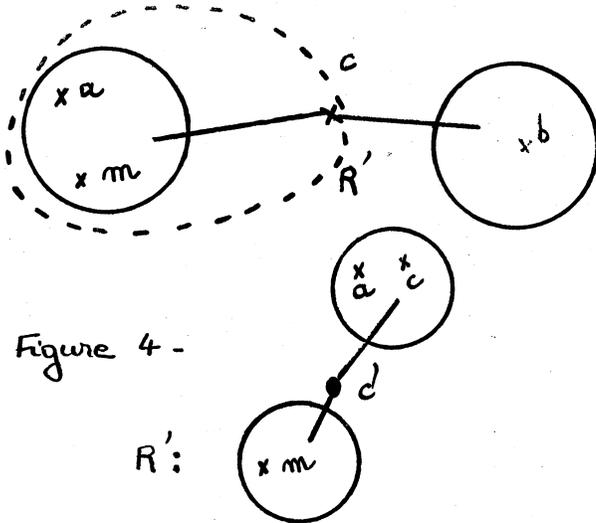


Figure 4 -

s'il n'existe pas de chaîne élémentaire (amb) ceci entraîne qu'il n'existe pas de chaîne élémentaire (amc) donc que le sous-réseau R' a une charnière c' séparant (ac) de m, c'est-à-dire que R' a la forme suivante (figure 4)

mais alors c' qui est une charnière de R ne sépare pas a et b ou est de puissance ≥ 3 il y a donc contradiction (figure 5)

La démonstration est identique dans le cas où l'on prend m dans C_2 .

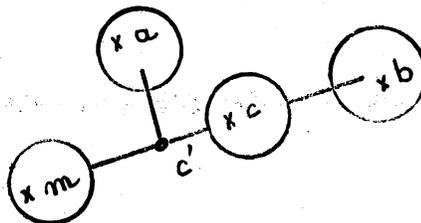


Figure 5 -

Les théorèmes 1 et 2 nous permettent de savoir simplement :

- si un réseau est régulièrement décomposable
- les articulations extrêmes par rapport auxquelles il est régulièrement décomposable.
- d'obtenir une liste d'HS convenant dans le cas favorable.

1.4. Algorithme de reconnaissance d'un réseau régulièrement décomposable.

(1) - Si R a plus de deux articulations pendants R n'est pas régulièrement décomposable sinon aller en (2).

(2) - Si R a une charnière de puissance ≥ 3 R n'est pas régulièrement décomposable sinon aller en (3).

(3) - En notant $(c_i)_{i \in I}$ les charnières de R et $(C_1^i, C_2^i)_{i \in I}$ les composantes connexes qu'elles engendrent on pose :

$$A_k = \bigcap_{j \in I} C_j^i$$

$$\left\{ \begin{array}{l} i \in I \\ j = 1 \text{ ou } 2 \end{array} \right.$$

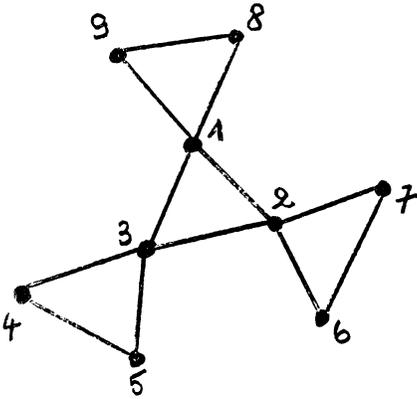
$$B_k = \bigcap_{j' \in I} C_{j'}^i$$

$$\left\{ \begin{array}{l} i \in I \\ j' = 1 \text{ si } j = 2 \\ j' = 2 \text{ si } j = 1 \end{array} \right.$$

tout couple (a, b) $a \in A_k$ $b \in B_k$ convient (si $|I| = n$ on a 2^{n-1} intersections à envisager).

(4) - Ayant un couple de points extrêmes la démonstration du théorème 1 nous fournit une liste d'HS.

Exemple :



les charnières sont 1, 2, 3 :

$$C_1^1 = (8, 9) \quad C_2^1 = (2, 3, 4, 5, 6, 7)$$

$$C_1^2 = (6, 7) \quad C_2^2 = (1, 3, 4, 5, 8, 9)$$

$$C_1^3 = (4, 5) \quad C_2^3 = (1, 2, 6, 7, 8, 9)$$

Il n'existe aucun couple de points séparés par toutes les charnières :

$$\bigcap_{i \in I} C_1^i = \emptyset \quad C_1^1 \cap C_2^2 \cap C_1^3 = C_2^1 \cap C_1^2 \cap C_1^3 = C_1^1 \cap C_1^2 \cap C_2^3 = \emptyset$$

1.5. Réseau sans charnière.

Théorème 3 :

Un réseau connexe est sans charnière si et seulement si il est régulièrement décomposable à partir de tout couple d'éléments.

La condition est nécessaire : si R a une charnière c engendrant les composantes connexes C_1, C_2, \dots ($c ; \dots$) n'est pas une hypercoupure simple donc R n'est régulièrement décomposable à partir d'aucun couple contenant c.

La condition est suffisante : Nous allons construire une liste de HS satisfaisantes à partir d'un couple quelconque $(a, b) \cdot (a ; \dots b)$ est simple puisque le réseau est sans charnière, nous raisonnons par récurrence sur le nombre d'éléments dans le l^{er} membre. Supposons que $(a \underbrace{a_1 \dots a_r}_{C_1} ; \dots \underbrace{b}_{C_2}) = H$ soit

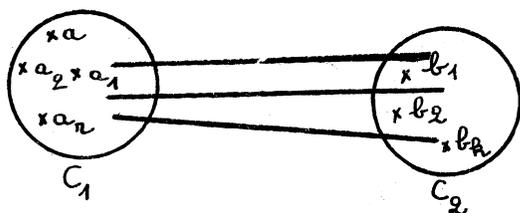
simple et notons C_1 et C_2 les sous-réseaux connexes engendrés par les deux membres de H.

On note $(b_i)_{i \in I}$ les éléments de C_2 adjacents à au moins un élément de C_1 et $\neq b$.

Nous raisonnons par l'absurde en supposant

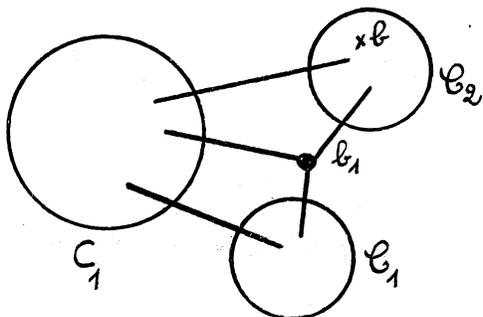
qu'il n'existe aucun b_i tel que

$(\underbrace{a_1 \dots a_r}_{C_1} b_i ; \dots)$ soit simple.



Si $(\underbrace{a \dots a_r}_{C_1} b_1 ; \dots b)$ n'est pas simple $(\underbrace{a \dots a_r}_{C_1} ; b_1 \dots b)$ étant

simple ceci entraîne que b_1 est une charnière de C_2 dont la suppression détermine au moins 2 composantes connexes \mathcal{C}_1 et \mathcal{C}_2 .



Dans \mathcal{C}_1 et dans \mathcal{C}_2 il existe au moins un

b_i sinon b_1 serait charnière de R ce qui

est contraire à l'hypothèse. Si $b \in \mathcal{C}_2$

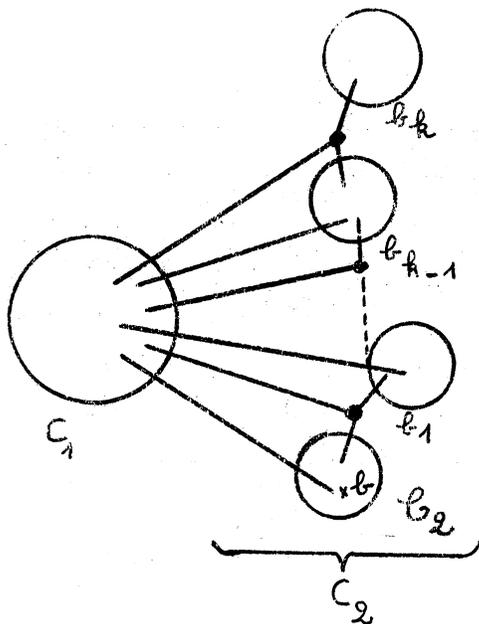
par exemple le nombre $b_i \in \mathcal{C}_1$ est stric-

tement inférieur au nombre de b_i éléments

de C_2 , s'il n'existe pas de b_k tel que

$(C_1 b_k ; \dots)$ soit simple ce nombre de

b_i deviendra nul et le dernier b_i considéré sera une charnière de R ce qui est contraire à l'hypothèse.



Théorème 4 :

Un réseau R est sans charnière si et seulement si quels que soient trois articulations a, b, c de R il existe une chaîne élémentaire (abc) .

Evident d'après les théorèmes 1 et 3.

Remarque :

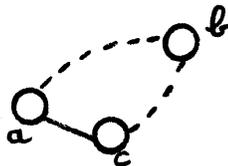
La démonstration des théorèmes 1 et 3 n'a pas nécessité l'utilisation du théorème de Menger (voir [1] p. 192) nous pouvons ainsi retrouver toutes les propriétés habituelles des réseaux sans charnière sans utiliser le théorème de Menger qui est de démonstration délicate.

Nous démontrons, à titre d'exemple, la propriété suivante des réseaux sans charnière :

Propriété :

Par deux éléments quelconques d'un réseau sans charnière il passe un cycle élémentaire.

En effet soient $a, b \in R$ supposé sans charnière. Considérons une articulation c adjacente à a , d'après le théorème 3 il existe une chaîne élémentaire $(abc) = \mu$ donc un cycle élémentaire passant par a, b qui est égal à μ suivi d'un couple de connexions joignant c à a .



Dans le cas où le réseau est fermé (toutes les connexions sont éléments d'un couple) on trouve dans ([3]) la démonstration directe de :

Si par deux éléments quelconques d'un réseau il passe un cycle élémentaire alors par deux couples de connexions quelconques il passe un cycle élémentaire.

1.6. Quelques propriétés.

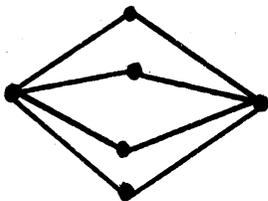
1 - Si un réseau a une chaîne hamiltonienne (chaîne élémentaire passant par toutes les articulations du réseau).

- Il est régulièrement décomposable entre les extrémités de la chaîne hamiltonienne.

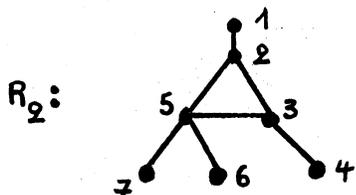
- Il est totalement décomposable. En effet soit $C = a_1 a_2 \dots a_m$ une chaîne hamiltonienne du réseau toute bipartition de cette chaîne correspond à une hypercoupure du réseau qui est donc simple de même le réseau est totalement décomposable car toute partition de la chaîne C est une décomposition en sous-réseaux connexes de R .

2 - Un réseau sans charnière n'est pas nécessairement totalement décomposable : R_1 n'admet pas de 2-décomposition.

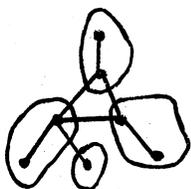
R_1 :



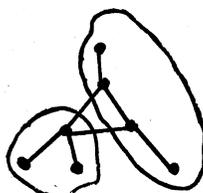
3 - Un réseau totalement décomposable n'est pas nécessairement régulièrement décomposable : R_2 a une charnière de puissance 3, 5, il est cependant totalement décomposable :



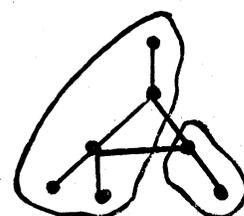
- | | |
|-----------------|-------|
| (12, 34, 56, 7) | k = 2 |
| (123, 567, 4) | k = 3 |
| (1234, 567) | k = 4 |
| (12567, 34) | k = 5 |



2-décomposition de R_2



3-décomposition
et 4-décomposition



5-décomposition

1.7. Cas d'un réseau de noeuds et d'étoiles.

On peut définir un réseau $R(X, M)$ de noeuds et d'étoiles régulièrement décomposable de la même façon que pour un réseau d'articulations, tous les résultats obtenus sont valables en remplaçant "articulation" par "étoile".

$R(X, M)$ réseau connexe de noeuds et d'étoiles est régulièrement décomposable si on peut ordonner ses m étoiles (e_1, e_2, \dots, e_m) de telle sorte que :

$(e_1 ; \dots e_m)$	}	soient des hypercoupures simples.
$(e_1 e_2 ; \dots e_m)$		
.....		
$(e_1 e_2 \dots e_{m-1} ; e_m)$		

Le théorème 1 de 1.2. devient :

Théorème 1' :

Une condition nécessaire et suffisante pour qu'un réseau connexe de noeuds X et d'étoiles M soit régulièrement décomposable entre 2 étoiles a, b est qu'il existe un couple d'étoiles (a, b) tel que :

quel que soit $m \in M$ il existe une chaîne élémentaire (amb) .

La démonstration est identique à celle du théorème 1.

Théorème 2' :

Un réseau connexe de noeuds et d'étoiles sans étoile charnière de puissance ≥ 3 est régulièrement décomposable entre a et b si et seulement si toutes ses étoiles charnières séparent a et b .

On peut ainsi transposer facilement tous les résultats sur les réseaux d'articulations aux réseaux de noeuds et d'étoiles, notamment aux réseaux de noeuds et de branches.

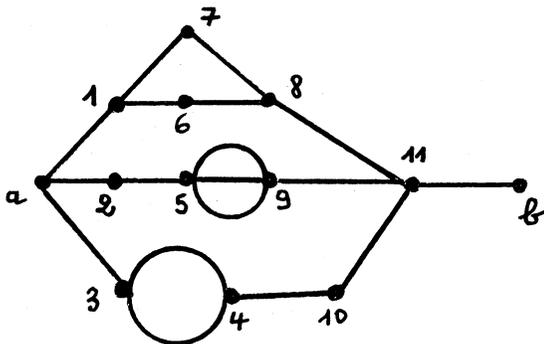
Propriété :

Un réseau série-parallèle entre deux noeuds a et b est régulièrement décomposable entre a et b .

En effet si R est série-parallèle entre a et b quelle que soit la branche α de R il existe une chaîne élémentaire joignant a à b et contenant α ([4], chapitre V) donc quel que soit le noeud c de R il existe une chaîne élémentaire (acb) , d'après le théorème 1 R est bien régulièrement décomposable entre a et b . Le théorème 2 nous permet de retrouver que toute charnière d'un réseau série-parallèle entre a et b sépare a et b et qu'un réseau série-parallèle n'a pas de charnière de puissance > 2 .

On obtient simplement des décompositions régulières d'un réseau série-parallèle entre a et b en considérant successivement les noeuds à la distance 1 de a, puis 2 etc...

Exemple :



- (a ; b)
- (a 1 ; b)
-
- (a 123 ; ... b)
- (a 1234 ; ... b)
-
- (a 1234567 ; ... b)
-
- (a 12 9 ; 10 ; 11, b)
-
- (a 12 11 ; b)

2 - DECOMPOSITION EN K SOUS-RESEAUX CONNEXES

2.1. Borne supérieure du nombre de décompositions.

Tout réseau connexe R ayant m articulations admet au moins une décomposition en k sous-réseaux connexes quel que soit $1 \leq k \leq m$.

En effet il existe au moins deux articulations non charnière ([1] p. 192) soit a l'une de celles-ci (a, R - a) est une décomposition de R en 2 sous-réseaux connexes on peut recommencer k - 1 fois sur R - a puisque $1 \leq k \leq m$. (Il existe une décomposition uniquement si k = m).

Considérons un réseau complet de m articulations et notons D_k^m le nombre de ses décompositions en k sous-réseaux connexes : il est égal au nombre de k -partitions distinctes (pas par l'ordre) d'un ensemble de m objets :

$$(1) \quad D_k^m = \frac{1}{k} \sum_{q=k-1}^{m-1} C_m^q \times D_{k-1}^q \quad \text{cette borne supérieure est exacte puisqu'}$$

atteinte pour un réseau complet.

En effet soit M un ensemble de m objets et $(A ; B)$ une bipartition de M avec $|A| = q$ $m - 1 \geq q \geq k - 1$ et $|B| = m - q$ il y a C_m^q façons de choisir A et D_{k-1}^q décompositions possibles de A en $k - 1$ sous-réseaux connexes mais alors chaque k -partition sera obtenue k fois puisqu'on ne tient pas compte de l'ordre des termes.

Quelques valeurs particulières :

$$\begin{aligned} D_2^{10} &= 511 & D_2^n &= 2^{n-1} - 1 \\ D_3^{10} &= 9.330 & D_3^n &= \frac{3^{n-1} - 2^n + 1}{2} \\ D_4^{10} &= 14.422 & D_4^n &= \frac{4^n - 3(3^n - 2^{n-1}) - 1}{6} \end{aligned}$$

2.2. Propriété Générale.

Toute décomposition en $k + 1$ sous-réseaux connexes $P_{k+1} = \{R_1, R_2 ; \dots R_{k+1}\}$ d'un réseau connexe R peut-être obtenue à partir de k hypercoupures simples de $R (C_i)_{i=1,2,\dots,k}$ par la formule suivante :

$$(2) \quad P_{k+1} = \{C_1, C_2 - C_1, \dots, C_k - (C_1 \cup \dots \cup C_{k-1}), C'_1 \cap C'_2 \cap \dots \cap C'_k\}$$

réciroquement à partir de k hypercoupures simples $(C_i)_{i=1, \dots, k}$ de R , que l'on peut ordonner de telle sorte que :

$$\left\{ \begin{array}{l} C_2 - C_1 \neq \emptyset \\ \dots\dots\dots \\ C_k - (C_1 \cup \dots \cup C_{k-1}) \neq \emptyset \\ C'_1 \cap C'_2 \dots \cap C'_k \neq \emptyset \text{ et connexe} \end{array} \right.$$

on obtient une décomposition en au moins $k + 1$ sous-réseaux connexes de R .

Démonstration :

Nous raisonnons par récurrence sur k sur le réseau condensé associé à la décomposition. Si $k = 1$ la propriété est vraie : en effet une HS de R , différente de $(M ; \emptyset)$, définit une décomposition en 2 sous-réseaux connexes ; supposons la propriété vraie quel que soit $p \leq k$ et soit :

$P_{k+1} = \{R_1, R_2, \dots, R_{k+1}\}$ une décomposition ayant R_C pour réseau condensé associé ($|R_C| = k + 1$). Il existe au moins un élément R_i de R_C tel que $R_C - R_i$ soit connexe, supposons, par exemple, que ce soit R_{k+1} . La partition $P_k = \{R_1, R_2, \dots, R_k\}$ de $R - R_{k+1}$ peut être obtenue à partir de $k - 1$ HS de $R - R_{k+1} : (C_i)_{i=1, \dots, k-1}$ si ces hypercoupures sont également simples dans R :

$$\left\{ \begin{array}{l} H_i = C_i \quad \forall i \leq k - 1 \\ H_k = R_{k+1} \end{array} \right.$$

on a k HS de R qui déterminent P_{k+1} par (2). Si C_i est non simple dans R , R_{k+1} n'est adjacent qu'à des éléments de C_i donc $(C_i \cap R_{k+1} ; \bar{C}_i)$ est simple dans R .

Si ℓ est le plus petit i tel que C_ℓ ne soit pas simple en posant :

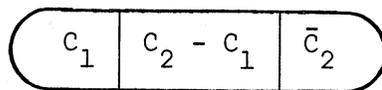
$$L \left\{ \begin{array}{l} H_i = C_i \quad \forall i < \ell \\ H_\ell = R_{k+1} \\ H_{\ell+1} = R_{k+1} C_\ell \\ H_{i+1} = \begin{cases} R_{k+1} C_i & \text{si } C_i \text{ n'est pas simple dans } R \quad i > \ell \\ C_i & \text{autrement} \end{cases} \end{array} \right.$$

cette liste d'HS de R permet bien d'obtenir P_{k+1} avec (2).

Remarques :

Un grand nombre de suites distinctes d'hypercoupures simples satisfaisant à la condition précédente pourront donner la même décomposition. Par exemple pour $k = 3$:

$$\begin{aligned} (C_1, C_2 - C_1, \bar{C}_1 \cap \bar{C}_2) &= (\bar{C}_2, \bar{C}_2 - \bar{C}_1, C_2 \cap C_1) = (\bar{C}_2, \bar{C}_2 - C_1, C_2 \cap \bar{C}_1) = \\ (C_1, \bar{C}_2 - C_1, \bar{C}_1 \cap C_2) &= (C_1, C_2 \cap \bar{C}_1, \bar{C}_1 \cap \bar{C}_2) \text{ quand on a :} \end{aligned}$$



Pour $k > 2$ il est difficile de savoir si deux suites distinctes conduiront aux mêmes décompositions ce qui fait que l'utilisation de (2) est rapidement limitée par le nombre de HS de R et par k .

2.3. Réseau condensé régulièrement décomposable.

Nous dirons que k HS $(C_i)_{i=1,2,\dots,k}$ forment une chaîne si on peut les ordonner totalement par inclusion de leur premier membre. Toute décomposition en $k + 1$ sous-réseaux connexes ayant un réseau condensé régulièrement décomposable peut-être obtenue par (3) $(C_i)_{i=1,2,\dots,k}$ étant une chaîne de k HS avec

$$\bigcup_{i=1}^k C_i \neq R :$$

$$(3) \quad P_{k+1} = \{C_1, C_2 - C_1, \dots, C_k - C_{k-1}, \bar{C}_k\}$$

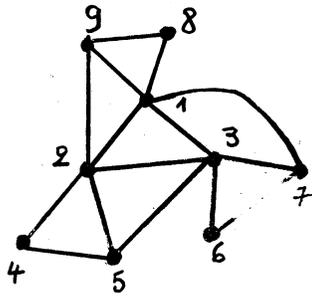
avec $C_i \subset C_{i+1} \quad \forall i < k \quad \bigcup_{i=1}^k C_i \neq R$

En effet si le réseau condensé a pour ensemble d'hypercoupures simples :

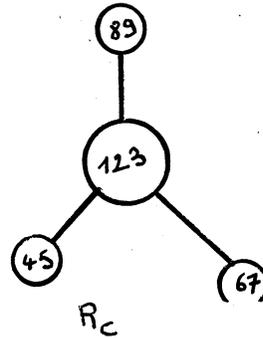
$$\left. \begin{array}{l} (a ; \dots b) \\ (a a_1 ; \dots b) \\ \dots\dots\dots \\ (a a_1 \dots a_{k-1} ; b) \end{array} \right\} \text{on prend} \left\{ \begin{array}{l} C_1 = a \\ C_2 = a a_1 \\ \dots\dots\dots \\ C_k = a a_1 \dots a_{k-1} \end{array} \right.$$

Remarque :

Un réseau régulièrement décomposable peut conduire à une décomposition n'ayant pas un réseau condensé associé régulièrement décomposable :



R est sans charnière



n'est pas régulièrement décomposable

Nous scindons le problème initial en 2 :

a) Etude des décompositions ayant un réseau condensé régulièrement décomposable.

b) Etude des autres décompositions.

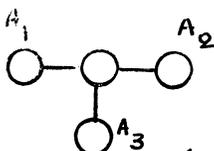
La condition pour une suite de HS d'être une chaîne est beaucoup plus restrictive que la condition de la propriété générale 2.2. Il n'y a pas de test de connexité à faire sur le dernier terme qui est égal à \bar{C}_k : ses raisons font que l'étude des décompositions ayant un réseau condensé associé régulièrement décomposable est moins compliquée que l'étude des autres.

Pour étudier les autres décompositions il est inutile de chercher tous les réseaux connexes de k articulations non régulièrement décomposables, en effet les HS d'un réseau partiel sont des HS du réseau. Le seul arbre régulièrement décomposable étant la chaîne on part d'un arbre à k éléments (différent de la chaîne) et on construit les réseaux non régulièrement décomposables en lui rajoutant des couples de connexions.

k = 3.

Le seul arbre ayant 3 éléments est la chaîne : il n'y a pas de décomposition du type b). Toutes les décompositions en 3 sous-réseaux connexes d'un réseau connexe s'obtiennent par la formule simple (3).

k = 4.

Le seul arbre différent de la chaîne est 

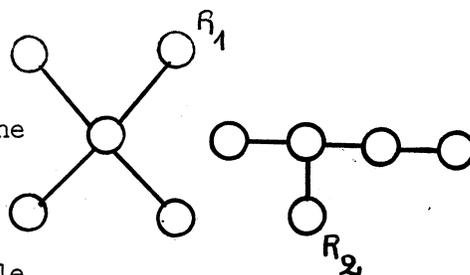
On voit facilement que c'est le seul réseau de 4 éléments non régulièrement décomposable. Pour les décompositions de type b) il suffira d'étudier les ensembles de 3 HS A_1, A_2, A_3 telles que :

$$A_1 \cap A_2 = A_1 \cap A_3 = A_2 \cap A_3 = \emptyset \quad A_1 \cup A_2 \cup A_3 \neq M$$

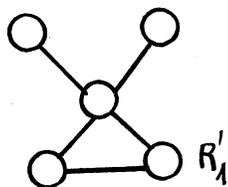
les A_i n'étant pas liés entre eux.

k = 5.

Deux arbres sont différents de la chaîne



un autre réseau est non régulièrement décomposable



On peut encore étudier simplement les décompositions du type b) :

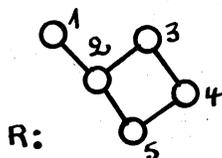
- Chercher 4 HS disjointes deux à deux, ayant une union différente de M et telles que 2 d'entre elles au plus soient liées (R_1 et R'_1).

- Chercher 4 HS trois d'entre elles étant disjointes 2 à 2 et non liées, la dernière étant liée à une seule des trois autres et la contenant.

k = 6.

Il existe 5 arbres non régulièrement décomposables et conduisant à de nombreux réseaux non régulièrement décomposables la méthode devient lourde.

Exemple :



Nous cherchons toutes les décompositions en 3 sous-réseaux connexes de R.

liste des hypercoupures simples de R :

Il y a donc 7 décompositions distinctes en 2 sous-réseaux connexes.

HS

}	(1 ; 2345)
	(3 ; 1245)
	(4 ; 1235)
	(5 ; 1234)
	(12 ; 345)
	(34 ; 125)
	(45 ; 123)

L'un des sous-réseaux connexes contient 1 et a au plus 3 éléments, ces sous-réseaux nous sont donnés par la liste HS ce sont 1, 12, 123, 125. Dans chaque cas il ne reste plus qu'à partager le reste en 2 sous-réseaux connexes on a les chaînes d'hypercoupures simples suivantes :

(1, 12) (1, 123) (1, 125) (1, 1234) (1, 1235), (1, 1245)
(12, 123) (12, 125) (12, 1234) (12, 1235) (12, 1245)
(123, 1234) (123, 1235)
(125, 1235) (125, 1245)

On obtient ainsi 11 décompositions contenant toutes les décompositions en 3 sous-réseaux connexes :

(1, 2, 345)	(3, 4, 125)	(4, 5, 123)	(5, 34, 12)
(1, 23, 45)		(4, 35, 12)	
(1, 25, 34)		(3, 45, 12)	
(1, 235, 4)			
(1, 234, 5)			
(1, 245, 3)			

La décomposition barrée est une décomposition en 4 sous-réseaux connexes : R a 10 décompositions distinctes en 3 sous-réseaux connexes. (Pour les trouver il suffit de faire un test de connexité sur le sous-réseau intermédiaire car si la partition est donnée par $(C_1, C_2 - C_1, \bar{C}_2)$ C_1 et C_2 sont connexes.

2.4. Décomposition en sous-réseaux connexes de même rayon.

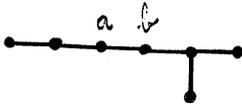
2.4.1. Position du problème.

Soit R un réseau connexe de m articulations et de rayon
 $\rho = \min_{x \in R} \max_{y \in R} d(x, y)$ la distance entre deux articulations étant mesurée par la longueur de la chaîne élémentaire la plus courte les joignant. Un élément x réalisant ce minimum est appelé un centre de R.

Problème 1 :

Chercher à décomposer R en un certain nombre de sous-réseaux connexes de même rayon $r < \rho$.

Ce problème n'a pas toujours de solution quels que soient r et R ; par exemple il est impossible de décomposer le réseau ci-dessous en sous-réseaux connexes de rayon 2 :



$$R : \begin{cases} \rho = 3 \\ r = 2 \\ c_1 = a \\ c_2 = b \end{cases}$$

Le problème que nous allons résoudre sera le suivant :

Problème 2 :

Chercher toutes les décompositions d'un réseau connexe en un nombre minimum de sous-réseaux connexes ayant un rayon $\leq r < \rho$.

Un ensemble d'éléments $C = (c_i)_{i \in I}$ de R est une r -couverture de R si tout élément de R est à une distance $\leq r$ d'un c_i .

Théorème : (voir [3])

Un ensemble d'éléments C est une r -couverture de R si et seulement si c est une 1-couverture de R^r .

R^r étant le réseau ayant pour éléments ceux de R , deux éléments étant adjacents s'ils sont à une distance $\leq r$ dans R .

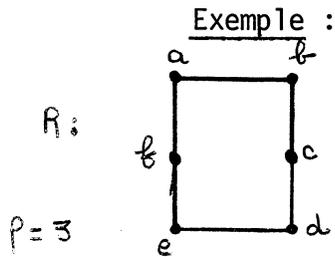
La connaissance des 1-couvertures (ensemble extérieurement stable) minimales de R^r nous donne les r -couvertures minimales du réseau donc tous les systèmes minimaux de centres de sous-réseaux de rayon $\leq r$. Pour résoudre le problème 2 il ne nous reste plus qu'à savoir construire toutes les décompositions de R associées à un ensemble de centres.

2.4.2. Algorithme.

- Les 1-couvertures s'obtiennent en représentant chaque élément du réseau par une lettre en écrivant l'ensemble des éléments adjacents à $x \in R$ sous forme d'une somme de lettres, on fait ensuite le produit de toutes les sommes obtenues quand x parcourt R : les plus petits monômes obtenus en développant ce produit sont les ensembles extérieurement stables minimaux de R (voir [4]).

- Soient $(c_i)_{i=1, \dots, p}$ un ensemble de centres. On construit les sous-réseaux connexes maximaux $(R_i)_{i=1, \dots, p}$ de centre c_i et de rayon r . S'ils sont disjoints 2 à 2, ou si les seuls points communs sont des c_i il existe une seule solution, sinon il faut envisager toutes les décompositions.

Il ne reste plus qu'à résoudre un problème de couverture en écrivant sous forme de somme les réseaux contenant une articulation en faisant le produit de ces sommes pour toutes les articulations et en ne conservant que les plus petits monômes. (On peut utiliser pour cela les algorithmes de recherche des bases premières d'une fonction booléenne).

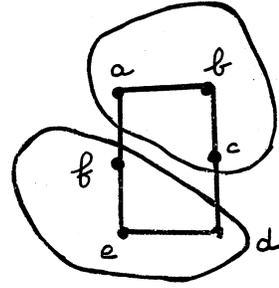
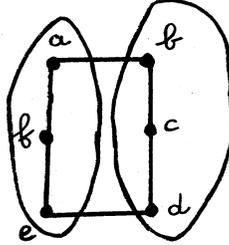
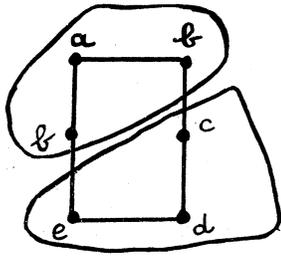


On cherche à décomposer R en un nombre minimum de sous-réseaux connexes de rayon 1.

les ensembles extérieurement stables minimaux de R sont donnés par :

$$(a+b+f) (a+b+c) (c+b+d) (d+c+e) (e+d+f) (f+e+d) = ad + cf + be + \dots$$

les systèmes de centres sont : (a, d) (c, f) (b, e) ils conduisent à des sous-réseaux disjoints il y a donc 3 décompositions satisfaisantes :



C H A P I T R E III

RESULTATS GENERAUX SUR LES ECRITURES DES FONCTIONS BOOLEENNES

1 - PRINCIPE DES METHODES

1.1. Position du problème. Définitions.

Toute fonction booléenne simple est représentable par une infinité d'expressions (parenthésées) utilisant les opérateurs somme (S) et produit (P) à partir des variables a, b, \dots et des variables complémentées a', b', \dots

Nous appellerons expression en S et P d'une fonction ϕ -booléenne simple $F = \underline{f} + \phi \bar{f}$ une expression d'une fonction booléenne g compatible avec $F : \underline{f} \leq g \leq \bar{f}$.

Une expression en S et P d'une fonction s'obtient en faisant des produits de plusieurs lettres, puis des sommes de ces expressions et ainsi de suite un nombre fini de fois. La suite alternée des symboles des opérations (S ou P) à effectuer, pour obtenir une expression E, écrite de la droite vers la gauche s'appelle la signature de E.

On peut représenter une expression en S et P par un réseau d'opérateurs arborescent d'une façon biunivoque, (à un isomorphisme près) en opérant comme pour la recherche de la signature d'une expression à partir des lettres apparaissant dans un produit, mais en remplaçant un opérateur O (S ou P) par une boîte ayant pour entrées ses opérandes O_1, O_2, \dots et pour sortie la fonction $O(O_1, O_2, \dots)$ et en n'utilisant qu'une fois chaque sortie d'opérateur.

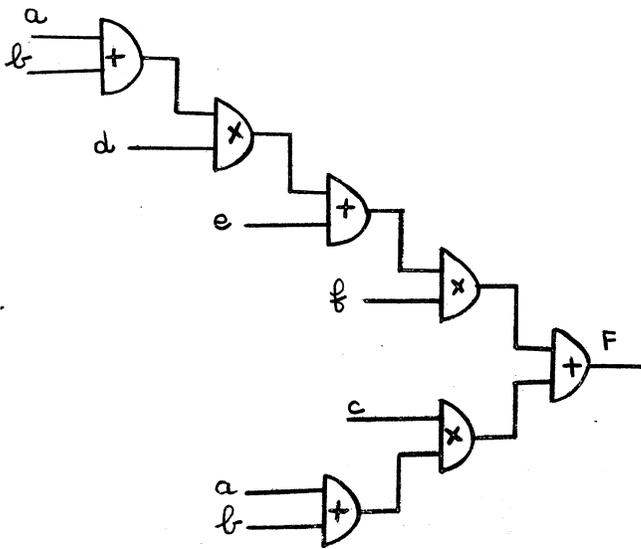
On peut également représenter une expression en S et P par un réseau de contacts série-parallèle entre une entrée et une sortie en considérant un contact par apparition de lettres : le réseau du produit (somme) de deux expressions étant représenté par la mise en série (parallèle) des réseaux des expressions.

Exemple :

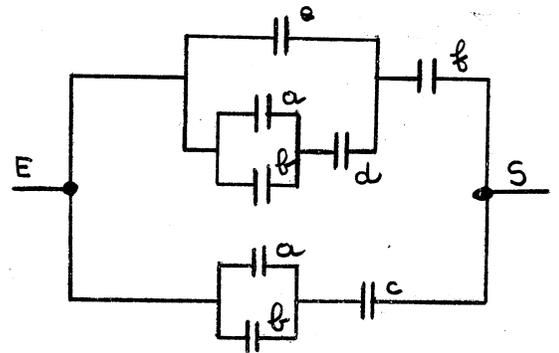
L'expression $E = (a+b)c + [(a+b)d + e] f$ de la fonction

$F = ac + bc + adf + bdf + ef$ a pour signature $\Sigma(E) = SPSPS$

son réseau d'opérateurs arborescent associé est R_E , un réseau de contacts C_E



R_E



le nombre de couches de R_E est égal au plus grand nombre d'opérateurs traversés par une même variable (longueur d'une chaîne maximale issue du dernier opérateur) c'est-à-dire au nombre d'éléments de la signature de E.

Pour comparer la complexité de deux expressions réalisant une même fonction il faut attribuer un coût à une expression. Les fonctions coût généralement considérées sont les suivantes :

- c_1 - nombre total de lettres de l'expression (nombre de contacts de C_E)
- c_2 - nombre d'opérateurs de R_E
- c_3 - nombre total d'entrées de R_E

Si l'on considère une réalisation d'une fonction par un réseau d'opérateurs ou un réseau de contacts quelconque il faut ajouter :

- c'_1 - nombre de contacts d'un réseau de contacts réalisant la fonction
- c'_2 - nombre d'opérateurs d'un réseau quelconque réalisant la fonction.

Dans le cas d'expressions ayant au plus 3 couches il existe des algorithmes ^{efficaces} permettant de minimiser rigoureusement c_1 , c_2 , c_3 . (Pour 2 couches voir [6] p. 124 pour des fonctions simples, [3] p. 65 pour des fonctions générales, pour 3 couches voir [9] et [12]).

Si les opérateurs ont 2 entrées $c_2 = c_1 - 1$.

La fonction coût considérée par la suite sera c_1 ; les expressions ayant un nombre de couches quelconque.

Les algorithmes sont heuristiques.

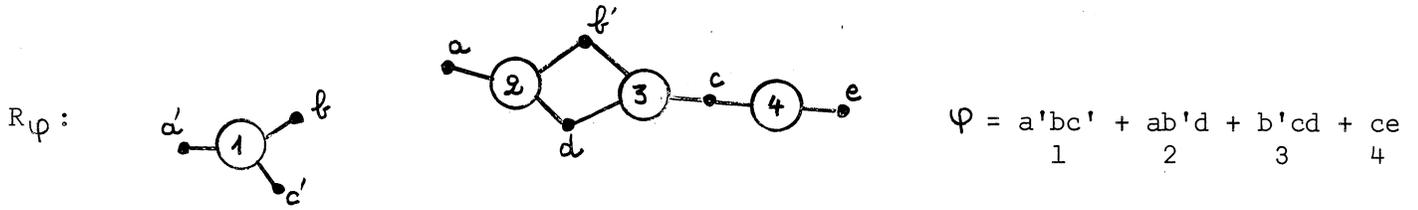
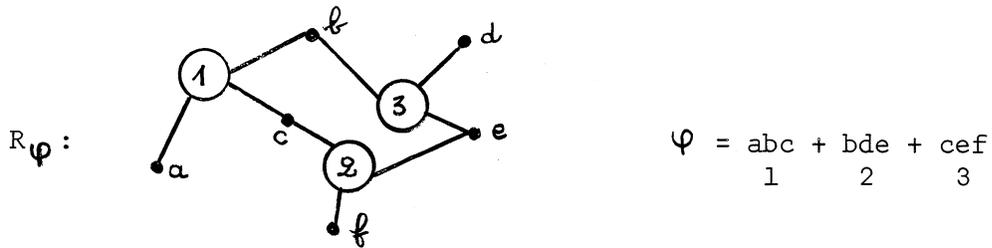
1.2. Représentation d'une somme de monômes par un réseau de noeuds et d'étoiles.

A toute somme φ de monômes croissants nous pouvons associer un réseau R_φ de noeuds et d'étoiles de la manière suivante : un monôme est représenté par une étoile ayant autant de branches que le monôme a de lettres, une variable est représentée par un noeud adjacent aux étoiles correspondant aux monômes la contenant.

La correspondance entre réseau de noeuds et d'étoiles et somme de monômes est biunivoque (à un isomorphisme près).

On peut de la même façon représenter des fonctions booléennes quelconques en considérant les variables complémentées comme de nouvelles variables.

Exemples :



1.3. Principe des méthodes.

1.3.1. Méthode par fractionnement.

Il existe deux types de méthodes permettant d'obtenir une expression d'une fonction booléenne à partir d'une famille d'opérateurs donnés et des variables.

Les méthodes par composition : on part des variables que l'on compose entre elles de toutes les façons possibles au moyen des opérateurs donnés jusqu'à ce qu'on obtienne la fonction cherchée. De telles méthodes permettent théoriquement d'obtenir des expressions minimales (au sens de c_1, c_2, c'_2) lorsque les opérateurs sont monotones, mais elles sont très longues. Nous ne nous intéresserons pas à ces méthodes. (Voir [2], [7], [10]).

Les méthodes par fractionnement : on décompose la fonction en fonctions plus simples telles qu'à partir des opérateurs et de ces fonctions on puisse réaliser la fonction et ainsi de suite jusqu'à ce qu'on obtienne les variables. De telles méthodes ont été étudiées pour les opérateurs \cup ([11]), Majorité ([10]), N_i ([5]) et S et P ([1]).

Nous les étudierons dans le cas des opérateurs S et P (et N_i) à partir d'une représentation originale de la fonction booléenne : celle par un réseau de noeuds et d'étoiles, ce qui permet notamment de définir des critères précis de fractionnement.

1.3.2. Fonction complète.

Soit $\varphi(X)$ une fonction booléenne complète, nous pouvons la fractionner en somme :

$$\varphi(X) = \varphi_1(X_1) + \varphi_2(X_2) + \dots + \varphi_p(X_p)$$

les (X_i) réalisent une partition de X ,

ou en produit :

$$\varphi(X) = \varphi_1(X_1) \times \varphi_2(X_2) \times \dots \times \varphi_p(X_p) \text{ ce qui correspond à}$$

un partage en somme de φ^* :

$$\varphi^*(X) = \varphi_1^*(X_1) + \dots + \varphi_p^*(X_p)$$

Un seul des partages suffit donc et nous utiliserons la décomposition en sous-réseaux connexes du réseau représentant une somme de monômes égale à la fonction φ (ou φ^*).

C'est le critère de choix entre la décomposition de R_φ ou celle de R_{φ^*} qui impliquera la convergence de la méthode.

1.3.3. Fonction incomplète.

Soit $F = \underline{f} + \phi \bar{f} = [\underline{f}, \bar{f}]$ une fonction incomplète. Si nous fractionnons \underline{f} en $\underline{f} = f_1 + f_2$ et si φ_1, φ_2 sont des fonctions complètes compatibles avec $[f_1, \bar{f}]$ et $[f_2, \bar{f}]$ respectivement, alors $\varphi_1 + \varphi_2$ est compatible avec $[\underline{f}, \bar{f}]$. En effet :

$$\left. \begin{array}{l} f_1 \leq \varphi_1 \leq \bar{f} \\ f_2 \leq \varphi_2 \leq \bar{f} \end{array} \right\} \text{entraîne } \underline{f} = f_1 + f_2 \leq \varphi_1 + \varphi_2 \leq \bar{f}$$

De même si nous fractionnons $\bar{f}^* = f_1^* + f_2^*$ et si φ_1 et φ_2 sont des fonctions complètes compatibles avec $[\underline{f}, f_1^*]$ et $[\underline{f}, f_2^*]$ respectivement, alors $\varphi_1 \times \varphi_2$ est compatible avec $[\underline{f}, \bar{f}^*]$. En effet :

$$\left. \begin{array}{l} \underline{f} \leq \varphi_1 \leq f_1^* \\ \underline{f} \leq \varphi_2 \leq f_2^* \end{array} \right\} \text{entraîne } \underline{f} \leq \varphi_1 \times \varphi_2 \leq f_1^* \times f_2^* = \bar{f}^*.$$

Comme dans le cas des fonctions complètes on utilisera la décomposition d'un réseau représentant une somme de monômes en sous-réseaux connexes.

Remarques :

a) Nous voyons qu'en conservant toutes les sous-fonctions obtenues en cherchant à réaliser une fonction booléenne à partir des principes ci-dessus, et en arrêtant le processus de fractionnement lorsqu'on a une fonction déjà obtenue ou une variable, on peut construire des algorithmes donnant un réseau réalisant la fonction booléenne non nécessairement arborescent. Ce qui est intéressant dans le cas où l'on cherche à minimiser la fonction coût c_2 = nombre total d'opérateurs d'un réseau réalisant la fonction.

b) De même pour obtenir un réseau réalisant une fonction booléenne générale, on réalise l'une des fonctions avec la méthode indiquée ci-dessus, puis on réalise la seconde en utilisant les sous-fonctions réalisées lors de la réalisation de la première etc... on obtient ainsi des réseaux quasi-minimaux en ce qui concerne c_2 .

c) Comme la fonction coût choisie est c_1 , nous ne considérerons que des synthèses arborescentes c'est-à-dire qu'après un fractionnement nous traiterons indépendamment les unes des autres les sous-fonctions obtenues.

2 - RESULTATS SUR LES ECRITURES MINIMALES

2.1. Décomposition d'une fonction booléenne.

2.1.1. Connexité. c-connexité.

Une fonction booléenne est dite connexe si elle n'admet pas de décomposition disjointe simple par rapport à la somme : c'est à dire s'il n'existe pas de partition (Y, Z) des variables X telle que $F(X) = \varphi(Y) + \psi(Z)$.

Théorème ([4]) :

Si $F(X)$ admet une décomposition disjointe simple par rapport à la somme suivant la partition (Y, Z) :

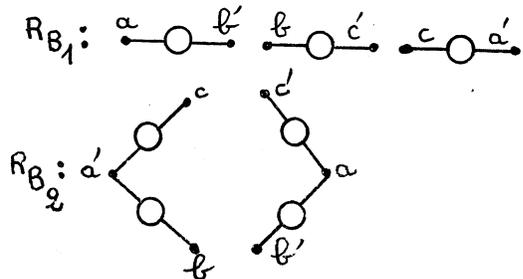
tout monôme premier de F ne contient soit que des variables de Y soit que des variables de Z et réciproquement.

Donc une fonction croissante sera connexe si et seulement si le réseau associé à sa base est connexe au sens de la théorie des réseaux.

La connexité du réseau associé à une somme de monômes sera appelée la c-connexité de cette somme de monômes. Dans le cas croissant une fonction est connexe si et seulement si le réseau associé à sa base est connexe donc la connexité est équivalente à la c-connexité. Pour une fonction non croissante, la non connexité entraîne la non c-connexité des réseaux représentant toutes les bases premières irrédondantes de la fonction. Les différentes bases premières irrédondantes d'une même fonction $F(X)$ n'ont pas nécessairement la même c-connexité :

$$B_1 = ab' + bc' + ca'$$

$$B_2 = B_3 = ab' + ba' + a'c + ac'$$



mais toutes les bases premières irrédondantes ont des réseaux associés ayant le même nombre noeuds en effet :

Théorème ([6]) :

Si une fonction est croissante en a toutes ses bases premières sont croissantes en a .

Donc quelle que soit la base première B_i de $F(X)$, les réseaux B_i ont un nombre minimum de noeuds.

2.1.2. Une condition nécessaire de décomposabilité par rapport au produit.

Propriété :

Si $F(X)$ admet une décomposition disjointe simple par rapport au produit : $F(X) = f(Y) \circ g(Z)$ (1) une base première irrédondante quelconque de F ayant k monômes est égale au produit d'une base première irrédondante de f ayant p monômes avec une de g ayant q monômes $pq = k$ et réciproquement.

En effet m_Y représentant un monôme sur les variables Y si $\sum_{i=1}^p m_Y^i$

et $\sum_{i=1}^q m_Z^i$ sont des bases premières irrédondantes de f et de g en développant

$\left(\sum_{i=1}^p m_Y^i\right) \left(\sum_{i=1}^q m_Z^i\right)$ on obtient une base première irrédondante ayant pq monômes en

effet :

nous savons qu'il n'y a pas de relation de consensus entre 2 monômes obtenus (voir [5] p. 104) et

$m_Y^i \circ m_Z^i \geq m_Y^k \circ m_Z^k \implies m_Y^i \geq m_Y^k$ et $m_Z^j \geq m_Z^l$ ce qui est impossible donc aucun monôme ne disparaît lorsqu'on développe le produit.

Réciproquement une base première irrédondante de F est de la forme :

$$F(X) = m_Y^1 \circ m_Z^1 + \dots + m_Y^k \circ m_Z^k \quad (2)$$

les monômes $(m_Y^i)_i$ sont des monômes premiers de $f(Y)$

les monômes $(m_Z^i)_i$ sont des monômes premiers de $g(Z)$

(Théorème 1 [13])

s'il y a p monômes différents parmi les k m_Y^i (et q parmi les k m_Z^i) ils forment une base première irrédondante de $f(Y)$ (resp. de $g(Z)$) car si m_Y^1 est redondant

dans $\sum_{i=2}^p m_Y^i$ les monômes $m_Y^1 \cdot m_Z^1, \dots, m_Y^1 \cdot m_Z^k$ sont redondants dans (2) ce qui est impossible.

Conséquence :

Si une fonction admet une base première irrédondante ayant un nombre premier de monômes elle n'est décomposable disjointement par rapport au produit que si elle admet un monôme en facteur $F(X) = m \cdot G(X)$.

Les diviseurs de k donneront les nombres de monômes d'une base première irrédondante des facteurs possibles de la décomposition.

2.1.3. Algorithme pour obtenir φ_1^* et φ_2^* connaissant

$$\underline{\varphi = \varphi_1 + \varphi_2, \varphi_1, \varphi_2 \text{ et } \varphi^*}.$$

Considérons une décomposition simple non disjointe d'une fonction φ par rapport à la somme :

$$\varphi(XYZ) = \varphi_1(XY) + \varphi_2(YZ) \quad X, Y, Z \text{ étant une partition}$$

de l'ensemble des variables. Nous supposons que nous disposons d'une base première irrédondante de $\varphi, \varphi_1, \varphi_2, \varphi^*$ et on cherche une expression de φ_1^*, φ_2^* sans dualiser.

Soit m un monôme premier de φ^* il s'écrit :

$m = m_X \cdot m_Y \cdot m_Z$ c'est-à-dire qu'il existe une partition de m_Y en m_Y^1 et m_Y^2 ($m_Y^1 \cdot m_Y^2 = m_Y$) telle que :

$$\left\{ \begin{array}{l} \mu = m_X \cdot m_Y^1 \quad \text{soit un monôme premier de } \varphi_1^* \\ \nu = m_Z \cdot m_Y^2 \quad \text{soit un monôme premier de } \varphi_2^* \end{array} \right.$$

En fractionnant tous les monômes de φ^* de toutes les manières possibles on obtient deux listes de monômes $(\theta_i)_{i \in I}$ et $(\eta_i)_{i \in I}$, tous les monômes de φ_1^* (resp. φ_2^*) sont dans la liste θ_i (resp. η_i). On ne conserve que ceux ayant au moins une lettre commune sous la même forme avec chaque monôme de φ_1 (pas nécessairement la même) on obtient ainsi une expression de φ_1^* .

En effet :

$$\mu \leq \varphi_1^* \iff \mu^{*'} \leq \varphi_1'$$

or tout monôme de φ_1' a une intersection vide avec tous les monômes $\leq \varphi_1$ c'est-à-dire au moins une lettre commune, mais sous la forme opposée, avec chaque monôme premier.

En réduisant l'expression ainsi obtenue on obtient une base de φ_1 (idem pour φ_2).

Cette méthode sera d'autant plus intéressante que φ^* contiendra moins de monômes de la forme $m_X \cdot m_Y \cdot m_Z$ avec m_Y petit.

Exemple :

$$\varphi = \underbrace{abd + cd'}_{\varphi_1} + \underbrace{de' + cf + df + ce'}_{\varphi_2}$$

$$X = \{a, b\}$$

$$Y = \{c, d\}$$

$$Z = \{e, f\}$$

$$\varphi^* = cd + ad'e'f + bd'e'f$$

θ_i	η_i
c	c
d	d
x cd	cd x
a	f
x ad'	e'f x
b	d'f
x bd'	d'e'f

les monômes marqués sont les seuls qui ont une lettre commune avec chaque monôme de φ_1 ou de φ_2 on obtient ainsi :

$$\begin{cases} \varphi_1^* = cd + ad' + bd' \\ \varphi_2^* = cd + e'f \end{cases}$$

2.2. Décomposabilité d'une écriture minimale.

Théorème ([14]) :

Si les bornes inférieure et supérieure d'une fonction incomplète admettent des décompositions disjointes simples par rapport au même opérateur + ou x suivant la même partition des variables alors il existe une écriture minimale ayant une décomposition disjointe simple par rapport au même opérateur et suivant la même partition des variables.

Démonstration :

Supposons que :

$$\left. \begin{aligned} \underline{f}(X, Y) &= \varphi_1(X) \cdot \varphi_2(Y) \\ \bar{f}(X, Y) &= \psi_1(X) \cdot \psi_2(Y) \end{aligned} \right\} X \cap Y = \emptyset$$

Soit $E(X, Y)$ une écriture minimale de $[\underline{f}, \bar{f}]$ nous avons :

$$\varphi_1(X) \cdot \varphi_2(Y) \leq E(X, Y) \leq \psi_1(X) \cdot \psi_2(Y)$$

il existe une valeur X_0 de X telle que $\varphi_1(X_0) = 1$ ce qui entraîne

$$\varphi_2(Y) \leq E(X_0, Y) \leq \psi_2(Y) \text{ de même il existe } Y_0 \text{ avec :}$$

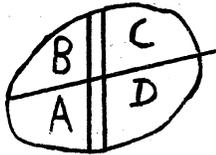
$$\varphi_1(X) \leq E(X, Y_0) \leq \psi_1(X)$$

donc $E(X, Y_0) \cdot E(X_0, Y)$ est compatible avec $[\underline{f}, \bar{f}]$ et a un coût \leq à celui de $E(X, Y)$.

Propriété :

\underline{f} et \bar{f}^* n'admettent pas simultanément de décomposition disjointe par rapport à l'addition. ($\bar{f} \neq 1$ et $\underline{f} \neq 0$).

En effet supposons que :



$$(2) \quad \underline{f} = f_1(A, B) + f_2(C, D) \leq g_1(A, C) \times g_2(B, D) = \bar{f}$$

il existe C_0, D_0 avec $f_2(C_0, D_0) = 1$

donc $g_1(A, C_0) \times g_2(B, D_0) = 1$ g_1 et g_2 sont in-

dépendants de A et B en rendant $f_1(A, B)$ égale à 1 on voit que (2) $\implies \bar{f} = 1$.

2.3. Une borne inférieure du coût minimum.

Proposition :

Si $\varphi(X)$ est une fonction de n variables : le coût minimum d'une expression de φ de signature $SP \dots$ est égal à n plus le nombre minimum de noeuds d'une fracture d'un réseau associé à une base première irrédondante que conque de φ .

Démonstration :

1°) Toute écriture de $\varphi(X)$ de signature $SP \dots$ est de la forme
(1) : $\varphi(XX') = \varphi_1(Y, T) + \varphi_2(T, Z)$ (Y, T, Z partition de $\{X, X'\}$) T est donc un ensemble de noeuds d'une fracture de R_φ puisqu'en dédoublant T on obtient :

$$\varphi_d(X, T_1) = \varphi_1(Y, T) + \varphi_2(T_1, Z) \quad R_{\varphi_d}$$

n'étant pas connexe.

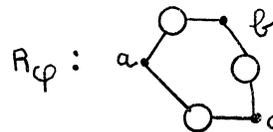
Réciproquement si T est une fracture de R_φ alors il existe au moins une décomposition de $\varphi(X)$ telle que (1).

2°) Le coût d'une écriture telle que (1) est supérieure ou égale à :

$$|Y| + |T| + |T| + |Z| \geq n + |T| \quad \text{avec } n = |X|$$

Exemple :

$$\varphi(X) = ab + bc + ca$$



Une fracture de R_φ a au moins 2 noeuds le coût minimum d'une expression de signature SP ... de φ est donc $2 + 3 = 5$ de même pour $\varphi^* = \varphi$ donc l'écriture $ab + c(b + a)$ est une écriture minimale de φ .

2.4. Fermeture indépendante.

$\varphi(XY)$ étant une fonction complète nous noterons par $\overline{\varphi(XY)}^X$ sa fermeture supérieure indépendante des variables X et par $\overline{\varphi(XY)}_X$ l'inférieure [cf. 8].

Nous avons :

$$\overline{\varphi(XY)}^X = \sum_{x_1 \dots x_p} \varphi(x_1 \dots x_p, Y) \quad \text{sommation et produit étendus aux } 2^p \text{ valeurs de } X.$$

$$\overline{\varphi(XY)}_X = \prod_{x_1 \dots x_p} \varphi(x_1 \dots x_p, Y)$$

φ étant donnée par sa base complète on obtient $\overline{\varphi}_X$ en supprimant les monômes contenant au moins une variable de X , $\overline{\varphi}^X$ peut être obtenue à partir d'une base quelconque en supprimant les variables de X dans les monômes où elles apparaissent.

Si $[\underline{f}, \overline{f}]$ est une fonction incomplète telle que :

$$\underline{f}(XY) \leq \overline{f}(YZ) \quad \text{avec } X \cap Y = Y \cap Z = X \cap Z = \emptyset$$

alors on a :

$$(3) \quad \underline{f}(XY) \leq \overline{\underline{f}(XY)}^X \leq \overline{\overline{f}(YZ)}_Z \leq \overline{f}(Y, Z)$$

En effet l'isotonie de la fermeture entraîne :

$$\underline{f}(XY) \leq \bar{f}(YZ) \implies \overline{\underline{f}}^X \leq \overline{\bar{f}}^X = \bar{f} \text{ puisque } \bar{f} \text{ est indépendante de } X$$

$$\overline{\underline{f}}^X \leq \bar{f} \implies \underbrace{\left(\overline{\underline{f}}^X\right)}_Z \leq \bar{f} \text{ mais } \overline{\underline{f}}^X \text{ est indépendante de } Z \text{ on a donc}$$

$$\overline{\underline{f}}^X \leq \underbrace{\bar{f}}_Z \text{ et l'inégalité (3).}$$

Si $[\underline{f}, \bar{f}]$ est une fonction incomplète croissante (ses deux bornes sont croissantes) avec $\underline{f}(XY)$, $\bar{f}(YZ)$ et $X \cap Y = Y \cap Z = X \cap Z = \emptyset$ alors toute écriture minimale croissante est compatible avec $\left[\overline{\underline{f}}^X, \underbrace{\bar{f}}_Y \right]$.

En effet : supposons qu'une écriture minimale de $[\underline{f}, \bar{f}]$ soit :

$$\psi(Y, X_1, Z_1) = E_{mn}(\underline{f}, \bar{f}) \begin{matrix} X_1 \subset X \\ Z_1 \subset Z \end{matrix}$$

nous avons :

$$\underline{f}(XY) \leq \psi(Y, X_1, Z_1) \leq \bar{f}(Y, Z)$$

$$\underline{f}(XY) \leq \psi(Y, 1, 0) \leq \bar{f}(YZ) \text{ puisque les fonctions sont croissantes}$$

$$\overline{\underline{f}}^X \leq \psi(Y, 1, 0) \leq \underbrace{\bar{f}}_Z \text{ d'après la définition d'une fermeture}$$

$$\text{d'autre part } c_{mn}(\psi(Y, 1, 0)) < c_{mn}(\psi(Y, X_1, Z_1)).$$

ce résultat est faux dans le cas de fonction non croissante :

$$\underline{f} = AB + BD + DC < AB + CD + x'D + xB + BD = \bar{f}$$

A, B, C, D étant des fonctions privilégiées sur des ensembles de variables disjoints 2 à 2 et ayant un coût minimum > 2 l'écriture minimale de

$$\left[\underline{f}, \bar{f} \right] \text{ est : } \psi = B(A + x) + D(C + x').$$

Remarque :

Si $F = \left[\underline{f}(X), \bar{f}(X) \right]$ est une fonction incomplète $Y \subset X$ est un ensemble de variables essentielles pour cette fonction s'il existe une fonction $\left[\underline{g}(Y), \bar{g}(Y) \right]$ compatible avec $F : \underline{f}(X) \prec \underline{g}(Y) \prec \bar{g}(Y) \prec \bar{f}(X)$ et si Y est minimale par inclusion pour cette propriété. L'écriture minimale de $\left[\underline{f}, \bar{f} \right]$ peut n'être pas compatible avec (g, \bar{g}) .

Exemple :

$\underline{f} = ab + bc + acd \prec g = \bar{g} = ab + bc + ac \prec \bar{f} = ab + bc + ac + ad + cd$
{a, b, c} est un ensemble de variables essentielles c $(ab + bc + ac) = 5$ alors que $(a + c)(b + d)$ est l'écriture minimale de $\left[\underline{f}, \bar{f} \right]_{mn}$.

2.5. Ecriture minimale d'une fonction incomplète dont l'une des bornes a une base première irrédondante d'au plus 2 monômes.

2.5.1. Cas de 1 monôme.

a) $\underline{f} \prec m = \bar{f}$

l'écriture minimale $E_{mn}(\underline{f}, \bar{f})$ est m . En effet les variables de m apparaîtront sous la même forme dans toute fonction $\prec m$ or elles n'apparaissent qu'une fois dans m .

b) $\underline{f} = m \leq \bar{f}$

$E_{mn}(\underline{f}, \bar{f})$ = un monôme de $\bar{f} \geq m$ ayant un nombre de lettres minimum (pas nécessairement unique). En effet supposons qu'une écriture minimale ait pour expression développée réduite : $(E_{mn}) = m_1 + m_2 + \dots + m_r > 1$ m est inférieur à un consensus d'une somme partielle μ ce consensus a un nombre de lettres strictement inférieur à celui de E_{mn} μ est donc une écriture de $[\underline{m}, \bar{f}]$ moins coûteuse que E_{mn} . (Le coût d'une expression étant supérieure ou égale au nombre de lettres distinctes obtenues lorsqu'on la développe).

2.5.2. Écriture minimale d'une fonction complète de 2 monômes.

Considérons une fonction complète f ayant pour base première irrédondante : $f = m_1 + m_2$ $m_1 = n n_1$ $m_2 = n n_2$ $n_1 \cap n_2 = \phi$

on a :

$$E_{mn}(m_1 + m_2) = n(n_1 + n_2) \quad (5)$$

En effet :

$$E_{mn}(m_1 + m_2) = \tilde{y} \dots \tilde{z} (\tilde{a} \dots \tilde{l}x + \tilde{m} \dots \tilde{n}x')$$

il peut y avoir des lettres communes entre $\tilde{a} \dots \tilde{l}$ et $\tilde{m} \dots \tilde{n}$ mais pas sous la même forme. f est dépendante de $\tilde{y} \dots \tilde{z}$, $\tilde{a} \dots \tilde{l}$, $\tilde{m} \dots \tilde{n}$ elle n'est ni croissante ni décroissante en x donc toute expression de f aura au moins autant de lettres que (5) qui est donc minimale.

2.5.3. Cas de 2 monômes.

a) $\underline{f} \leq m_1 + m_2 = \bar{f}$ base première irrédondante

- si m_1 (ou m_2) est $\succ \underline{f}$ $E_{mn}(\underline{f}, \bar{f}) = m_1$

- s'il n'y a pas de consensus entre m_1 et m_2 :

$$E_{mn}(\underline{f}, \bar{f}) = E_{mn}(m_1 + m_2)$$

- si m_3 est le consensus entre m_1 et m_2

$$E_{mn}(\underline{f}, \bar{f}) \begin{cases} = E_{mn}(m_1 + m_2) \text{ ou} \\ = m_3 \text{ si } \underline{f} \text{ est indépendante de la variable par} \end{cases}$$

rapport à laquelle il y a consensus.

Démonstration :

a) - le premier résultat découle de 3.1.1.

b) - s'il n'y a pas de consensus entre m_1 et m_2 , \underline{f} est dépendante de toutes les variables de $m_1 + m_2$ sinon l'un des monômes m_1 ou m_2 est supérieur à \underline{f} . En effet si \underline{f} est indépendante de $a \in m_2$ alors $\underline{f} = \underbrace{(\underline{f})}_a \prec \underbrace{m_1 + m_2}_a = m_1$ de même si $m_1 + m_2$ n'est pas monotone en x \underline{f} ne sera pas monotone en x (prendre les fermetures monotones) nous avons vu en 3.2. que $E_{mn}(m_1 + m_2) = n(n_1 + n_2)$

c) - s'il y a consensus entre m_1 et m_2 soit m_3 ce consensus :

$$m_1 = nn_1 \quad n_1 = px \quad m_3 = npq$$

$$m_2 = nn_2 \quad n_2 = qx'$$

si \underline{f} est indépendante de x alors $\underline{f} \prec pqn$ pqn est alors l'écriture minimale, (3.1) sinon on a le même résultat qu'en b).

$$b) \underline{f} = m_1 + m_2 \leq \bar{f}$$

$m_1 + m_2$ base première irrédondante de \underline{f} .

Notons $(M_1^i)_{i \in I}$ les monômes premiers de $\bar{f} \geq m_1$

$(M_2^j)_{j \in J}$ les monômes premiers de $\bar{f} \geq m_2$

$$(6) E_{mn}(\underline{f}, \bar{f}) = E_{mn}(M_1^h + M_2^k) \text{ avec}$$

$$\text{coût} \{E_{mn}(M_1^h + M_2^k)\} = \min_{\substack{i \in I \\ j \in J}} \left[\text{coût} \{E_{mn}(M_1^i + M_2^j)\} \right]$$

Démonstration :

$E_{mn}(\underline{f}, \bar{f}) = E_{mn}(n_1 + n_2)$ supposons que l'expression développée d'une écriture minimale soit $(E_{mn}) = \ell_1 + \ell_2 + \dots + \ell_r$ avec $r > 2$. Le coût de E_{mn} est supérieur au nombre total de lettres distinctes dans (E_{mn}) mais :

$$m_1 \leq \mu \text{ consensus d'une somme partielle de } (E_{mn})$$

$$m_2 \leq \nu \text{ idem}$$

le coût de $E_{mn}(\mu + \nu) < \text{coût de } E_{mn}$ puisque des lettres ont disparues et que chaque lettre n'apparaît qu'une fois dans $E_{mn}(\mu + \nu)$ il reste à montrer que n_1 et n_2 sont des monômes premiers de \bar{f}

$E_{mn}(n_1 + n_2) = p(q_1 + q_2)$ si pq_1 n'est pas un monôme premier de \bar{f} il existe $p'q'_1 > pq_1$ avec un nombre de lettres inférieures donc

$$\text{coût} \{E_{mn}(p'q'_1 + pq_2)\} \leq \text{coût} \{E_{mn}(pq_1 + pq_2)\}$$

C H A P I T R E IV

ALGORITHMES D'ECRITURE DES FONCTIONS BOOLEENNES EN S ET P

1 - FONCTION COMPLETE CROISSANTE

1.1. Algorithmes.

Si $\varphi(X)$ est une fonction croissante nous noterons R_φ le réseau de noeuds et d'étoiles représentant sa base. Les algorithmes décrits conduisent à une écriture croissante de la fonction.

1.1.1. Respect des composantes connexes.

Si $\varphi(X)$ admet une décomposition disjointe simple par rapport à la somme $\varphi(X) = F(Y) + g(Z)$ $Y \cap Z = \emptyset$ on traite indépendamment $F(Y)$ et $G(Z)$. De même si $\varphi^*(X)$ n'est pas connexe on traite indépendamment ses composantes ceci en vertu du théorème 2.2. Chapitre III

Il n'y a pas d'ambiguïté car $\varphi(X)$ ne peut pas admettre de décomposition disjointe simultanément par rapport au produit et à la somme.

1.1.2. Critère de choix. Convergence.

Si R_φ et R_{φ^*} sont connexes, nous décomposerons R_φ si le coût de l'écriture minimale en somme de produits de φ est inférieur à celui de φ^* :
 $c_m(SP\varphi) < c_m(SP\varphi^*)$.

Ce critère permet d'obtenir l'écriture minimale dans le cas où elle a 2 couches il permet également d'affirmer la convergence de l'algorithme puisque à chaque pas on diminue strictement soit le nombre de variables de la fonction traitée soit son coût minimum en SP.

1.1.3. Critères de décomposition.

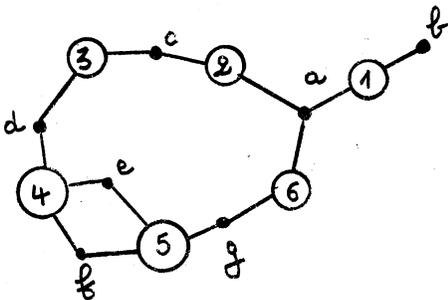
a) Décomposition "minimale" (conduit à l'algorithme 1).

Nous réalisons une décomposition "minimale" d'un réseau connexe R_φ de noeuds et d'étoiles si nous le disconnectons en dédoublant le moins de noeuds possibles. Nous avons résolu ce problème au chapitre 1 et nous disposons de programme nous permettant de décomposer un réseau connexe suivant une fracture ayant un nombre minimum de noeuds. S'il y a plusieurs fractures minimales nous choisirons une de celles qui partagent le mieux les étoiles c'est-à-dire correspondant à une hypercoupure simple ayant un nombre maximum d'étoiles dans son premier membre.

b) Décomposition "équilibrée" (conduit à l'algorithme 2).

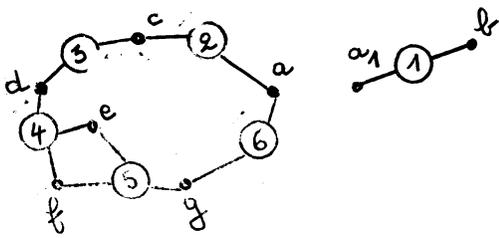
Nous inversons les critères précédents c'est-à-dire que parmi les hypercoupures simples ayant un nombre maximum d'étoiles dans leur premier membre nous choisirons une de celles qui ont une fracture associée ayant un nombre minimum de noeuds. Nous avons présenté au chapitre 1 des programmes résolvant ce problème.

Exemple :



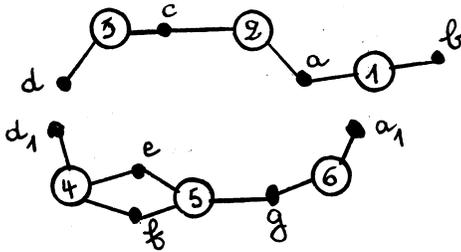
$$\varphi = ab + ac + cd + def + efg + ag$$

il y a une seule décomposition minimale qui conduit à :



$$\begin{cases} \varphi_1 = a_1 b \\ \varphi_2 = ac + cd + def + efg + ag \end{cases}$$

une des décompositions équilibrées est la suivante :



$$\begin{cases} \varphi_1 = ab + ac + cd \\ \varphi_2 = a_1 g + efg + d_1 ef \end{cases}$$

c) Décomposition "mixte" (conduit à l'algorithme 3).

On peut combiner les deux décompositions définies précédemment par exemple si la décomposition minimale est trop déséquilibrée on choisira la décomposition équilibrée.

Pour donner un exemple précis supposons que la décomposition minimale conduise à $R_\varphi = R_{\varphi_1} + R_{\varphi_2}$ le nombre d'étoiles de R_{φ_1} étant inférieur à la moitié de celui de R_{φ_2} nous choisirons alors la décomposition équilibrée en notant $m =$ nombre d'étoiles de R_φ .

Nous n'acceptons le découpage minimal que s'il conduit à $m_1 > \frac{m}{3}$. Le choix d'une telle décomposition pourra se faire à partir des résultats obtenus avec les décompositions a) et b).

1.1.4. Introduction de monômes redondants.

Les décompositions définies ci-dessus correspondent à une partition de la base de la fonction croissante et nous savons (voir ex. 1) qu'une écriture minimale n'est pas toujours obtenue à partir de la base.

Considérons une décomposition :

$$\varphi(XYZ) = \varphi_1(XY) + \varphi_2(YZ)$$

X, Y, Z étant une partition de l'ensemble des variables

φ_1 et φ_2 réalisant une partition de la base de φ

φ , φ^* , (φ_1^* ou φ_2^*) étant connexes.

Une écriture minimale de la fonction incomplète $[\varphi_1, \varphi]$ a un coût inférieur ou égal à celui d'une écriture minimale de la fonction complète φ_1 . Une telle fonction s'écrit :

$$(1) \quad \begin{cases} \psi_1(XYZ) = \varphi_1(XY) + f(XYZ) \\ \psi_1 \leq \varphi \\ c_m(\psi_1) \leq c_m(\varphi_1) \end{cases}$$

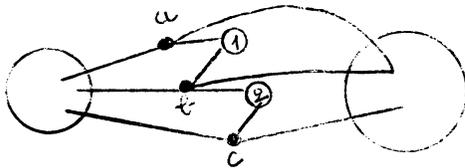
Toutes les fonctions étant supposées croissantes il existe une écriture minimale de $[\varphi_1, \varphi]$ ne dépendant pas des variables Z. En effet supposons que $\psi_1(XYZ)$ soit une telle fonction, $\psi_1(XY0)$ satisfait aux conditions (1) elle a de plus un coût inférieur à celui de $\psi_1(XYZ)$. Nous ne savons pas trouver les fonctions $\psi_1(XY)$ satisfaisant (1) nous nous limiterons à la recherche des monômes m_{XY} tels que :

$$(1') \left\{ \begin{array}{l} \varphi_1(XY) + m_{XY} \leq \varphi(XYZ) \\ (\varphi_1(XY) + m_{XY})^* \text{ admet une décomposition disjointe simple par rapport} \\ \text{au produit.} \end{array} \right.$$

(1') permet de conserver la convergence de l'algorithme puisque nous diminuons strictement le nombre de variables des fonctions à étudier.

D'autre part un monôme m_{XY} satisfaisant (1') est $\leq \varphi_2(YZ)$ donc il existe un monôme premier m_Y de φ_2 ne contenant que des variables de Y avec $m_{XY} \leq m_Y$. Si nous utilisons la décomposition minimale il existe au plus un tel monôme m_Y en effet supposons qu'il existe 2 monômes premiers sur Y , ils ont chacun un nombre de lettres $< Y$ donc Y n'est pas une fracture minimale.

Exemple :



$Y = \{a, b, c\}$ mais en dédoublant seulement a et b on disconnecte le réseau donc Y n'est pas minimal.

C'est le seul cas que nous envisagerons. On utilise l'algorithme suivant :

- ① S'il existe un monôme premier m de $\varphi_2(YZ)$ composé uniquement de variables de Y aller en 2 sinon aller en 5.
- ② Si φ_1^* n'est pas connexe aller en 6 sinon aller en 3.
- ③ Regarder si l'un des monômes θm avec $\theta = 1, x_1, x_2, \dots, x_p$ $\{x_1, x_2, \dots, x_p\} = X$ conduit à $(\varphi_1 + \theta m)^*$ non connexe. Si $\theta = 1$ convient et si φ_2^* est connexe aller en 4 sinon en 6.

④ On refait 3 en échangeant φ_1 et φ_2 , si on ne trouve aucun θ qui convienne on supprime m de φ_2 et on va en 6.

⑤ Refaire 2, 3, 4 en échangeant φ_1 et φ_2 s'il existe un monôme premier m de φ_1 sur Y , sinon aller en 6.

⑥ Continuer.

Exemple 1 :

En utilisant le critère de décomposition minimale nous obtenons une écriture minimale de :

$$\varphi = \underbrace{af + ed + ef}_{\varphi_1} + \underbrace{ad + ab + bc + cdg}_{\varphi_2}$$

c'est le cas le plus compliqué envisagé par l'algorithme, on remplace

$$\left\{ \begin{array}{l} \varphi_1 \text{ par } \varphi_1 + ad \\ \varphi_2 \text{ par } \varphi_2 + adg - ad \end{array} \right.$$

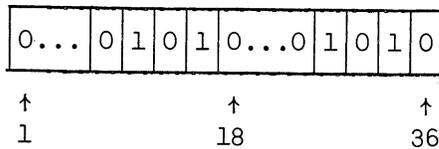
on obtient ainsi $\varphi = (a + d)(e + f) + (a + c)(b + dg)$ qui a bien un coût minimum en vertu de 2.3. du chapitre 3 car une fracture minimale de R_φ a 2 éléments et une de R_{φ^*} en a 3 :

$$\varphi^* = \underset{1}{acd} + \underset{2}{abd} + \underset{3}{acef} + \underset{4}{bdef} + \underset{5}{bgef}$$

1.2. Programmes.

Nous avons écrit une procédure en ALGOL^{de} 4.000 unités syntaxiques utilisant, en plus des procédures usuelles en CODE et des procédures ALGOL de traitement des réseaux de noeuds et d'étoiles déjà décrites, des procédures ALGOL de traitement des fonctions booléennes. Le programme complet est d'environ 9.000 unités syntaxiques, son temps d'assemblage et de compilation est de l'ordre de 3 minutes. (IBM 7.044).

La représentation d'une somme de monômes booléens en machine est la même que celle d'un réseau de noeuds et d'étoiles (voir chapitre I 3.2.2.), mais les 18 derniers bits de la mémoire représentent les compléments des variables, c'est-à-dire que s'il y a 4 variables la mémoire suivante



représente le monôme $a'bc'd$, nous nous limiterons

donc aux fonctions d'au plus 18 variables et 100 monômes.

On trouvera la liste des procédures de calcul booléen, avec des commentaires, dans le programme complet présenté en annexe.

Le choix de la décomposition utilisée est une option de la procédure SOP0 qui donne sous forme préfixée l'écriture de la fonction traitée.

procedure SOP0(F,M,N,OPT,R,T) ; valeur F,M,N,OPT ;

entier M,N,OPT,T ; entier tableau F,R ;

commentaire : cette procédure range dans le tableau R entre les indices 1 et T l'écriture en notation préfixée de la fonction (F,M,N) croissante de M monômes et N variables obtenue par l'algorithme i si OPT = i ;

1.3. Résultats.

Nous avons traité 5 fonctions de 6, 7, 8 variables que nous avons obtenues en tirant un certain nombre de monômes "au hasard". Un monôme croissant sur un ensemble de n variables étant représenté par un nombre entier compris entre 1 et $2^n - 1$ nous avons tiré au hasard un certain nombre de tels entiers

Pour une question de commodité nous avons écrit φ (resp. φ^*) dans les tableaux si le coût SP de φ (resp. φ^*) est inférieur à celui de φ^* (resp. φ).

On remarque que l'algorithme 3 constitue dans certains cas une amélioration des algorithmes 1 et 2 il sera donc intéressant de combiner les 2 critères de découpage suivant la nature de la fonction à traiter.

φ ou φ^*	coût SP de $\varphi = c_1$	coût SP de $\varphi^* = c_2$	coût mn (Supposé) c mn	coût obtenu avec Alg 1 co_1	co_2	co_3	en secondes		
							t_1	t_2	t_3
ab + bcef + bcd + bdef + acdf	17	23	12	13	12	12	3.7	1.9	1.9
acd + bcf + ace + cdf + abef + bce	23	19	12	14	13	13	4.8	2.2	2.2
aef + acde + be + cdf + acf + abf	18	26	13	14	13	13	8.5	2.0	2.1
abc + abd + bcd + bf + acd + ae + acf	19	19	11	13	13	11	8.7	4.4	4.6
bcd + abe + ace + cef + bde + acd + adef	25	22	12	17	14	13	10.4	6.1	6.0

écart moyen relatif de co_1 /coût minimum = 18.3 %
 écart moyen relatif de co_2 /coût minimum = 8.3 %
 écart moyen relatif de co_3 /coût minimum = 3.3 %

écart moyen relatif de co_1 /coût SP = 32.7 %
 écart moyen relatif de co_2 /coût SP = 38.4 %
 écart moyen relatif de co_3 /coût SP = 41.4 %

temps moyen 1 = 7.2 sec.
 temps moyen 2 = 3.3 sec.
 temps moyen 3 = 3.4 sec.

φ ou φ^*	c_1	c_2	co_1	co_2	co_3	en secondes		
						t_1	t_2	t_3
$ab + ac + cdg + dfg + bc + ef + ce$	16	30	10	10	10	3.8	3.7	3.7
$ag + acde + abd + bdef + abf + cg$	36	18	14	14	13	4.0	3.7	3.8
$bcd + cde + afg + adef + bdefg$	19	28	14	13	13	3.5	2.6	2.8
$acf + cdg + abeg + bcd + abdg + acd$	23	20	13	13	13	6.0	4.2	4.3
$df + aeg + cfg + acf + ef + abc + bf + dg$	29	20	16	14	16	20.0	14.1	16.8

Fonctions de 7 variables

$\left. \begin{array}{l} \text{écart moyen relatif de } co_1/\text{coût SP} = 43.9 \% \\ \text{écart moyen relatif de } co_2/\text{coût SP} = 46.4 \% \\ \text{écart moyen relatif de } co_3/\text{coût SP} = 45.6 \% \end{array} \right\}$	$\left. \begin{array}{l} \text{temps moyen 1} = 7.5 \text{ sec.} \\ \text{temps moyen 2} = 5.7 \text{ sec.} \\ \text{temps moyen 3} = 6.3 \text{ sec.} \end{array} \right\}$
---	--

φ ou φ^*	c_1	c_2	co_1	co_2	co_3	en secondes		
						t_1	t_2	t_3
cdfh + acdeg + adeh + be + cfgh + bf + ef	26	23	16	16	16	9.8	9.8	9.9
cdg + acdf + adef + bcd + abcef + bcegh + fh	39	26	21	17	18	21.1	8.6	8.9
aefg + bcfg + dh + ab + egh + ad + cdfg	21	34	16	14	14	13.5	6.3	6.5
abcfgh + bcdh + egh + aefg + bceg + cdeh + acde + beh	32	43	25	21	21	45.4	23.1	23.5
bf + bce + gh + dg + ch + adef + cef + cde	43	21	16	15	16	10.7	9.8	10.0

Fonctions de 8 variables

écart moyen relatif de co_1 /coût SP = 38.9 % écart moyen relatif de co_2 /coût SP = 46.1 % écart moyen relatif de co_3 /coût SP = 44.8 %	}	temps moyen 1 = 20.1 sec.
		temps moyen 2 = 11.5 sec.
		temps moyen 3 = 11.7 sec.

2 - FONCTION COMPLETE QUELCONQUE

2.1. Algorithmes.

2.1.1. Choix de la base de départ.

Dans le cas d'une fonction booléenne croissante, on fractionnait la base première unique, ce fractionnement étant éventuellement corrigé par l'introduction de monômes redondants. La recherche d'une base première de coût SP minimum étant trop longue dans le cas général, nous fractionnerons la base première irrédondante obtenue par une méthode de consensus (chapitre 4 [16]) [Th. TISON] puis nous corrigerons ce fractionnement par l'introduction de monômes redondants.

Nous savons qu'une écriture minimale n'est pas nécessairement obtenue à partir d'une base première irrédondante, (voir exemple ci-dessous), cependant si l'introduction de monômes était "rigoureuse", c'est-à-dire conduisait à une écriture minimale des fonctions incomplètes $[f_1, f]$ et $[f_2, f]$ si $f = f_1 + f_2$, on pourrait obtenir une écriture minimale de f en considérant toutes les décompositions d'une base quelconque de f (ou de f^*).

Exemple :

La fonction $\varphi(X) = ae + af + bdu + cdu + be + ce + bf + cf + ax + dx'$
a pour écriture minimale :

$$E_{mn}(\varphi) = (a + b + c) (du + e + f) + ax + dx'$$

obtenue en ajoutant adu , monôme non premier, à la base première irrédondante de $\varphi(X)$. (Les algorithmes permettent de l'obtenir).

Le choix d'une base première irrédondante minimise le nombre de lettres distinctes utilisées (voir chapitre 3) et permet d'obtenir facilement les décompositions disjointes par rapport à la somme.

2.1.2. Respect des composantes connexes.

Si la fonction φ (resp. φ^*) admet une décomposition disjointe simple par rapport à la somme, on traite indépendamment ses composantes connexes d'après le théorème 2.2. du chapitre III, on obtient ces décompositions à partir de toute base première irrédondante de φ (resp. φ^*) (voir [12]).

Si φ et φ^* sont connexes, on considère la base première irrédondante de φ , obtenue par la méthode de 2.1.1., si elle a un coût inférieur à celle de φ^* (par exemple).

Dans ce cas si φ est non c-connexe on traite indépendamment ses composantes c-connexes sinon on la décompose suivant l'un des critères ci-dessous.

2.1.3. Critères de décomposition.

Etant donné que l'on décompose un réseau connexe, les critères de décomposition utilisés sont les mêmes que pour les fonctions croissantes (voir 1.1.3. chapitre IV).

a) décomposition "minimale". (conduit à l'algorithme 1).

b) décomposition "équilibrée". (conduit à l'algorithme 2).

2.1.4. Introduction de monômes redondants.

Lorsque φ n'est pas c-connexe ou après un fractionnement nous avons :

$$\varphi(XYZ) = \varphi_1(\tilde{x}_1 \tilde{y}_2 \dots \tilde{y}_k) + \varphi_2(\tilde{y}_1 \tilde{y}_2 \dots \tilde{y}_k Z)$$

X,Y,Z réalisant une partition des variables effectives de φ . S'il existe une fonction ψ telle que :

$$\left\{ \begin{array}{l} \varphi_1 \leq \psi \leq \varphi \quad (1) \\ c_{mn}(\psi) \leq c_{mn}(\varphi_1) \quad (2) \end{array} \right.$$

l'écriture de φ obtenue en traitant ψ sera meilleure que celle obtenue en traitant φ_1 . La condition (2) n'étant pas vérifiable nous pouvons la remplacer par :

$$(3) \quad \left\{ \begin{array}{l} \psi^* = (\varphi_1 + f)^* \text{ non connexe} \\ c_{mn}(f) < d_{mn} = \text{coût de la décomposition minimale de } R_{\varphi_1} \text{ ou } R_{\varphi_1^*} \end{array} \right.$$

En remplaçant $c_{mn}(f)$ par une valeur particulière, par exemple $c_{mn}(SPf)$, les conditions (3) sont théoriquement vérifiables et conduisent à une amélioration de l'écriture obtenue, mais la recherche des fonctions ψ compatible avec $[\varphi_1, \varphi]$, décomposables disjointement par rapport au produit est longue (voir [13]). Le critère que nous utiliserons sera le suivant :

(4) nous remplacerons φ_1 par $\varphi_1 + m_{XY}$, m_{XY} étant un monôme sur les variables (X, Y) si :

- $m_{XY} \leq \varphi$, $m_{XY} \notin \varphi_1$,

- $(\varphi_1 + m_{XY})^*$ n'est pas connexe

- le nombre de lettres de m_{XY} est $<$ au minimum du nombre de lettres d'un monôme de φ_1 ou de φ_1^* (sinon le coût de m_{XY} est $> d_{mn}$).

Il est en effet inutile de considérer des monômes $\leq \varphi_1$ car dans ce cas $\varphi_1 + m_{XY} = \varphi_1$.

De même si $m = nz$ $z \in Z$ et si $(\varphi_1 + nz)^*$ est décomposable disjointement par rapport à la somme il en est de même de φ_1 en effet :

$$\begin{aligned}
 (\varphi_1 + nz)^* &= (z + \tilde{x}_1 + \tilde{x}_2 + \dots + y_1 + \dots) \varphi_1^*(XY) \\
 &= \psi_1(X_1 Y_1) + \psi_2(X_2 Y_2 z)
 \end{aligned}$$

donc $(\varphi_1)^* = \psi_1(X_1 Y_1) + \psi_2(X_2 Y_2 0)$ est décomposable car ψ_2 dépend effectivement d'une variable $\neq z$ sinon $z \varphi_1^*(XY) = \tilde{z}$ impossible.

Remarque :

Dans le cas croissant nous avons démontré que si une fonction $\psi(XYZ)$ satisfaisait à (1) et à (2) on pouvait trouver $\psi_1(XY)$ satisfaisant à (1) et à (2) avec $c_{mn}(\psi_1) < c_{mn}(\psi)$ dans le cas de fonctions non croissantes ce résultat n'est plus valable :

$$\varphi_1 = AB + BD + DC \leq \psi \leq \varphi = AB + CD + x'D + xB + BD$$

A, B, C, D étant des fonctions privilégiées sur des ensembles de variables disjoints 2 à 2 et ayant un coût minimum > 2 , l'écriture minimale de

$[\varphi_1, \varphi]$ est :

$$\psi = B(A + x) + D(C + x')$$

une écriture de $[\varphi_1, \varphi]$ ne contenant pas x est une écriture de $\varphi_1 = AB + BD + DC$ or dans toute écriture de φ_1 une des variables au moins est répétée 2 fois, comme chaque variable a un coût > 2 ce ne sont pas des écritures minimales de $[\varphi_1, \varphi]$.

2.1.5. Convergence des algorithmes.

A chaque pas de l'algorithme on diminue strictement :

- le coût SP de la fonction traitée ou
- son nombre de variables ou
- le nombre de variables qui apparaissent sous les 2 formes.

L'une de ces 3 éventualités se réalisant nécessairement les algorithmes convergent.

2.2. Programmes.

La procédure SOP écrite en ALGOL est de 1.592 unités syntaxiques elle est récursive, ce gain en simplicité étant naturellement contrebalancé par un manque de souplesse dans l'utilisation des tableaux.

Comme nous utilisons de nouvelles procédures de calcul booléen, recherche des monômes premiers d'une fonction, des bases premières etc... (qui se trouvent dans la bibliothèque de l'Equipe de Logique), le programme complet est, comme dans le cas des fonctions croissantes, de 9.000 unités syntaxiques son temps d'assemblage et de compilation est de 3 minutes 40 secondes.

Le choix de la décomposition utilisée est une option de la procédure SOP qui donne sous forme préfixée l'écriture de la fonction traitée.

procedure SOP(F,M,N,OPT,R) ; valeur F,M,N,OPT ;

entier M,N,OPT ; entier tableau F,R ;

commentaire : Cette procédure range dans le tableau R entre les indices 1 et T l'écriture en notation préfixée de la fonction complète (F,M,N) de M monômes et N variables obtenue par l'algorithme i si OPT = i. Cette procédure étant réursive un certain nombre d'identificateurs sont déclarés à l'extérieur notamment T qui doit être initialisé à 0 avant le premier appel de la procédure ; les limites d'utilisation de cette procédure sont identiques à celles de SOP0 (voir 1.2.).

2.3. Résultats

Nous avons traité 5 fonctions de 5,6 variables obtenues comme dans le cas de fonctions croissantes. Pour une question de commodité nous avons écrit φ (resp. φ^*) dans les tableaux si le coût SP de φ (resp. φ^*) est inférieur à celui de φ^* (resp. φ).

	φ ou φ^*	c_{mn}	c_1	c_2	co_1	co_2	en secondes	
							t_1	t_2
Fonctions de 5 variables	$ce' + a'e' + a'bcd + be' + ab'd + ab'c$	≥ 9	16	18	13	12	9.3	4.4
	$a'bd + a'bc + abd' + b'c'$	8	11	9	8	8	1.2	1.2
	$a'b'd' + bc' + cde' + abe'$	10	11	16	10	10	2.3	2.1
	$c'e + c'd + ab'e' + a'bd' + b'de' + a'b'e$	11	16	14	11	11	2.8	2.5
	$de + b'cd' + acd' + a'b'd'e'$	8	12	11	8	8	1.6	1.7
Fonctions de 6 variables	$a'b'd' + ab'df + b'ef' + ace + a'def' + abc'$	≥ 11	20	26	16	15	7.1	6.3
	$a'cdf' + a'bce'f + a'b'ce + ab'ce'$	≥ 12	17	19	12	12	3.9	3.7
	$b'c'ef' + abd'e' + abc'd'f' + a'def'$	≥ 13	24	17	12	14	3.7	4.5
	$b'd'e'f' + ab'df + abd'ef' + a'b'cd'$	≥ 11	17	18	14	12	4.7	3.5
	$a'b'd' + bce + ce'f' + a'c'e' + ace$	≥ 11	20	15	12	12	3.9	3.8

fonctions de 5 variables.

écart moyen relatif de co_1 /coût minimum $\leq 9 \%$

écart moyen relatif de co_2 /coût minimum $\leq 6 \%$

écart moyen relatif de co_1 /coût SP = 25.4. %

écart moyen relatif de co_2 /coût SP = 27 %

temps moyen 1 = 3.4 sec.

temps moyen 2 = 2.4 sec.

fonctions de 6 variables.

écart moyen relatif de $co_1/c_{mn} \leq 14 \%$

écart moyen relatif de $co_2/c_{mn} \leq 12 \%$

écart moyen relatif de $co_1/\text{coût SP} = 32 \%$

écart moyen relatif de $co_2/\text{coût SP} = 33 \%$

temps moyen 1 = 4.7 sec.

temps moyen 2 = 4.4 sec.

3 - FONCTION INCOMPLETE

3.1. Algorithmes.

Les deux bornes \underline{f} et \bar{f} sont données sous la forme d'une somme de monômes et nous disposons d'algorithmes permettant de calculer la base complète et une base première irrédondante d'une fonction booléenne.

3.1.1.

Calculer la base complète de \bar{f} et une base première irrédondante de \underline{f} .

3.1.2.

Regarder s'il existe une variable x_i telle que $\underline{f} < \tilde{x}_i < \bar{f}$ \tilde{x}_i est alors l'écriture choisie de $[\underline{f}, \bar{f}]$.

3.1.3.

Supprimer les monômes inutiles : c'est-à-dire les monômes premiers de \bar{f} tels que leur intersection avec \underline{f} soit vide.

3.1.4.

Si $\underline{f} = \underline{f}(XY)$ et $\bar{f} = \bar{f}(YZ)$ X, Y, Z réalisant une partition de l'ensemble des variables alors on remplace \underline{f} par sa fermeture supérieure indépendante de X $\overline{\underline{f}}^X$ et \bar{f} par $\overline{\bar{f}}^Z$ (voir [7]). Nous avons vu en 2.4. Chapitre III que le passage à la fermeture inférieure se faisait sur la base complète alors que le calcul de la fermeture supérieure peut se faire à partir de n'importe quelle base.

Remarque :

Il est inutile de recommencer 3.1.3. après 3.1.4. En effet si $n \in \overline{\bar{f}}^Z$ est un monôme tel que $n \overline{\underline{f}}^X = 0$ alors $n \cdot \underline{f} = 0$ puisque :
 $\underline{f} < \overline{\underline{f}}^X$ et si n est élément de la base complète de $\overline{\bar{f}}^Z$ il est également monôme de la base complète de \bar{f} donc monôme inutile de $[\underline{f}, \bar{f}]$.

On recommence 3.1.4. jusqu'à ce que les 2 bornes aient le même ensemble de variables effectives.

Exemple :

$$\underline{f} = ab + ad + bc + cd < a + c + bde = \bar{f}$$

$$X = \emptyset \quad Y = \{a, b, c, d\} \quad Z = e$$

$$\underline{f}^X = ab + bc + ad + cd < a + c = \bar{f}^Z$$

$$X_1 = b \quad Y_1 = \{a, c\} \quad Z_1 = \emptyset$$

$$\overline{\underline{f}}^{X_1} = a + c < a + c = \overline{\bar{f}}^{Z_1}$$

3.1.5.

Si le nombre d'éléments d'une base première irrédondante de \underline{f} ou de \bar{f} est ≤ 2 on connaît alors une écriture minimale de $[\underline{f}, \bar{f}]$ (voir chapitre III).

3.1.6.

Si \underline{f} et \bar{f} admettent une décomposition disjointe simple par rapport au même opérateur (+ ou \times) et suivant la même partition des variables on traite indépendamment les sous-fonctions incomplètes obtenues en prenant les composantes qui se correspondent en vertu du théorème

Par exemple si l'on a :

$$\underline{f} = \varphi_1(x_1) + \varphi_2(x_2) \leq \bar{\varphi}_1(x_1) + \bar{\varphi}_2(x_2) = \bar{f}$$

on traite indépendamment $[\varphi_1, \bar{\varphi}_1]$ et $[\varphi_2, \bar{\varphi}_2]$. Si ψ_1 et ψ_2 sont des écritures de ces fonctions $\psi_1 + \psi_2$ est l'écriture obtenue de $[\underline{f}, \bar{f}]$.

3.1.7.

Si \underline{f} admet une décomposition disjointe simple par rapport à l'addition $\underline{f} = \sum_i \varphi_i(x_i)$ on traite indépendamment les fonctions $[\varphi_i, \bar{f}]$, une solution de $[\underline{f}, \bar{f}]$ étant obtenue en faisant la somme des solutions obtenues pour les $[\varphi_i, \bar{f}]$. De même si \bar{f} est décomposable disjointement par rapport au produit $\bar{f} = \prod_i \varphi_i(x_i)$ on traite indépendamment les fonctions $[\underline{f}, \varphi_i]$ en faisant le produit des solutions obtenues on obtient une écriture de $[\underline{f}, \bar{f}]$. Nous avons vu au chapitre III qu'il n'y avait pas d'ambiguïté puisque ces 2 décompositions ne peuvent exister simultanément.

3.1.8.

On recommence 3.1.6. en considérant x et x' comme des variables indépendantes, dans ce cas \underline{f} et \bar{f} peuvent être simultanément décomposables on décomposera suivant celle qui a le plus de monômes.

3.1.9.

Si \underline{f} et \bar{f}^* sont c -connexes on fractionne \underline{f} si son réseau associé à une fracture minimale dont le nombre de noeuds est inférieur à celui d'une fracture minimale du réseau associé à \bar{f}^* . Si ces 2 nombres sont égaux on fractionne la borne qui a le plus de monômes.

Exemples :

$$\textcircled{1} \quad \underline{f} = abc + abd + afg + bfg + fgh < ab + gx + fx' + fg + hi + af = \bar{f}$$

$$X = \emptyset \quad Y = \{a, b, c, d, f, g, h\} \quad Z = \{i, x\}$$

$$\overline{\underline{f}}^X = \underline{f} < ab + fg + af = \bar{f} = \bar{g}$$

mais alors

$$X_1 = \{h, c, d\} \quad Y = \{a, b, f, g\} \quad Z = \emptyset$$

$$\overline{\underline{g}}^{X_1} = \underline{g} = ab + fg < ab + fg + af = \bar{g}$$

on obtient ainsi $ab + fg$ comme écriture qui est minimale.

② $\underline{f} = x'y + xy' + xt' + x'tz = \bar{f}$ $\bar{f}^* = xyt + xyz + x'y't'$ 3 et 4
sont inutiles puisque c'est une fonction complète \underline{f} et \bar{f}^* sont connexes mais non
c-connexes on fractionne donc celle qui a le plus de monômes :

$$\underline{f} = f_1 + f_2 \qquad f_1 = x'y + x'tz \qquad f_2 = xy' + xt'$$

on doit maintenant traiter indépendamment les 2 fonctions incomplètes $\left[f_1, \bar{f} \right]$
et $\left[f_2, \bar{f} \right]$ on arrive à l'écriture :

$$\underline{f} = \bar{f} = x'(y + tz) + x(y' + t')$$
 qui est minimale.

3.2. Programmes.

La procédure INSOP écrite en ALGOL est de 2.900 unités syntaxiques.
Elle est récursive et utilise toutes les procédures utilisées par SOP (voir 2.2.)
Le programme complet est de 10.000 unités syntaxiques, son temps d'assemblage et
de compilation est de 4 minutes, il est en annexe.

Le choix de la décomposition utilisée est une option de la procédure
INSOP qui donne sous forme préfixée l'écriture de la fonction traitée.

procedure INSOP(IN,MI,DSU,PS,OPTI) ; valeur IN,DSU,MI,PS,OPTI ;
entier MI,PS ; entier tableau IN,DSU ; booleen OPTI ;
commentaire cette procédure range dans R(1 : T) l'écriture en notation préfixée
de la fonction incomplète ayant IN pour borne inférieure et DSU pour duale de la
borne supérieure (MI et PS étant les nombres de monômes respectifs), obtenue par
l'algorithme 1 si OPTI est vrai par l'algorithme 2 autrement.

Cette procédure étant récursive un certain nombre d'identificateurs
sont déclarés à l'extérieur, notamment T qui doit être initialisé à 0 avant le
premier appel de la procédure ;

les limites d'utilisation de cette procédure sont identiques à celles de SOPO (voir 1.2.).

3.3. Résultats.

Nous avons traité 5 fonctions de 5 et 6 variables obtenues comme en 1.3. et en enlevant un certain nombre de monômes de la duale.

Dans les tableaux suivants c est le coût de la base première irrédondante (obtenue par BASEPREM1) de \underline{f} (resp. \bar{f}^*) s'il est inférieur à celui de \bar{f}^* (resp. \underline{f}).

	\underline{f} et \underline{f}^*	c	co_1	co_2	t_1	t_2
fonctions de 5 variables	$\underline{f}=b'cd'+ad'e'+a'bc'e+cde$ $\underline{f}^*=d'e+abc+ab'e+a'ce'+b'c'de'$	13	11	11	2.9	2.8
	$\underline{f}=a'c+a'e'+a'bd+ab'de+bcd$ $\underline{f}^*=cde'+abce'+a'd+a'be+a'b'c$	14	12	11	6.9	4.6
	$\underline{f}=bc'd'+b'de'+ac'd'e'+a'b'de'+a'c'd$ $\underline{f}^*=a'be'+a'b'd+b'c'+c'e'+a'c'd+abcd$	17	10	10	3.6	3.4
	$\underline{f}=a'c'd'+abde'+a'c'e'$ $\underline{f}^*=a'b+a'd+ac'+d'e'$	8	7	7	2.0	2.0
	$\underline{f}=a'cd'+cd'e'+a'b'd'e'+a'bc'de'$ $\underline{f}^*=ce'+bd'+c'd'+a'e'+b'cd$	11	10	10	3.6	3.6
fonctions de 6 variables	$\underline{f}=a'b'd'+ab'df+b'ef'+ace+a'def'+ab'c$ $\underline{f}^*=a'cdf'+ad'e'+ad'f'+b'e'+b'cd+a'b'c$	18	14	14	18.0	14.5
	$\underline{f}=a'cdf'+a'bce'f+a'b'ce+ab'ce'$ $\underline{f}^*=b'e'f'+b'df+abef'+abde+a'b'$	16	12	11	8.8	4.2
	$\underline{f}=b'd'e'f'+ab'df+abd'ef'+a'b'c'd'$ $\underline{f}^*=b'd'+a'bde'+bce'f+a'be'f+cdf'$	17	9	10	9.8	7.3
	$\underline{f}=b'c'ef'+abd'e'+abc'd'f'+a'def'$ $\underline{f}^*=a'b'd'+ab'd+c'de'+a'c'd'+bc'd$	15	9	9	4.4	4.3
	$\underline{f}=a'b'd'+bce+ce'f'+a'c'e'+ace$ $\underline{f}^*=a'c+b'ce'+cd'e'+b'c'ef'+a'ef'+c'd'ef'$	15	9	9	9.6	9.2

4 - REMARQUES AU SUJET DE LA SYNTHÈSE EN OPÉRATEURS NI.

4.1. Remarques au sujet des algorithmes.

Les algorithmes programmés cherchent à minimiser le nombre total de lettres dans une écriture en S et P d'une fonction booléenne. Ils n'essayent pas de minimiser le nombre d'opérateurs d'un réseau en S et P réalisant la fonction sauf dans le cas où les opérateurs ont 2 entrées et le réseau arborescent puisqu'alors on peut associer à toute expression un réseau convenant avec :

nombre total de lettres - 1 = nombre d'opérateurs.

En effet : les propriétés de décomposabilité utilisées (si $\psi(X) = f(y) + g(Z)$ $Y \cap Z = \emptyset$ on obtient une écriture minimale de ψ à partir d'une de f et d'une de g), les résultats sur l'écriture minimale d'une fonction incomplète ayant une borne dont la base irrédondante a au plus 2 monômes, ne sont valables que lorsque le coût est mesuré par le nombre de lettres. Notamment si une fonction est privilégiée par rapport à la somme et au produit les algorithmes permettent d'obtenir l'écriture minimum unique mais qui n'est pas nécessairement intéressante si le coût est mesuré par le nombre d'opérateurs (à plus de 2 entrées) :

Exemple

$$F = (ab(cd+ef)+gh)i$$

le réseau correspondant à l'écriture privilégiée a 7 opérateurs celui correspondant à l'écriture SP en a seulement 4. Le tableau I donne le nombre d'opérateurs du réseau correspondant à l'écriture minimale 2 couches et du réseau obtenu par les algorithmes, ce dernier est toujours supérieur au premier.

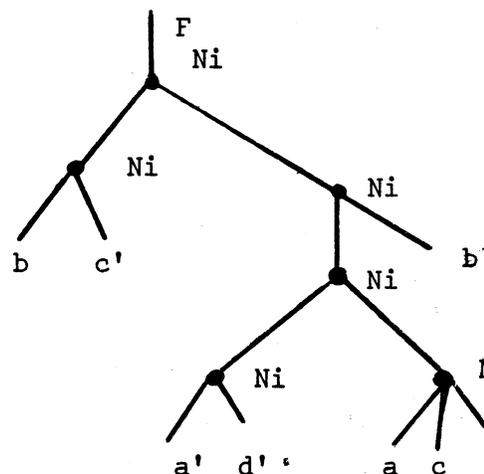
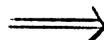
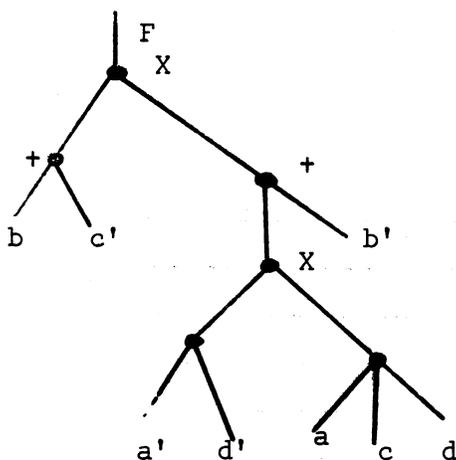
Les critères de décomposition utilisés qui cherchent à minimiser le nombre de variables dédoublées n'ont également de sens que si le coût est mesuré par le nombre de lettres.

4.2. Passage d'un réseau S et P à un réseau NI.

4.2.1. Si le réseau obtenu par les algorithmes correspondant à une expression de signature... SPS P (c'est-à-dire si 2 opérateurs successifs sont différents le 1^{er} étant un P) nous savons facilement déduire un réseau de NI en remplaçant chaque opérateur par un NI en prenant les lettres figurant dans une somme sous leur forme actuelle, celles figurant dans un produit sous leur forme opposée. Si l'expression est de signature... SPS il suffit de rajouter un opérateur NI à une entrée en aval du 1^{er} opérateur.

Exemple :

$$F = a'bd + a'bc + abd' + b'c'$$



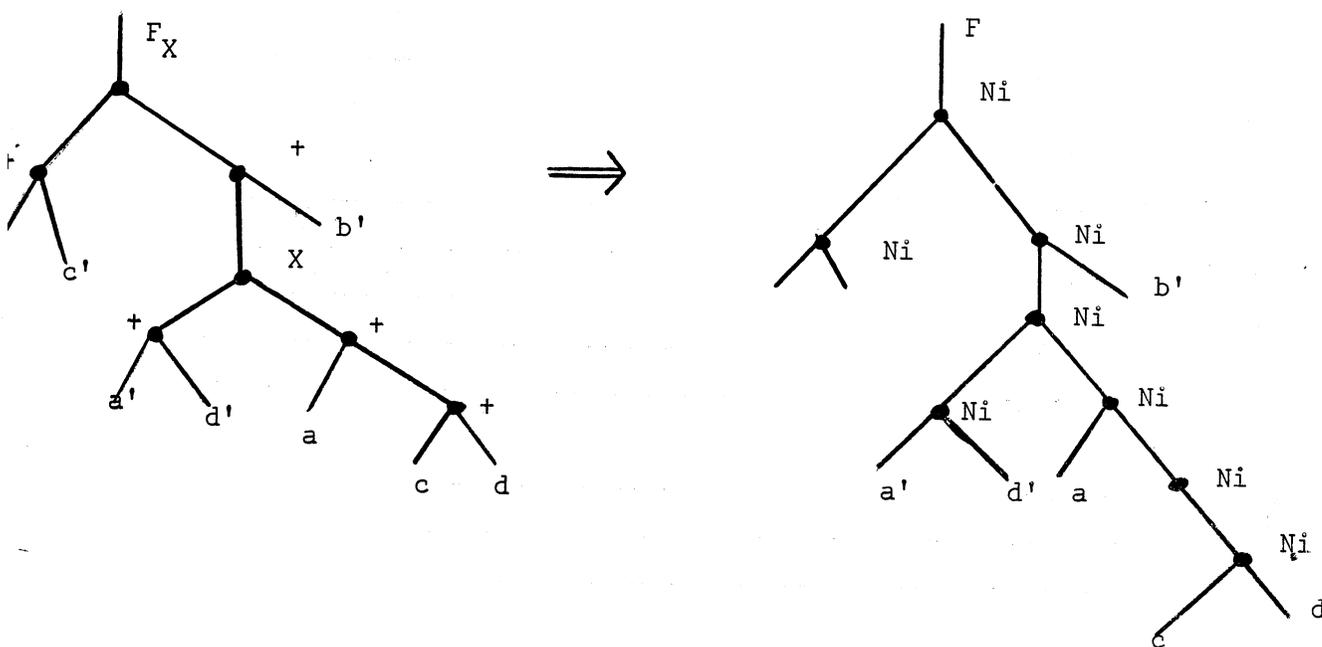
il correspond au nombre minimum de lettres (donc au nombre minimum d'opérateurs à 2 entrées)

le réseau Ni est également minimum en ce qui concerne le nombre de lettres.

4.2.2. Si le réseau en S et P ne correspond pas à une expression de signature.... PS ouSP, notamment si le nombre d'entrées des opérateurs est borné, il est également facile d'obtenir un réseau d'opérateurs NI, dont le nombre d'entrées est borné par le même nombre, réalisant la fonction.

On remplace tous les opérateurs + et X par des NI (on rajoute un NI à une entrée si le 1^{er} opérateur est un +), lorsque 2 opérateurs successifs du réseau SP sont identiques on intercale entre les 2 un opérateur NI à une seule entrée. On laisse les lettres sous la même forme si le nombre de NI en aval est pair sinon on prend leur forme opposée.

Donc dans le cas d'opérateurs NI ayant au plus 2 entrées nous auront un réseau quasi-minimal en ce qui concerne le nombre d'opérateurs à 2 entrées notamment avec l'exemple ci-dessous :



On trouve dans [4] les résultats de la synthèse en opérateurs NI à au plus 2 entrées des fonctions de 5 variables étudiées en 3.3. Le tableau II compare les résultats lorsqu'on transforme un réseau S et P en réseau NI par la méthode ci-dessus.

Tableau I

Fonctions	nombre de lettres minimum	nombre de lettres avec l'algorithme 1	nombre de lettres de l'écriture minimale 2 couches	nombre d'opérateurs à partir de l'écriture minimale 2 couches	nombre d'opérateurs avec l'algorithme 1
ce'a'a'+a'bcd+b'e'+ab'd+ab'c	> 9	12	16	7	10
a'bd+a'bc+abd'+b'c'	8	8	11	4	6
a'b'd'+bc'+cde'+abe'	10	10	13	5	7
c'e+c'd+ab'e'+a'bd'+b'de'+a'b'e	11	11	16	7	9
de+b'cd'+acd'+a'b'd'e'	8	8	12	5	7

Tableau II

Résultats de [4]

Résultats de 3.3.

Fonctions de 5 variables données en 3.3.	Nombre d'opérateurs			Nombre d'opérateurs		
	à 2 entrées	à 1 entrée	temps en sec.	à 2 entrées	à 1 entrée	temps en secondes.
1	14	2		10	5	2.8
2	14	1		11	8	4.6
3	22	3		9	9	3.4
4	10	1		6	6	2.0
5	15	2		9	7	3.6
	$\Sigma_1=75$	$\Sigma_2=9$	$\Sigma_3=57.0$	$\Sigma'_1=45$	$\Sigma'_2=35$	$\Sigma'_3=16.4$

Les réseaux que nous obtenons sont arborescents, pas ceux obtenus par [4].

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

