

n° d'ordre : 406

376
978
240

50376
1978
240

THESE

présentée

à l'Université des Sciences et Techniques de Lille

pour l'obtention

du grade de Docteur es Sciences Physiques

par

Jean-Yves Ranchin

Licencié es Sciences

Ingénieur EIM et IAG

**Contribution à la définition
et au traitement
des images numériques**



Soutenue le 13 Mars 1978 devant la Commission d'Examen :

| | |
|---------------|------------|
| MM. P. VIDAL | Président |
| L. BOLLIET | Rapporteur |
| S. CASTAN | Rapporteur |
| V. CORDONNIER | Rapporteur |
| Y. ANGEL | Examineur |
| M. BOURTON | Examineur |



030 043700 4

à Marie-Blanche

à mes parents

à mon épouse

à mes parents

Je désire par ces quelques mots exprimer ma reconnaissance aux personnes qui ont contribué à l'aboutissement de ce travail :

- Monsieur le Professeur VIDAL, pour avoir accepté la direction de ces recherches,
- Monsieur le Professeur BOLLIET, pour m'avoir guidé de ses conseils,
- Monsieur le Professeur CASTAN, pour m'avoir fait découvrir il y a déjà longtemps le traitement des images,
- Monsieur le Professeur CORDONNIER, pour l'aide et les conseils qu'il m'a donnés dans la rédaction de cette thèse,
- Monsieur le Professeur ANGEL, pour l'aide qu'il m'a apportée,
- Monsieur le Professeur BOURTON, pour avoir bien voulu participer au jury,
- Monsieur le Professeur BOUIX, directeur du CETHEDDEC, et à M. RIZZI pour leur aide très importante sans laquelle ce travail n'aurait pu être mené à bien,
- Monsieur le Professeur NAMIAN, qui m'a permis d'entreprendre et de poursuivre ces travaux dans son laboratoire,
- Messieurs BORRIN, BAREAU et DUBOIS, pour les études de simulation et de réalisation,
- Madame MULLER, pour une réalisation matérielle que j'ai particulièrement appréciée.

J . R .

SOMMAIRE

| | Pages |
|--|-------|
| <u>Résumé</u> | 3 |
| <u>Introduction</u> | 4 |
| 1 Origine et cadre de l'étude | 4 |
| 2 Présentation informelle des problèmes | 5 |
| 3 Conséquences et extensions | 10 |
| 4 Présentation du texte. | 10 |
| <u>Première partie</u> Informations structurées et supports. | 12 |
| <u>Chapitre 1</u> Lot d'informations, notions préliminaires, définition et propriétés. | 13 |
| 1.1 Introduction descriptive | 13 |
| 1.2 Lot d'informations | 16 |
| 1.3 Propriétés des lots et des supports | 20 |
| 1.4 Conclusion. | 22 |
| <u>Chapitre 2</u> Adressage des mémoires à accès direct, application a la réalisation des lots. | 24 |
| 2.1 Notion d'adressage | 24 |
| 2.2 Emploi de l'adressage relatif | 27 |
| 2.3 Structure induite par l'adressage indexé | 28 |
| 2.4 Conclusion. | 29 |
| <u>Deuxième partie</u> Images numériques et états des automates cellulaires, dimension des espaces discrets, réalisations associées. | 31 |
| <u>Chapitre 3</u> L'image numérique, lot d'informations. | 32 |
| 3.1 Images théoriques et expérimentales | 32 |
| 3.2 Image numérique quelconque | 35 |
| 3.3 Propriétés des relations de base | 37 |
| 3.4 Structures algébriques sur les relations de base | 43 |
| 3.5 Conclusion. | 46 |

| | | |
|-------------------------|--|-----|
| <u>Chapitre 4</u> | Images régulières et systèmes cellulaires, | |
| | dimension des espaces discrets. | 48 |
| 4.1 | Introduction | 48 |
| 4.2 | Relations orthogonales et connexité | 49 |
| 4.3 | Dimension d'un espace discret | 54 |
| 4.4 | Compatibilité et maillages | 64 |
| 4.5 | Systèmes cellulaires. | 70 |
| <u>Chapitre 5</u> | Réalisation des lots d'image, | |
| | structures d'indexation. | 72 |
| 5.1 | Rappels sur les structures algébriques | 72 |
| 5.2 | Structure générale par indexation | 72 |
| 5.3 | Index signés et structures locales | 75 |
| 5.4 | Réalisation des lots d'images | 76 |
| <u>Troisième partie</u> | Définition et simulation d'un système | |
| | de traitement numérique des images. | 80 |
| <u>Chapitre 6</u> | Architecture du système de traitement | |
| | d'images. | 81 |
| 6.1 | Les systèmes ou machines de traitement de | 81 |
| | l'information | |
| 6.2 | Méthodes de définition | 82 |
| 6.3 | Essai de définition d'une voie systémique | 85 |
| 6.4 | Architecture du système de traitement d'images | |
| | numériques | 92 |
| <u>Chapitre 7</u> | Description détaillée et simulation du | |
| | système. | 95 |
| 7.1 | Description détaillée | 97 |
| 7.2 | Simulation | 99 |
| <u>Conclusion</u> | | 112 |
| <u>Bibliographie</u> | | 114 |
| <u>Annexe</u> | | 118 |

Résumé

Désirant établir une définition formelle des images, adaptée au traitement numérique, on a été amené à définir une classe plus vaste d'informations, munie d'une structure algébrique : le lot d'informations.

On a alors précisé par restriction la définition d'une image et on a montré qu'à tout instant un automate cellulaire est également un lot d'informations. On établit leur dimension d'espace discret.

Utilisant un type particulier d'adressage, par indexation des mémoires à accès direct, on montre que l'on peut toujours construire un isomorphisme entre un lot d'informations et une telle mémoire. De là, on déduit une architecture de calcul adaptée au traitement des lots et des images et à la simulation des automates cellulaires.

INTRODUCTION

I - ORIGINE ET CADRE DE L'ETUDE

A l'origine de cette étude se trouvent les besoins en manipulation scientifique d'images qui ont pris une grande importance dans les dernières années. Ce développement a trois raisons majeures :

Une image de type photographique est un support qui accepte une grande densité d'information directement perceptible par l'homme. Il a donc un avantage comparable à celui du support magnétique sans en avoir l'inconvénient de ne pouvoir être lu par l'opérateur.

Les techniques d'enregistrement sont largement diffusées et employées, elles dépassent aujourd'hui la capacité des supports de restitution de gris (1 000 niveaux de gris par point ne peuvent être distingués sur une pellicule).

Enfin les ordinateurs donnent la possibilité de traitements précis sur les grandes masses de données disponibles.

Or, ces besoins grandissants sont assez mal satisfaits, souvent par le sous équipement des laboratoires mais surtout pour deux raisons touchant au fond du problème :

- la première tient à l'ambiguïté de la notion d'image et à l'absence d'une définition opératoire dans la perspective d'un calcul numérique (cf. § 2.1).
- la seconde est un manque d'adéquation de la structure séquentielle unilinéaire du fonctionnement des ordinateurs usuels face à la nature multidimensionnelle des images (cf. § 2.4).

On pourrait objecter à cela que des traitements optiques, directs, existent et sont très employés. S'y cantonner nous prive des ressources du calcul automatique et de la souplesse de sa programmation. Ces trai-

tements sont fondés sur des disciplines de la physique et ne relèvent pas de la formation que nous présentons ici.

Nous avons, suivant en cela R. Thom [1], essayé de mettre en évidence l'explication des phénomènes quantitatifs. Par exemple, nous n'avons pas cherché à quantifier l'information de structure (cf. § 2.2) dans la première partie, mais plutôt à expliquer sa nature et ce qui la distingue du contenu.

Plusieurs questions étaient posées dès l'abord du sujet ou l'ont été dans le cours de la recherche, nous les présentons ici pour éclairer la démarche suivie.

2 - PRESENTATION INFORMELLE DES PROBLEMES

Voulant aborder le traitement des images numériques ou numérisées, sur ordinateur, on rencontre quatre problèmes :

Qu'est-ce qu'une image numérique ?

Quelles sont ses caractéristiques ?

Quelles représentations logiques choisir pour la représenter ?

Comment définir une architecture de traitement en liaison avec l'algorithmique en cause ?

L'analyse informelle des réponses à ces questions nous a amené aux développements suivants :

2.1. Qu'est-ce qu'une image ?

Dans l'acception usuelle, sont des images : les dessins, la peinture, les photographies, les radiographies, les sculptures, etc..., plus généralement les représentations en surface ou en volume de scènes ou d'objets. Les nécessités de la manipulation scientifique ont amené

A. ROSENFELD [2] à définir, non pas les images, mais les fonctions d'images. Curieusement, pour un essai de formalisation, il établit qu'une fonction d'image est indistinguable successivement :

- d'une fonction absolument continue,
- d'une fonction discrétisée,
- d'une fonction quantifiée.

Introduites ainsi comme objets de l'analyse mathématique, ces fonctions d'images se ressentent, dans toute la suite [2] de l'hypothèse initiale de continuité, absente dans tous les cas d'application par les limitations dues aux capteurs et à l'appareillage de traitement. Or, l'hypothèse de continuité présente des inconvénients sérieux quand elle ne peut pas être vérifiée, au moins asymptotiquement, ce qui est le cas des images définies par un nombre fini de points et de niveaux de gris.

Par exemple :

- on sait que la notion de dimension d'un espace topologique discret est distincte et irréductible à celle d'un espace continu [3].
- la discrétisation d'un modèle différentiel modifie les singularités et le domaine de stabilité du système décrit [4].
- les propriétés de connexité d'un sous-espace et de son complément ne sont pas conservées dans un espace discret [5], [6].

C'est pourquoi, nous proposons de définir une image de façon à rendre compte, dès l'abord, de sa nature discrète imposée par l'instrumentation et, à la fois, disposer d'un formalisme adapté à l'outil de calcul numérique.

2.2. Quelles sont les caractéristiques d'une image ?

On peut donner des réponses qui s'organisent en trois classes : les réponses quantitatives, les réponses descriptives et les réponses explicatives.

Les réponses quantitatives sont du type : quantité d'informations au sens de Shannon, géométrie et métrique des déformations, analyse de gradients. Elles participent également au choix des procédures de traitement et de transmission.

Les réponses descriptives peuvent être faites à deux niveaux au moins : celui des origines et des supports d'images d'une part, celui de l'analyse des formes et des scènes d'autre part. Le premier niveau peut servir à choisir des capteurs adaptés au support fourni, à définir le type de traitement : compression d'information, extraction de paramètres, ... etc. Le deuxième niveau donne des indications sur les algorithmes à appliquer pour atteindre un but fixé : classification p.e.

Nous avons rangé dans la troisième classe : réponses explicatives d'autres caractères propres à la notion d'image et non générateurs de partitions dans l'ensemble des images. Ils sont essentiellement attachés au signal et à l'information portée :

Le signal d'image est essentiellement multidimensionnel : deux dimensions d'espace et une luminance pour une photographie ; d'autres dimensions encore si l'on prend le même cliché dans plusieurs canaux. Quatre dimensions au moins, pour une sculpture colorée. Cette observation banale a des conséquences importantes, non seulement sur les algorithmes de traitement comme on peut s'y attendre, mais aussi sur les notions de calcul d'image et d'information d'image. On trouvera, dans la première partie de cette étude, un lien entre les images et une variante du schéma de Von Neumann de calcul à une adresse.

Quant à l'information portée par une image, on peut reprendre le résumé de C. Castoriadis [7] qui, comme Mac Kay [8], [9], Payne [10], Lowitz [11], s'interroge sur la capacité explicative de l'information quantitative : " Le concept strict de quantité d'information (et son

équivalent thermodynamique) établi par la théorie du même nom, suffit-il à expliquer, ou même à décrire le fonctionnement et le comportement d'un automate tant soit peu complexe ? Ou bien, est-il indispensable d'introduire d'autres dimensions de l'information, et peut-être d'autres concepts - comme pertinence, poids, valeur, signification de l'information ou du "message" - pour saisir l'automate ? ".

Dans la première partie ci-après, nous avons délibérément choisi, en en tirant les conséquences dans la 3ème partie, de définir une image en séparant l'information purement quantitative (ensemble de nombres décrivant la luminance p.e.) de l'information apportée par la structure de l'image.

Pour illustrer cette distinction, prenons l'exemple d'un jeu à damier :

L'image constituée par une situation dans un jeu d'échecs peut se résumer, si l'on s'attache au jeu proprement dit et non à sa facture, en deux points : la description du damier, support du jeu constitué d'éléments jointifs selon un certain motif, les cases ; l'inventaire des positions occupées et la nature de l'occupant : tour noire, roi blanc, ... etc.

On se trouve, ainsi, en présence de deux types d'informations irréductibles l'un à l'autre : d'une part, l'information de structure du support, d'autre part, l'information de valeur ou de poids qui est attachée à chaque case vide, occupée par un cavalier, une reine, ... etc.

Les auteurs cités plus haut [8] à [11], ont introduit sous des noms différents : Logons, Varons, des mesures d'information de structure. Nous n'en ferons pas usage, mais à informations de natures distinctes, nous appliquerons des traitements distincts, ce qui est un début de réponse à la quatrième question. [\$2.4]

2.3. Quelles représentations logiques choisir ?

Nous entendons, par représentation logique, un mode de représentation non ambigu dans une machine. Deux types de machines sont envisageables :

- un automate spécifique : un réseau cellulaire, par exemple (*).
- un calculateur universel, avec sa souplesse algorithmique, mais l'inconvénient majeur de son fonctionnement séquentiel pour des données multidimensionnelles.

Pour conserver cet acquis, nous avons choisi ce dernier, mais nous y représenterons différemment les deux informations à manipuler.

L'information numérique sera inscrite en mémoire sans difficultés. L'information de structure devra être représentée par une structure puisque nous ne l'avons pas rendue quantitative. Ce sera réalisé par une organisation de la mémoire superposée à son ordre total (première partie : chapitre 2).

2.4. Comment définir une architecture de traitement en liaison avec l'algorithmique en cause ?

Une première démarche, d'origine purement algorithmique, consiste à définir, a priori, une sémantique de langage d'instruction [12], de là, en déduire un interpréteur et, utilisant des formalismes récents [13], une architecture d'exécution.

Cette démarche est en quelque sorte guidée par les traitements que l'on veut exécuter. Elle se prête bien à la définition de systèmes de calcul adaptés aux besoins divers de classes d'utilisateurs tels que

(*) mais on ne pourra y exécuter aisément des calculs très différents.

géophysiciens dans l'exemple du 13 , physiciens, gestionnaires.

Or, nous ne disposons pas des types de traitements déterminés à appliquer à des informations diverses, mais nous avons des données d'une classe particulière sur lesquelles les traitements sont sinon quelconques, du moins non encore caractérisés. On a donc, dans une autre démarche, que l'on propose, cherché à spécifier la matière à traiter et à en déduire une organisation générale du système.

3 - CONSEQUENCES ET EXTENSIONS

La décomposition d'un lot d'informations en nombres et support permet de considérer l'état global instantané d'un automate cellulaire comme un lot. La correspondance ainsi introduite permet d'envisager l'utilisation du système proposé pour les images, à la simulation de ces automates cellulaires.

4 - PRESENTATION DU TEXTE

Il est ordonné en trois parties.

La première partie contient deux chapitres.

Dans le premier chapitre on définit les lots d'informations ou images numériques quelconques. On en établit les propriétés essentielles, ainsi que la notion de support de lot. On caractérise enfin le support universel sur lequel tout lot d'information peut être réalisé.

Dans le second chapitre on montre qu'un processeur quelconque muni de l'adressage indexé fait un support universel d'une mémoire ordonnée. On explicite le processus de réalisation d'un lot d'informations sur ce support.

La deuxième partie contient trois chapitres.

Le premier est consacré aux relations de base, constitutives des structures algébriques des images.

Dans le second, on précise ces structures et l'on définit leur dimension en tant qu'espaces discrets. On montre que les automates cellulaires peuvent être considérés à chaque instant comme des images numériques.

Le troisième chapitre contient la caractérisation du processus d'indexation qui permet l'implantation des images usuelles sur une mémoire banale.

La troisième partie contient deux chapitres.

Dans le premier chapitre on définit l'architecture d'un système de traitement.

Dans le second, on donne la description détaillée du système et sa simulation.

P R E M I E R E P A R T I E

INFORMATIONS STRUCTUREES ET SUPPORTS

Après une introduction descriptive, on définit le lot d'informations comme un couple formé d'un ensemble de valeurs et d'une structure algébrique la plus générale sur cet ensemble. On aborde ensuite la notion de support qui peut être affecté à un lot. Le support contient la structure algébrique. Il pourra ultérieurement être identifié à la mémoire à accès direct d'un ordinateur numérique, on fournit les schémas d'isovalence et d'équivalence des lots. Un certain nombre de définitions et de propriétés peuvent être établies localement, c'est à dire sur un sous-ensemble de l'ensemble des valeurs. Elles sont détaillées à la fin du premier chapitre.

Le second chapitre est consacré aux propriétés générales de l'indexation dans un ordinateur numérique. On identifie l'indexation avec une structure algébrique sur l'ensemble des adresses. En montrant que les relations d'indexation sont équivalentes aux relations binaires sur cet ensemble, on dote la mémoire de la propriété de support universel.

CHAPITRE 1

LOT D'INFORMATIONS, NOTIONS PRELIMINAIRES, DEFINITION ET PROPRIETES

1.1. Introduction descriptive

Un système, technologique ou vivant, non isolé, est sensible à des apports extérieurs : ordres et bruits que l'on groupe sous le nom d'informations. On apprécie leur influence sur le récepteur par le changement d'état qu'elles provoquent, ou le maintient dans un état que le système, libre, aurait quitté.

Ces informations dont on ne restreint pas la généralité peuvent être interprétées par le système, isolément ou en groupe. Isolément, elles portent chacune une signification propre, indépendante de celles de leurs voisines dans le temps ou l'espace. Sinon, elles sont en groupe, c'est-à-dire liées par des relations spatiales ou temporelles, relations appréciées par le système (on ne considère pas ici les voisinages de sens, à la base de processus d'apprentissage et de reconnaissance). On suppose ainsi donnée a priori une structure entre informations.

Intuitivement, les plus simples parmi ces relations engendrent des structures linéaires dont la construction et la manipulation sont usuelles.

La structure linéaire est naturelle si les informations sont ordonnées, c'est le cas des observations successives au cours du temps d'un même phénomène sur une échelle de repérage ou de mesure.

Elle peut n'être qu'un artifice de commodité de représentation ou une nécessité d'ue à l'emploi d'un support particulier. C'est le cas pour des informations complexes que l'on veut exprimer par l'écriture fondamentalement linéaire. Il en est de même lorsqu'on les manipule dans un calculateur numérique : la structure ordonnée des principaux organes de mémoire, l'habitude et la facilité d'emploi des listes, piles et tables conduisent fréquemment à exprimer des organisations non linéaires de façon linéaire, quitte à employer plus ou moins rationnellement des instruments hétérogènes par rapport aux précédents, comme les pointeurs.

Ces organisations non linéaires (au sens de la succession ou de la relation d'ordre total) se rencontrent fréquemment : écriture chinoise, dans chaque caractère, jeux à damier, ... etc. Le point de vue de l'informaticien a été résumé par C. PAIR [14] qui distingue trois niveaux de structures.:

- . niveau mathématique, formalisé, algébriquement ou non.
- . niveau abstrait, qui correspond à la représentation intuitive : graphe, tableau n-dimensionnel.
- . niveau concret, celui de la représentation adoptée en mémoire d'ordinateur : table, liste, chaînage.

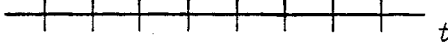
Cette notion de structure, proche de la théorie des systèmes, amène à une première distinction entre information et support. Trois exemples facilitent la mise en évidence du contenu informationnel proprement dit d'un support par les relations entre les informations qu'il contient.

Exemple 1

Une séquence d'entrée d'un automate impulsif est un groupement d'informations, de structure linéaire en ce que les unes et les autres se placent selon une relation d'ordre temporel. Le support est constitué par les ins-

tants d'apparition en relation d'ordre, les valeurs sont les informations proprement dites : + V ou 0.

V, 0, 0, V, 0, V, V, 0



séquence de valeurs

support

Exemple 2

La description d'une famille se compose des renseignements d'état civil propres à chaque personne et des renseignements de filiation qui constituent les relations entre parents et enfants. On établit un support par un graphe arborescent dont les noeuds portent les individus et les liaisons portent les relations de filiation. Ce support n'est pas le seul possible, on pourrait, à chaque génération, par exemple établir une simple liste des membres de la famille. On est ici au niveau d'une structure abstraite au sens de C. PAIR [14] construite au mieux pour les besoins de la transmission écrite.

Certains groupements d'informations sont déjà pourvus dans le domaine sensible d'un support. C'est le cas des jeux sur damiers ou sur grille.

Exemple 3

Le jeu de Go se joue sur une grille de 19 x 19 intersections de lignes. Les autres éléments sont deux sortes de jetons. La valeur d'une configuration s'apprécie par les capacités relatives d'encerclement des deux sortes de jetons, on est là dans le domaine du sens du jeu. Au dessous, dans l'ordre de complexité, on relève les relations de voisinage entre jetons. Ces relations sont portées par la structure de la grille, indépendante de la configuration du jeu.

Revenant aux mémoires à accès direct, on distinguera l'atome de mémoire, élément technologique, le nom ou adresse de cet atome et le contenu de l'atome suivant Raymond[15]. On distinguera également les relations entre ces atomes, supports des contenus, introduites par la manipulation des registres des relations entre les contenus (égalité par exemple).

Ces distinctions se retrouveront dans l'être très général, le lot d'informations que l'on définit ci-après sous une forme qui rendra ultérieurement aisée l'attribution d'un support.

1.2. Lot d'informations

Notations : On utilise les notations usuelles du produit cartésien $E \times E$ d'un ensemble E par lui-même et de la relation binaire r définie sur E : $r \subseteq E \times E$. De même un automate a sera défini par un quintuplet $a \triangleq (X, Q, S, f, g)$ où X est l'ensemble des entrées, Q celui des états internes, S celui des sorties, f et g sont des applications :
 $f : X \longrightarrow Q$, $g : Q \longrightarrow S$.

Définition 1 : Ensemble d'informations.

Un ensemble d'informations I est une collection homogène ou hétérogène d'éléments divers : noms, nombres, propositions, énoncés.

Considérant une telle collection, où les répétitions des éléments sont admises, on distinguera :

- les éléments, items, au sens de la théorie ensembliste. Ils formeront l'ensemble des données : $D(I)$. Ces éléments ne peuvent être désignés que par leur énoncé.
- les noms des éléments, débarassés de leur énoncé. Ici ne figure aucune répétition, puisque deux éléments distincts de $D(I)$ ont des noms différents, même si leur énoncé est le même. Ils for-

ment $N(I)$

- l'ensemble des valeurs $V(I)$ formé des éléments de $D(I)$ sans répétition.

On peut alors définir le lot d'informations par :

Définition 2 : Lot d'informations.

Un lot d'informations est un doublet (I, R_I) où I est un ensemble d'informations et R_I un ensemble de relations binaires défini sur $N(I)$ et par là sur $D(I)$.

Exemples

Considérons une partie d'échecs à un instant donné. On trouvera l'ensemble des noms (et couleurs) de ces pièces $D(I)$, les mêmes pièces considérées comme un ensemble d'éléments par l'affectation d'un indice quelconque $N(I)$. Les relations entre les objets de I sont établies sur l'échiquier indépendamment des noms $d(i)$ par exemple :

---" est à la droite immédiate de "---

---" est immédiatement au dessus de "---

Ces relations ont pour arguments les $d(i)$ ou les $n(i)$ mais non les $v(i)$ pris dans $V(I)$.

Si l'on reprend l'exemple d'une famille, X né le ..., fils de ... et de ... est un élément de I , son énoncé un élément de $D(I)$. Les relations peuvent être construites à partir de tels énoncés.

Remarque : L'ensemble de relations binaires R_I n'est pas nécessairement unique. en effet, on peut remplacer dans le dernier exemple, le couple :

r : ---" est le père de "---

r' ---" est la mère de "---

par le couple :

r'' ---" est le fils de "---

r''' ---" est la fille de "---

sans changer l'organisation de la filiation. On ne peut donc, dans le cas le plus général, poser qu'une hypothèse :

l'ensemble R_I des relations binaires sera toujours supposé décrire complètement les relations de structure de I .

On se place ainsi dans le cadre des structures algébriques quelconques (cf. Lévy Bruhl [16]).

Notion de support

On a vu dans le § 1.1 l'intérêt que peut représenter la distinction entre information et support vis à vis des relations de structure.

Une définition purement abstraite est la suivante :

Soit une collection C d'éléments tous identiques repérés par un indice. Elle constituera un support d'un lot quelconque $L = (I, R_I)$ par la construction d'une injection :

$$N(I) \longrightarrow C$$

$$\text{et d'un homomorphisme } h : R_I \longrightarrow \bar{R}_C$$

où \bar{R}_C est l'ensemble de toutes les relations binaires construites sur C .

Cette définition englobe les représentations usuelles des informations : dessins, graphes. Elle a besoin d'être précisée si l'on veut la rendre opératoire pour le calcul numérique. C'est pourquoi on adoptera pour la suite un cas particulier du support le plus général :

Un élément de support est un automate sans mémoire

$$a \stackrel{\Delta}{=} (X, Q, S, f, g) \text{ où } X = Q = S \text{ et } f = g$$

Un support S d'un ensemble d'informations est un ensemble d'éléments de support repérables par un indice (ou adresse).

Un support est structurable si l'on peut définir des relations binaires entre éléments de S .

Affectation d'un support à un ensemble d'informations

Sous réserve de $|S| \geq |I|$ où $| \cdot |$ désigne le cardinal d'un ensemble, on affecte à I le support S par la construction de deux injections :

$$\phi_1 : N(I) \longrightarrow S$$

$$\phi_2 : V(I) \longrightarrow X$$

ϕ_2 est une opération de codage des éléments de $V(I)$ dans l'ensemble des états des éléments de support. On considérera plus généralement dans la suite une injection $V \longrightarrow X$ où $V(I) \subseteq V$. V est alors l'ensemble de toutes les valeurs possibles pouvant figurer dans un lot de cette nature ; par exemple, l'échelle de quantification du signal, même si, dans le lot, toutes les valeurs ne sont pas atteintes. Nous pouvons alors passer du support d'un ensemble d'informations à un support de lot.

Définition :

Un support de lot (I, R_I) est un couple (S, R_S) défini par :

S est un support de I

Un homomorphisme $\psi_1 : R_I \longrightarrow \bar{R}_S$ où \bar{R}_S est l'ensemble de toutes les relations binaires définies sur S .

Remarque : Si l'on restreint S à $E = \phi_1(N(I))$, la restriction correspondante ψ de ψ_1 est un isomorphisme et ϕ restriction de ϕ_1 une bijection.

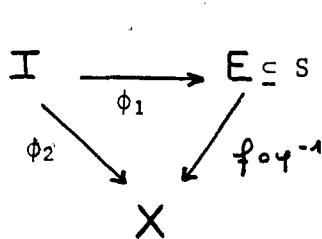
Les ensembles d'informations s'avèrent, comme on l'avait annoncé, voisins de la notion de mémoire par la distinction entre l'objet, le nom de l'objet et le contenu de l'objet. C'est à cette propriété que l'on devra ultérieurement, au moins en partie, la réalisation d'un lot sur une mémoire.

Le lot d'informations et le support de lot tels qu'on les a définis, unifient des notions éparses dans les publications : les images numéri-

ques et l'état global d'un automate cellulaire sur lesquels nous reviendrons longuement, mais aussi les structures de données de Knuth [17], organisations en réseaux, arbres, listes qui permettent d'inscrire des structures abstraites selon [14] en mémoire ; les structures de données et les graphes de données de Rosenberg [18], graphes fortement connexes et structures de données vues comme des collections d'items munies de relations sur ces items (data structure viewed as a collection of primitive data items in conjunction with a set of relations on these items).

1.3. Propriétés des lots et des supports

D'après les définitions du § 1.2 la réalisation (ou inscription) d'un lot d'informations $L \stackrel{\Delta}{=} (I, R_I)$ sur un support de lot (S, \bar{R}_S) s'exprime par les deux schémas :



$$R_I \begin{array}{c} \xleftarrow{\psi^{-1}} \\ \xrightarrow{\psi} \end{array} R_{E \subseteq R_S}$$

où E et R_E sont des sous ensembles respectivement de S et R_S .

et où I est pris, pour ϕ_2 , au sens de $V(I)$ et, pour ϕ_1 au sens de $N(I)$, ce qui n'altère pas la compréhension du schéma.

Par l'intermédiaire des supports, on peut établir des relations entre lots :

Relations entre structures : Lots isomorphes et homomorphes

Deux lots L_1 et L_2 seront dits respectivement homomorphes ou isomorphes si, et seulement si les structures algébriques des supports (E_1, R_{E_1}) et (E_2, R_{E_2}) sur lesquels ils sont réalisés sont respectivement homomorphes ou isomorphes.

Remarque : Deux lots isomorphes ou homomorphes n'ont pas nécessairement même ensemble de valeurs. Deux supports isomorphes ou homomorphes n'ont pas nécessairement, pour leurs éléments, le même nombre d'états internes. La propriété d'isomorphisme induit sur un ensemble de lots une partition en classes d'équivalences.

Appliquée aux traitements numériques, cette propriété sera utilisée dans la machine décrite dans la troisième partie. Elle implique que, pour des lots isomorphes, les programmes de traitement sont identiques.

De même l'homomorphisme induit une relation d'ordre sur un ensemble de lots.

Exemples :

Deux images numériques, comme on le verra dans la deuxième partie, sont isomorphes pourvu qu'elles aient même trame ou maillage.

Deux bancs de mémoire de même dimension en nombre de mots, mais dont les mots ont des longueurs différentes en nombre de bits, peuvent être isomorphes en tant que support et en tant que lot réalisé si leurs mécanismes d'accès sont identiques.

Relations entre valeurs

Lots d'informations isovalents.

Deux lots $L_1 \stackrel{\Delta}{=} (I_1, R_{I_1})$ et $L_2 \stackrel{\Delta}{=} (I_2, R_{I_2})$ sont isovalents si et seulement s'il existe une bijection $D(I_1) \longleftrightarrow D(I_2)$.

De ce fait, et à l'opposé des relations entre structures, celle-ci ne porte que sur les informations. Elle sert à caractériser deux lots qui ne diffèrent que par leur structure algébrique.

Exemples :

Un lot constitué par un mot alphabétique a pour isovalents, les anagrammes du mot.

Deux images numériques sont isovalentes si leurs histogrammes sont identiques.

Relations mixtes

Lots équivalents.

Deux lots sont équivalents si et seulement s'ils sont isovalents et isomorphes.

Exemples :

Une phrase codée en remplaçant chaque lettre par celle qui la suit avec un décalage circulaire de n positions est équivalente à la phrase originale.

Deux images ne différant que par une translation des radiométries des points sont équivalentes.

Propriétés locales

La réalisation d'un lot d'informations comme les relations précédentes entre lots seront dites locales si elles ne sont vraies que pour un sous-ensemble du lot $L' \subseteq L$, c'est à dire pour $I' \subseteq I$ et la restriction de R_I à I' .

1.4. Conclusion

Dans ce chapitre, nous avons essayé de donner des bases précises à des notions souvent présentées comme intuitives. L'essentiel des problèmes posés dans l'introduction demeure. Leur solution sera déduite des définitions précédentes. On précisera l'organisation des mémoires

qui permettent la réalisation des lots dans le chapitre suivant.

Dans la deuxième partie, on traitera des images et des structures cellulaires en tant que lots à structure algébrique particulière.

Enfin, dans la troisième partie, on utilisera la distinction entre traitement des valeurs d'un ensemble d'informations et traitement de sa structure algébrique pour la conception partielle, puisqu'indépendantes des algorithmes, d'un système de traitement de l'information.

CHAPITRE 2

ADRESSAGE DES MEMOIRES A ACCES DIRECT, APPLICATION A LA REALISATION DES LOTS

2.1. Notion d'adressage

Les mémoires à accès direct auxquelles nous nous intéressons sont caractérisées par le fait que, l'adresse de l'emplacement en mémoire (entier positif) étant donné et inscrit dans un registre (registre d'adresse de mémoire), un dispositif électronique câblé réalise la lecture et le chargement dans un autre registre du contenu de l'emplacement.

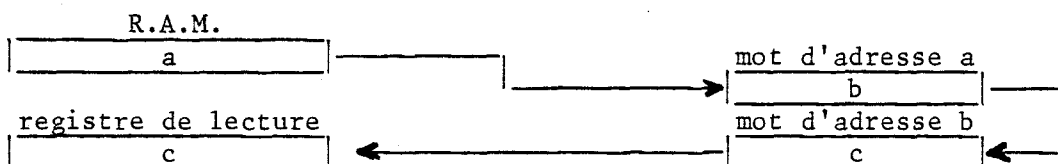
On appelle modes d'adressage le résultat de l'application de dispositifs annexes qui permettent, à partir du contenu du registre d'adresse de mémoire, de lire le contenu d'un autre emplacement.

Les plus courants parmi ces modes sont :

- l'adressage indirect,
- l'adressage relatif,
- l'adressage différentiel,
- l'adressage indexé.

L'adressage indirect

Dans cet adressage le contenu lu est celui du mot dont l'adresse est le contenu du mot dont l'adresse était dans le registre d'adresse de mémoire (R.A.M.) :

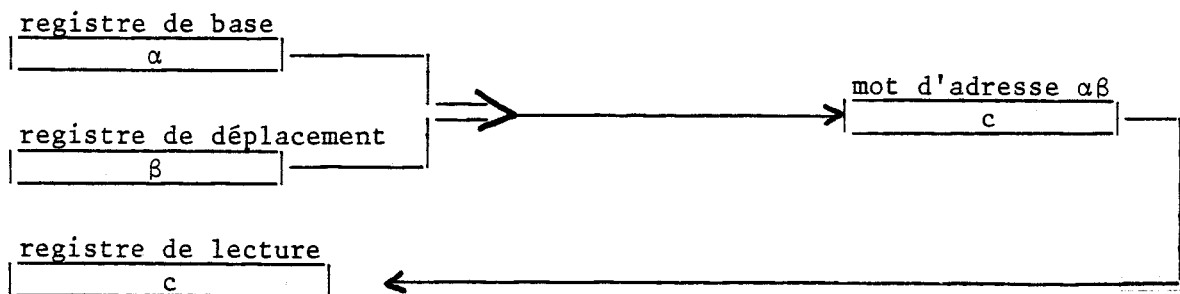


Si l'on poursuit le procédé, on chargera dans le registre de lecture non c , mais le contenu du mot d'adresse c etc... On réalise ainsi un adressage indirect à 2, 3, etc... niveaux de profondeur.

L'adressage relatif

Ici, le contenu du R.A.M. n'est qu'une partie (partie basse ou de faible poids, ou encore de déplacement) de l'adresse complète du mot dont on ira chercher le contenu.

La partie haute est contenue dans un autre registre, dit registre de base. L'adresse complète est obtenue par concaténation des deux parties.



L'adressage différentiel

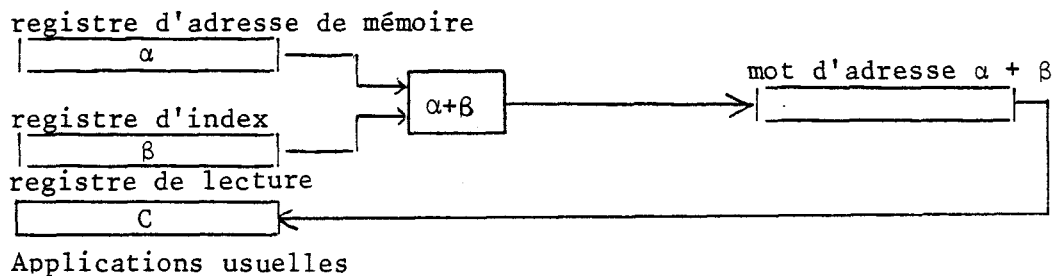
Le contenu du registre d'adresse de mémoire est formé par l'addition ou la soustraction au contenu du compteur ordinal (adresse courante du programme en cours d'exécution) d'une quantité entière.

Ce mode d'adressage permet des sauts limités dans les deux sens, dans le programme : retour à une adresse antérieure, ou saut à une adresse postérieure.

L'adressage indexé

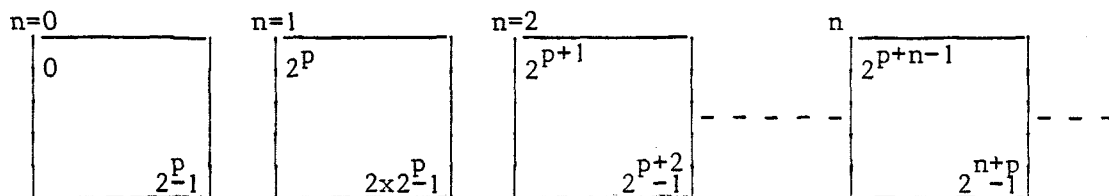
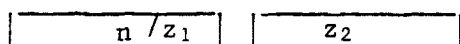
L'adresse du mot à lire est constituée par addition du contenu du registre d'adresse de mémoire au contenu d'un registre signé ou par addition ou soustraction au contenu d'un registre non signé.

Ce registre supplémentaire est un registre d'index.



Les deux modes d'adressage les plus couramment utilisés sont le relatif et l'indexé..

L'adressage relatif permet, à partir d'une adresse fixe $n \cdot 2^p$ où p est la longueur en bits du registre de déplacement et n le contenu du registre de base, de parcourir un banc ou page de dimension 2^p en nombre de mots



Le câblage de ce dispositif introduit une pagination de l'espace de mémoire.

Dans le cas où l'on veut atteindre un mot d'adresse quelconque z , on décompose de façon unique z en z_1 et z_2 tels que z_1 soit de la forme $n \cdot 2^p$ (partie base) et $z_2 \leq 2^p - 1$ (partie déplacement) que l'on charge séparément dans les registres correspondants.

L'adressage indexé n'introduit pas de pagination. Il permet, à partir d'une adresse quelconque de faire des sauts d'amplitude fixée par l'index β .

$$\alpha \longrightarrow \alpha + \beta$$

$$\alpha + 1 \longrightarrow \alpha + 1 + \beta. \quad \text{etc...}$$

2.2. Emploi de l'adressage relatif

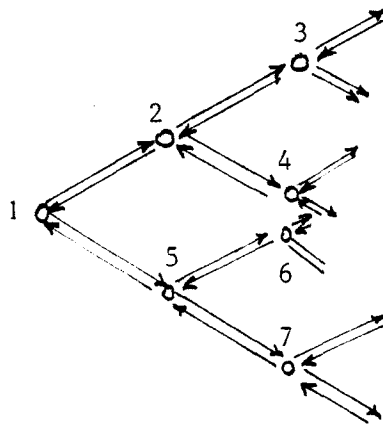
Dans une publication citée au § 1.2.[18], A. Rosenberg a systématisé l'emploi de l'adressage relatif aux graphes de données (data graphs).

Un graphe de données est pour l'auteur un couple ordonné $\Gamma = (C, \Lambda)$ où C est un ensemble dénombrables de cellules de données comparables à nos éléments de support, et Λ est un ensemble fini de transformations partielles sur C sous la condition de connexité forte, c'est à dire que pour tout c et $d \in C$ il existe une transformation ξ au moins du monoïde libre Λ^τ construit sur Λ : $\xi \in \Lambda^\tau$ telle que $c \xi = d$.

Identifiant ces transformations partielles (pas dans le graphe) à un sous ensemble de \bar{R}_S , ces graphes de données sont un cas particulier des supports de lots généraux définis plus haut.

Un cas particulier des graphes de données est constitué par ceux qui ont une racine. Cette racine est un élément $c_0 \in C$ tel que pour tout $d \in C$, $\xi \in \Lambda^\tau$ est unique, tel que $c_0 \xi = d$, c'est à dire qu'à partir de c_0 tout élément du graphe est atteint par un chemin unique.

Exemple



La cellule 1 est racine du graphe

Le résultat principal obtenu par l'auteur est que :

Les graphes de données à racines sont réalisables par l'adressage relatif. C'est à dire que le schéma d'adressage relatif peut être rendu isomorphe de l'ensemble des transformations sur le graphe.

Les résultats que l'on présente dans la suite diffèrent pour leur partie comparable sur deux points majeurs : le mode d'adressage est l'indexé, les graphes, quelconques au départ, ne seront jamais des graphes à racines.

2.3. Structure induite par l'adressage indexé

Hypothèses sur l'espace des adresses.

Nous admettrons dans la suite que le nombre des atomes ou mots de mémoire est borné par un nombre $N > 0$ arbitraire et donc potentiellement infini.

On réservera également la possibilité de numérotter les atomes en utilisant des entiers relatifs par une application isotone $[0, N] \rightarrow \mathbb{Z}$.

Élément de support et support d'information

Un mot de mémoire vive correspond très exactement à la définition du § 1.2. d'un élément de support.

Un ensemble de mots de mémoire organisés par un dispositif d'accès en lecture et écriture et muni d'adresse est un support d'ensemble d'informations. Pour en faire un support de lot il faut pouvoir lui donner une structure isomorphe à celle du lot.

Proposition

L'indexation appliquée à un sous ensemble M_a de l'ensemble des adresses M est une relation binaire entre adresses.

Si $|M| = N$, toutes les opérations qui suivent s'entendent modulo N .

En effet, soit $a \in M_a$, p un index, son application à a construit $a + p \in M$ (on suppose toujours que $a + p \leq N$), adresse d'un mot.

Soit plus généralement τ_p l'opération d'index appliquée à $M_a \subseteq M$; $\tau_p(M_a) \subseteq M$ et $(M_a \times \tau_p(M_a)) \subseteq M \times M$ et est donc une relation binaire sur M .

Proposition

Toute relation binaire r sur l'ensemble M des adresses est décomposable en un ensemble fini d'opérations d'indexation.

Soit $r \subseteq M \times M$, $r = (a_{11}, a_{21}), (a_{12}, a_{22}) \dots (a_{1n}, a_{2n})$

Notons τ_k l'opération d'indexation d'index k applicable au mot d'adresse α

Construisons la suite d'index (a_{1i}, τ_{p_i}) $i=1$ à n où $p_i = a_{2i} - a_{1i}$. l'application de l'index correspondant à un élément quelconque a_{1i} , construit le couple $(a_{1i}, a_{1i} \tau_{p_i}(a_{1i}))$ et la suite d'index construit r .

Admettant la définition suivante :

Un support de lot est dit universel si l'on peut y imposer toute structure algébrique, c'est à dire si toute partie de \bar{R}_S est réalisable, alors :

Théorème fondamental

Une mémoire est un support universel par indexation.

2.4. Conclusion

Nous disposons des éléments de base nécessaires pour la suite de l'étude, à savoir :

Une classe générale d'informations pourvues de structures indépendantes des contenus, un instrument de réalisation isomorphe à savoir les

mémoires à accès direct munies d'index.

On a établi que les travaux de nature voisine cités peuvent être considérés comme des cas particuliers de nos résultats : les graphes de données et structures de données vis à vis des lots d'information.

Les schémas d'adressage relatif sont aussi des cas particuliers de l'adressage indexé.

En effet, un système base-déplacement est un index statique particulier :

Soient $a_{p-1} \dots a_1 a_0$ les chiffres du déplacement et

$a_n \dots a_{p+1} a_p$ ceux de la base

L'adresse courante du système à indexation sera :

$$\underbrace{0 \dots 000}_{n-p} a_{p-1} \dots a_1 a_0$$

et l'index : $a_n \dots a_{p-1} a_p \underbrace{00 \dots 0}_p$

P

D'un point de vue pratique, les lots d'informations n'ont pas une structure absolument quelconque qui serait très lourde à manipuler.

Nous étudierons plus particulièrement les structures algébriques représentant des images numériques. Elles s'avèreront "aisément" réalisables, c'est à dire nécessitant un nombre restreint d'index. De plus, ces structures s'avèreront régulières ; un index ne sera pas affecté à une adresse, mais applicable à presque toutes les adresses.

DEUXIEME PARTIE

IMAGES NUMERIQUES ET ETATS DES AUTOMATES CELLULAIRES

DIMENSION DES ESPACES DISCRETS

REALISATIONS ASSOCIEES

Après une étude expérimentale, on construit les images numériques quelconques comme des lots d'informations dont la structure algébrique est constituée par des relations binaires originales.

On étudie ensuite les propriétés et les lois de composition de ces relations et la notion de dimension d'un espace discret. On propose pour cela une nouvelle définition d'un tel espace qui généralise une définition connue.

Les images usuelles et les supports d'automates cellulaires sont ensuite précisées par les maillages, cas particuliers des structures algébriques.

L'application systématique de l'indexation d'une mémoire est utilisée pour la réalisation des maillages, on retrouve alors les espaces discrets à structure de groupe.

On conclut sur l'utilité de systèmes de traitement de l'information adaptés aux données à structures de lot.

CHAPITRE 3

L'IMAGE NUMERIQUE, LOT D'INFORMATIONS

3.1. Images théoriques et expérimentales

Nous avons, de nombreuses fois, employé les mots image, et image numérique. De quoi s'agit-il ? Et tout d'abord qu'est-ce qu'une image !

Rosenfeld [2], comme on l'a vu dans l'introduction, pose qu'une image est indistingable successivement :

- d'une fonction absolument continue,
- d'une fonction discrétisée,
- d'une fonction quantifiée.

L'examen des publications récentes les plus complètes montre, tantôt une absence de définition [19], tantôt une définition en termes de fonctions :

Fonction de deux variables $g(x, y)$, réelle, définie dans un domaine fini du plan xy , ou, ce qui revient au même, définie partout, mais nulle à l'extérieur de ce domaine : Estournet [20], soit encore des définitions purement discrètes : un sous ensemble d'un groupe abélien fini chez Mylopoulos et Pavlidis [21]

Admettons pour l'instant la définition en termes de fonctions. On est nécessairement amené, en vue des applications, comme on le voit dans [20], à considérer les différentes dégradations apportées à la fonction par les manipulations auxquelles l'image théorique est soumise :

Création par impression : dégradation due à l'appareil,

Enregistrement : dégradation due au support physique,

Utilisation : dégradation due à l'appareil d'analyse.

Ces dégradations, analysées dans Estournet [20], Huang et al [22] et Hunt et Breedlove [23] sont de trois types principaux :

filtrages passe-bas, qui diminuent la finesse de l'image,

superposition de bruits,

distortions, en général non linéaires.

Les conclusions qui nous intéressent peuvent être résumées ainsi :

(a) On dispose, en bout de chaîne, pour le traitement en ordinateur, d'une suite finie de nombres qui représentent l'image.

(b) Ces nombres sont attachés chacun à un domaine de discrétisation de l'image complète et obtenus par quantification de la valeur de l'intégrale :

$$f(x, y) = \iint_d g(x_1, y_1) h_d(x - x_1, y - y_1) dx_1 dy_1$$

où g est la distribution de l'intensité de la tâche définie sur le domaine d : $h_d(x, y) = 0$ si $(x, y) \notin d$.

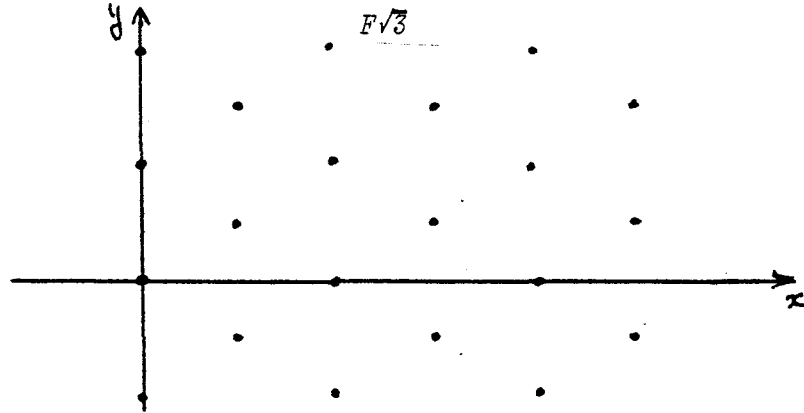
La fonction f est alors prise sur un ensemble fini de points (x_i, y_i) .

(c) L'analyse fréquentielle justifie l'emploi de taches circulaires puisqu'elles doivent avoir un centre de symétrie, la fonction g étant réelle.

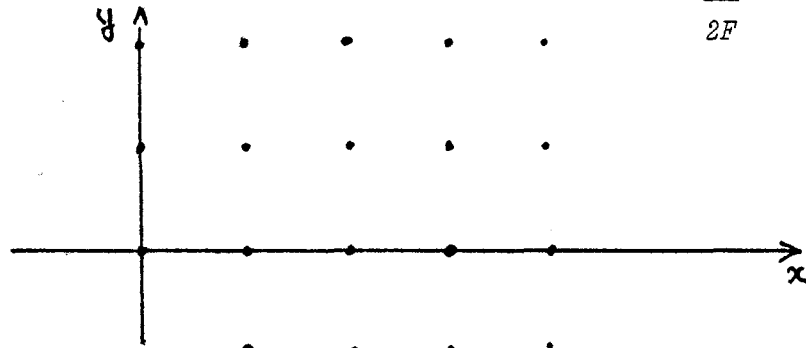
(d) De plus, sous les réserves habituelles d'applicabilité, on obtient des critères d'échantillonnage minimal régulier (cf. [20]) à partir du théorème de Shannon à deux dimensions.

Exemples

Une image isotrope vis-à-vis de ses caractéristiques fréquentielles et de fréquence limite F , a pour échantillonnage optimal, une brosse à mailles triangulaires équilatérales de côté $\frac{1}{F\sqrt{3}}$



Dans les mêmes conditions, cet échantillonnage peut être remplacé par une brosse orthogonale de côté $\frac{1}{2F}$



Remarque : Ces critères permettent de vérifier si un échantillonnage conserve l'information du message, mais ne permettent pas de construire des brosses quelconques.

Conséquences :

D'après (a), les traitements seront de nature algorithmique : opérations de nature algébrique, sur les nombres, syntaxiques sur des groupements en reconnaissance des formes p.e. Mais ces traitements peuvent tenir compte ou non du fait que deux nombres sont issus de domaines voisins.

Les transformations orthogonales, les lissages, en tiennent compte, des traitements statistiques aboutissant ou opérant sur des histogrammes ne le font pas.

(On pourra vérifier plus loin qu'il s'agit là d'applications de l'isovalence des lots).

On est donc amené, au moins pour les premiers, à essayer de formaliser, en vue de leur manipulation, ces notions de proximité ou de voisinage. On emploiera des relations binaires pour se rattacher à la classe des lots d'informations.

3.2. Image numérique quelconque

Voisinage de points.

Le voisinage de deux points est, au départ, une notion purement physique. Deux points de discrétisation sont voisins si les domaines correspondants sont jointifs. Comme deux domaines peuvent être sécants et, à la limite, superposés, un point est voisin de lui-même et si a est voisin de b , b est voisin de a . La relation de voisinage la plus générale est donc :

Relation de voisinage :

Dans un ensemble E , une relation binaire interne de voisinage est réflexive et symétrique. On la désignera par σ .

Remarque 1 : Elle n'est ni transitive, ni intransitive.

Remarque 2 : En prenant cette définition, on masque le phénomène essentiel dû au déplacement du faisceau optique ou électronique : un point a a un prédécesseur et un successeur.

C'est pourquoi on adoptera une définition restrictive :

Définition 1 :

Nous appellerons dans la suite relation de base une relation binaire ρ , définie dans un ensemble quelconque E : $\rho \subseteq E \times E$, telle que :

- 1) ρ soit une relation de voisinage
- 2) $a \in E$, si $\rho(a) = \{x \mid x \in E \text{ et } x\rho a\}$, $\text{card}(\rho(a)) \leq 3$

C'est-à-dire que a a pour voisins autres que lui-même, au plus deux éléments. Une image numérique quelconque peut alors être définie, indépendamment de toute qualité et sous son aspect topologique, c'est-à-dire à une déformation quelconque près, par :

Définition 2 : Image numérique quelconque.

Une image numérique est un lot d'informations $L = (I, R_I)$, réalisé sur un support (E, R_E) , où :

E : est un ensemble d'éléments appelés "points"

R_E : est un ensemble de relations de base,

V : est une échelle de quantification d'un signal pour une image noir et blanc ou monochrome, un vecteur pour une image en couleur.

Les images numériques réelles sont obtenues par des relations de base et une organisation de celles-ci particulières comme on peut le voir pour les peignes du § précédent.

C'est pourquoi après l'étude de leurs propriétés, on établira les structures de relations qui forment des maillages.

Remarque : La relation d'ordre peut paraître directement utilisable pour notre propos à la place de la relation de voisinage. On trouvera en remarque page 42 les objections à son emploi.

3.3. Propriétés des relations de base

On conservera par la suite le symbole ρ éventuellement indicé, pour désigner une relation de base, sans reprendre les notations $[E, \rho, E]$ ou $\rho \subseteq E \times E$. On notera :

$$\rho(a) = \{x \mid x \in E \text{ et } x \rho a\}, \text{ de même :}$$

$$\sigma(a) = \{x \mid x \in E \text{ et } x \sigma a\} \text{ pour les relations de voisinage.}$$

3.3.1 Composition des voisinages

Construisons pour σ et donc pour ρ , la loi \circ de composition des voisinages, définie par :

$$(\sigma \circ \sigma)(a) = \sigma(\sigma(a)) = \left\{ x \mid x \in E, \quad y \in \sigma(a) \quad \text{et} \quad x \sigma y \right\}$$

Notons $(\sigma \circ \sigma)(a) = \sigma^{(2)}(a)$ et par itération :

$$\sigma^{(n)}(a) = (\sigma \circ \sigma^{(n-1)})(a) = \sigma(\sigma^{(n-1)}(a)) \text{ pour tout } n \text{ entier positif}$$

Proposition 1 : Les suites $(\sigma^{(k)}(a))_k$ et $(\rho^{(k)}(a))_k$ de termes généraux $\sigma^{(k)}(a)$ et $\rho^{(k)}(a)$ respectivement, indicées par k sont chacune ordonnées par inclusion.

On établira la proposition pour σ ce qui la démontre pour ρ .

Montrons d'abord que $a \in \sigma^{(k)}(a)$ $k > 0$, $a \in \sigma(a)$, supposons que $a \in \sigma^{(k)}(a)$, alors, puisque $a \sigma a$ par réflexivité, $a \in \sigma^{(k+1)}(a)$.

Soit maintenant $x \in \sigma(a)$, alors $x \sigma a$ et donc $x \in \sigma^{(2)}(a)$, de même si $x \in \sigma^{(k)}(a)$, puisque $x \sigma a$, $a \sigma x$ et $x \in \sigma^{(k+1)}(a)$, q. e. d.

Si E est fini, les suites $(\sigma^{(n)}(a))_n$ et $(\rho^{(n)}(a))_n$ sont constantes au delà d'un certain rang, notons, dans ce cas $\hat{\sigma}(a) = \sup_k (\sigma^{(k)}(a))$ et $\hat{\rho}(a) = \sup_k (\rho^{(k)}(a))$

Montrons alors que si $x \in \hat{\sigma}(a)$, alors $a \in \hat{\sigma}(x)$.

Par définition, il existe alors une suite $x, x_1, x_2, \dots, x_n, a$ telle que $x \sigma x_1, x_1 \sigma x_2, \dots, x_n \sigma a$ et donc $a \in \sigma^{(n)}(x) \subseteq \hat{\sigma}(x)$.

On obtient alors directement que $\hat{\sigma}(a) = \hat{\sigma}(x)$.

Proposition 2 : Une relation de base, comme une relation de voisinage, induisent chacune une partition sur leur ensemble de définition, si celui-ci est fini.

Comme précédemment, nous démontrerons la proposition pour la seule relation de voisinage.

Il est clair que :

- $\forall a \in E$, $\hat{\sigma}(a) \neq \emptyset$ puisque $a \in \hat{\sigma}(a)$.
- l'ensemble des $\hat{\sigma}(a)$, $\forall a \in E$, forme un recouvrement de E.

Montrons que si $\hat{\sigma}(a) \neq \hat{\sigma}(b)$, alors $\hat{\sigma}(a) \cap \hat{\sigma}(b) = \emptyset$. S'il n'en était pas ainsi, il existerait $c \in \hat{\sigma}(a) \cap \hat{\sigma}(b)$, donc $a \in \hat{\sigma}(c)$ et $b \in \hat{\sigma}(c)$ comme on l'a vu précédemment, en particulier $\hat{\sigma}(a) = \hat{\sigma}(c) = \hat{\sigma}(b)$ ce qui est contraire à l'hypothèse.

On peut alors énoncer la définition :

3.3.2 Définition d'une ρ -chaîne

Une ρ -chaîne est un voisinage complet $\hat{\rho}(a)$ d'un point quelconque a dans un ensemble fini E. En d'autres termes, un élément de la partition induite par ρ .

Caractérisation des ρ -chaînes

Partant de a qui a, au plus, deux voisins, nommons les a_{-1} et a_{+1} , on peut écrire :

$$a_{-1} \rho a \rho a_{+1} \text{ ou encore, en omettant } \rho : a_{-1}, a, a_{+1},$$

Il en est de même pour a_{-1} et a_{+1} , et par itération, on construit l'écriture condensée d'une ρ -chaîne : $\dots a_{-2} a_{-1}, a a_{+1} a_{+2} \dots$

où $a_i \rho a_{i+1} \forall i$ tel que $a_i \in \hat{\rho}(a)$.

Proposition : Dans l'écriture sous forme de suite condensée $a_i \neq a_j \quad \forall i \neq j$, dans une même ρ -chaîne.

En effet, supposons que pour $i \neq j$, $a_i = a_j$, alors :

$\rho(a_i) = \rho(a_j)$ et $a_{i-1}, a_i, a_{i+1} = a_{j-1}, a_j, a_{j+1}$; alors :

soit $a_i = a_j$ a plus de deux voisins, ce qui est contraire à l'hypothèse, soit, les deux sous-suites sont égales. Par exemple :

$a_{i+1} = a_{j-1}$ et $a_{i-1} = a_{j+1}$.

D'après la construction de la ρ -chaîne il existe au moins un a_k commun aux deux sous-suites prolongées. Par exemple :

$\dots a_i a_{i+1} a_{i+2} \dots a_k \dots a_{j-1} a_j \dots$

Soit $a_{k-1} a_k a_{k+1}$ le ρ -voisinage de a_k , on a alors soit $a_k = a_{k+1}$ (ou $a_{k-1} = a_k$), soit $a_{k-1} = a_{k+1}$ d'après leurs constructions.

Dans le premier cas, la suite s'écrit :

| | | | | | |
|-----------|-----------|---------|-------|-----------|---------------------------|
| a_k | a_{k-1} | \dots | a_i | a_{i-1} | les deux sous-suites sont |
| | | | | | confondues |
| a_{k+1} | a_{k+2} | \dots | a_j | a_{j+1} | |

Dans le second cas :

| | | | | | |
|-----------|-----------|---------|-------|-----------|---------------------------|
| a_k | a_{k-1} | \dots | a_i | a_{i-1} | les deux sous-suites sont |
| | | | | | également confondues |
| a_{k+1} | a_{k+2} | \dots | a_j | a_{j+1} | |

La caractérisation est alors la suivante :

- i) si $\text{card } \rho(a) = 1$, la ρ -chaîne est constitué d'un élément isolé.
- ii) si $\exists a$ dans la ρ -chaîne, tel que $\text{card } \rho(a) = 2$, a est dit extrémité et $\exists b$ unique dans ρ -chaîne tel que $\text{card } \rho(b) = 2$.

En effet, soit $a_1 \in \rho(a)$, écrivons $\hat{\rho}(a)$:

$a a_1 a_2 \dots a_n \dots$

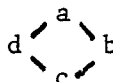
La ρ -chaîne est finie puisque E est fini, tous ses éléments sont distincts, donc il existe b tel que soit $\text{card } \rho(b) = 2$, soit $\rho(b) = b_{-1}, b, a$. Ce qui est contraire à l'hypothèse.

iii) Il existe des ρ -chaînes telles que \forall a leur appartenant :

$$\text{card } \rho(a) = 3.$$

Vérifions le en construisons sur $E = \{a, b, c, d\}$.

la relation $\rho = \{(a, b), (b, c), (c, d), (d, a)\}$

 est une ρ -chaîne satisfaisant à l'énoncé.

Le cas ii) définit une ρ -chaîne ouverte,

Le cas iii) définit une boucle.

3.3.3 Caractérisation des boucles et des chaînes ouvertes par les couplages.

Un couplage (cf. [24]) est défini de la façon suivante :

Soit une relation binaire $\rho \subset E_1 \times E_2$, soit $E'_1 \subseteq E_1$.

Une injection $C : E'_1 \rightarrow E_2$ est un couplage si et seulement si, pour tout $e \in E'_1$, $(e, C(e)) \in \rho$.

Si $E_1 = E_2 = E$ le couplage est dit autocouplage.

Si, de plus, on impose à un autocouplage de vérifier que pour

$\forall e_1, e_2$ de $E' \subset E$, $(e_1, e_2) \in \rho \quad (C(e_1), C(e_2)) \in \rho$

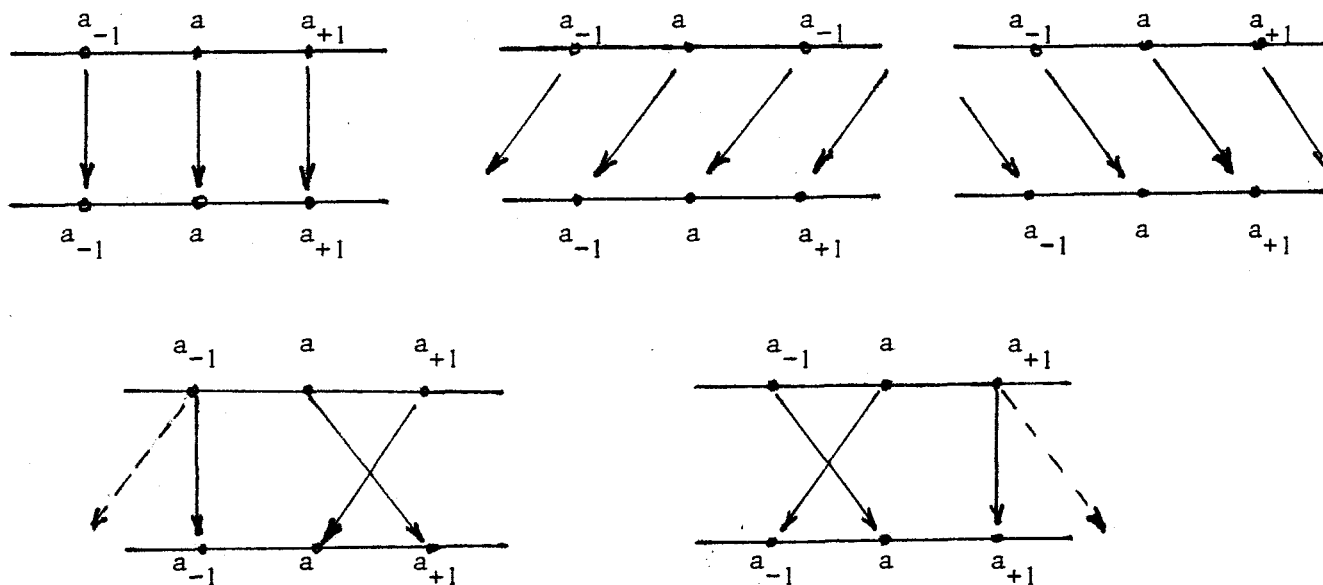
l'autocouplage est un endomorphisme.

Application aux relations de base

Considérons une relation de base ρ sur un ensemble E puisqu'elle induit une partition sur E , un autocouplage appliquera nécessairement un élément de la partition dans lui-même.

Soit $\hat{\rho}(a)$ un tel élément, considérons les autocouplages de

$\{a_{-1}, a, a_{+1}\} = E'$ dans E . Leur construction systématique nous fournit les cinq schémas suivants où les flèches pleines représentent l'injection et les flèches tiretées les variantes possibles :



Seuls C_1 , (l'injection canonique), C_2 et C_3 sont des endomorphismes,

puisque :

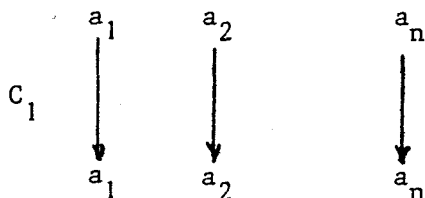
$$(C_4(a_{-1}), C_4(a)) \notin \rho \quad \text{et} \quad (C_5(a), C_5(a_{+1})) \notin \rho.$$

Etendant les autocouplages à toute la chaîne, c'est à dire prenant

$E' = \hat{\rho}(a)$, l'endomorphisme, s'il existe sera un isomorphisme.

On obtient :

1) pour une ρ -chaîne ouverte



puisque $C_2(a_1)$ et $C_3(a_n)$ n'existent pas.

2) pour une boucle les trois autocouplages C_1, C_2, C_3 .

On peut donc énoncer :

Une ρ -chaîne est une boucle si et seulement si il existe sur elle deux autocouplages isomorphismes en plus de la bijection canonique.

Terminologie

On emploiera, dans la suite, les désignations suivantes : pour une relation de base ρ définie sur E :

- Une relation ρ est complète si $\rho(a) = E$, cette $\hat{\rho}$ -chaîne unique couvre E .
- Une section S de ρ , $S \subseteq E$ est formée par un élément et un seul pris dans chaque ρ -chaîne.
- Une sous-section est un sous-ensemble d'une section.
- Une section initiale est une section dont les éléments extraits des ρ -chaînes ouvertes sont une extrémité de chacune.
- Une section initiale réduite est une sous-section initiale formée seulement d'extrémités de ρ -chaînes ouvertes.
- Une section complète est l'union sur les chaînes ouvertes de toutes les sections initiales.
- Une sous-chaîne est un sous-ensemble d'une chaîne formant chaîne.

Remarques : une sous-chaîne de boucle est, soit la boucle, soit une chaîne ouverte.

une sous-chaîne de chaîne ouverte est ouverte.

Le type de relations que nous venons d'étudier a, vis à vis du problème qui nous intéresse plusieurs avantages qui n'apparaissent pas nécessairement dans leurs propriétés.

Tout d'abord, comme on l'a dit plus haut, on se rattache directement aux lots d'informations, puisque l'on utilise des relations binaires.

Mais pourquoi ces relations de base et non des relations classiquement connues : ordres, préordres ou équivalences ?

On se rappelle que le phénomène de base est la notion d'alignement (sur une courbe quelconque) de points, suivant la progression du faisceau, ceci exclut les deux dernières relations. Quant à la relation d'ordre, on peut y opposer une raison d'une autre nature. Elle exprime explicitement ou implicitement, non seulement un alignement, mais aussi une succession temporelle ou causale. Si cette dernière existe, elle

n'est que le fait de la dynamique du processus de lecture, donc d'un moyen et n'exprime pas une propriété intrinsèque à l'image. En effet, comme le note Estournet [20] : le principe de causalité, contrainte physique dans l'espace monodimensionnel temporel n'a pas d'équivalent dans l'espace bidimensionnel [des images].

On verra plus loin des emplois immédiats de cette propriété : raccordements d'images p.e., qui s'expriment plus simplement au moyen de relations de base que de relations d'ordre. Quand un ordre sera nécessaire il pourra être introduit (§3.4) sur les relations, ou (§4.4) sur les chaînes.

3.4. Structures algébriques sur les relations de base

Cette étude nous sera utile pour exprimer les liens entre plusieurs explorations distinctes faites par le faisceau de lecture sur une même image, c'est à dire les liens entre ρ -chaînes.

3.4.1 Préordre sur les relations de base

Relation de contenance

Définition : Soient E et E' deux ensembles sur lesquels ont été définis deux relations de base ρ et ρ' respectivement. On dira que E contient E' s'il existe un homomorphisme $E' \rightarrow E$ tel que toute ρ' -chaîne de E' est transformée en une sous chaîne de E :

On notera $E' \leq E$.

Remarque : d'après les résultats du § précédent, l'homomorphe d'une boucle de E' est une boucle de E .

Propriétés de la relation de contenance

- 1) Elle est réflexive par bijection canonique.
- 2) Si E_1, E_2, E_3 sont trois ensembles tels que $E_1 \leq E_2 \leq E_3$, alors $E_1 \leq E_3$ par composition des homomorphismes, la relation est donc transitive sur une famille d'ensembles, d'où l'énoncé :

La relation de contenance est un préordre.

Les classes d'équivalence de ce préordre sont composées par des sous familles de la familles d'ensembles munies de relations de base isomorphes.

- 3) Considérons maintenant une famille d'ensembles constituée par la répétition d'un même ensemble E ($E_1 = E_2 = \dots = E$). Soit une famille de relations binaires ρ_i indicées par i .

La relation \leq entre ensembles s'applique maintenant aux relations ρ_i avec les mêmes propriétés de préordre : réflexivité et transitivité.

On peut donc écrire $\rho_i \leq \rho_j$ p.e., sous entendant :

$$E \text{ muni de } \rho_i \leq E \text{ muni de } \rho_j.$$

Les classes d'équivalences sont composées des sous ensembles de l'ensemble des relations de base, tels qu'il existe un automorphisme de E pour tout couple (ρ_i, ρ_k) d'éléments de la classe.

Ceci nous permet d'énoncer un théorème relatif aux lots :

Théorème

Un lot (I, R_I) formant image numérique est défini sur son support à une bijection b près : $b : R_E \longrightarrow R'_E$ où ρ et $b(\rho)$ appartiennent à une même classe d'équivalence de contenance.

3.4.2 Composition externe des relations de base

Soient E_1 et E_2 deux ensembles disjoints munis des relations de base ρ_1 et ρ_2 respectivement :

On appellera ensemble composé E , à relation de base ρ l'ensemble :

$E = E_1 \cup E_2$, muni de :

$E \times E$ tel que $\rho = \rho_1$ dans $E_1 \times E_1$ et

$\rho = \rho_2$ dans $E_2 \times E_2$.

E sera un ensemble raccordé si :

- 1) E est composé
- 2) $\exists a \in E_1$ et $b \in E_2$ tel que $a \rho b$.

Ceci implique nécessairement que a et b appartiennent respectivement à une section initiale réduite de E_1 et E_2 : S_1 et S_2 .

Il est alors évident que E ne peut être raccordé que si E_1 et E_2 contiennent chacun au moins une boucle ouverte.

Un raccordement complet consiste en la construction de tous les couples (a, b) $a \in S_1$ et $b \in S_2$ compatibles avec la relation de base.

On a ainsi défini de façon générale la juxtaposition de deux images par le raccordement de leurs supports. Le résultat du raccordement est une image qui contient toutes les chaînes des images de départ.

On verra plus loin, page 65 la décomposition d'une image en sous images.

Remarque : Le raccordement n'agit pas sur les boucles.

3.4.3 Composition interne de relations de base

La somme de deux relations de base n'est pas, en général, une opération de composition interne, puisqu'on ne conservera pas la pro-

propriété $\text{card } \rho(a) \leq 3$.

Le produit ou intersection est une loi de composition interne, puisque si $\rho_1 \cap \rho_2 = \rho$, $\forall a \in E$.

$$\text{card } \rho(a) = \min (\text{card } \rho_1(a), \text{card } \rho_2(a)) \leq 3.$$

structure engendrée par l'intersection.

Soit E l'ensemble sur lequel on opérera.

- 1) $\rho \cap \rho = \rho$ donc l'intersection est idempotente
- 2) l'opération est associative et commutative puisqu'elle opère sur des sous-ensembles de $E \times E$.
- 3) la relation de contenance entre relations de base sur un même ensemble est telle que :

$$\rho_1 \leq \rho_2 \implies \rho_1 \cap \rho_2 = \rho_1 \text{ d'après la définition de la contenance.}$$

L'ensemble des relations de base sur un même ensemble a donc pour l'intersection une structure d'inf-demi-treillis.

Sa borne inférieure est ρ_0 telle que $\forall a \in E \rho_0(a) = \{a\}$.

Rappelant que deux relations ρ_1 et ρ_2 sont \cap -complémentaires si et seulement si $\rho_1 \cap \rho_2 = \rho_0$, on établit aisément que l'ensemble des relations \cap -complémentaires à une relation donnée forme un sous-inf-demi-treillis.

En effet, les propriétés d'inf-demi-treillis sont conservées et le sous-ensemble est fermé :

Si $\rho_1 \cap \rho = \rho_0$ et $\rho_2 \cap \rho = \rho_0$, alors :

$$\rho \cap (\rho_1 \cap \rho_2) = \rho_0.$$

3.5. Conclusion

On a défini une image numérique quelconque comme un lot d'informations particulier.

Sa structure algébrique est obtenue par le moyen de relations binaires nouvelles qui formalisent la notion d'exploitation de l'image par

le capteur de discrétisation.

Cette définition est topologique uniquement, puisque la répartition des points d'échantillonnage sur le support physique est indépendante de toute déformation. On pense ainsi avoir échappé au préalable souvent posé d'une définition métrique : explicitement pour des points à coordonnées entières (Z^n) ou implicitement comme dans [21] et [25] où l'image est repérée par un groupe abélien fini de points ou un monoïde dont les éléments de base représentatifs sont des vecteurs libres.

Mais la répartition des points d'échantillonnage est rarement quelconque. Nous allons montrer dans le chapitre suivant que notre définition est opératoire, puisque l'on retrouvera, en précisant l'ensemble des relations de base employé, les schémas de discrétisation classiques.

La généralité de l'outil employé, le lot d'information, nous permettra, après avoir rattaché les systèmes cellulaires à notre développement, d'établir des schémas communs de traitement.

CHAPITRE 4

IMAGES REGULIERES ET SYSTEMES CELLULAIRES

DIMENSION DES ESPACES DISCRETS

4.1. Introduction

Les structures présentées dans le chapitre précédent ne peuvent pas être qualifiées de "régulières". Elles autorisent des configurations où figurent : chaînes de longueurs quelconques, boucles et points isolés. Or, la discrétisation des images est faite au moyen d'explorations systématiques du support, à la fois pour des raisons technologiques de mise en oeuvre du capteur, et aussi pour se placer à l'intérieur du cadre du deuxième théorème de Shannon.

Nous avons vu également que les réalisations sur une mémoire de support d'une structure de lot quelconque sont particulièrement complexes, vu le grand nombre d'index nécessaires.

C'est pourquoi nous devons préciser une classe particulière de familles de relations de base, dont on vérifiera ultérieurement :

- a) qu'elle contient les structures couramment admises (§4.4),
- b) qu'elle est réalisable sans grandes difficultés sur une mémoire (chapitre 5),
- c) qu'elle possède une propriété d'invariant topologique : la dimension d'espace discret.

Cette dimension sera ensuite étendue aux structures algébriques quelconques et donc aux systèmes cellulaires orientés usuels.

On appellera maillages ces familles de relations de base.

Il existe des liens étroits entre certains de ces maillages et la théorie des cartes : polyèdres réguliers de Erréra [26], cartes régulières d'Altschuler [27]. Ce qui diffère essentiellement est le point de vue adopté : les cartes sont des décompositions cellulaires imposées par un graphe ; l'étude porte sur les objets du graphe : nombres de sommets, de côtés et de cellules (faces de la carte). On cherche à priori les cartes régulières, c'est à dire celles qui peuvent être construites avec des cellules de même configuration (mêmes nombres de sommets et de côtés).

Si l'on représente les relations de base par des arcs joignant les éléments de E , eux mêmes représentés par des points, on peut dire que nous nous intéressons aux modes de construction de ces graphes. Les cartes régulières évoquées plus haut sont alors des cas particuliers de ces constructions.

4.2. Relations orthogonales et connexité

Une exploration systématique par un capteur :

- a) ne laissera pas des points isolés,
- b) si elle comporte des boucles, ne comportera que des boucles (balayage circulaires) et réciproquement, si elle comporte des chaînes, on ne trouvera que des chaînes (balayages linéaires). Dans toute la suite on ne considèrera que des chaînes ouvertes.
- c) il y aura une relation de voisinage répétitive entre points déjà explorés. On la composera au moyen d'autres relations de base, pour conserver l'unité de la formulation. Ces autres rela-

tions de base correspondront à des programmes d'exploration différents, que l'on aurait pu utiliser pour la discrétisation de l'image.

Pour celà, nous utiliserons la seule loi de composition interne des relations sous la forme d'un cas particulier de l' n -complémentarité.

4.2.1 Orthogonalité de deux relations de base ρ_1 et ρ_2

ρ_1 et ρ_2 construites sur un même ensemble E sont orthogonales si et seulement si :

- i) elles sont n -complémentaires ($\rho_1 \cap \rho_2 = \rho_0$).
- ii) toute ρ_1 -chaîne (resp ρ_2 -chaîne) est une sous-section de ρ_2 (resp ρ_1).

On notera $\rho_1 \perp \rho_2$

L'ensemble des relations est évidemment fermé par intersection : si $\rho \perp \rho_1$ et $\rho \perp \rho_2$, alors $\rho \perp (\rho_1 \cap \rho_2)$ par intersection de sous-sections.

Elles forment donc un sous-inf-demi-treillis.

Remarque : Tout couple de relations n -complémentaires peut être transformé en relations orthogonales par décomposition des chaînes de l'une en sous-sections de l'autre.

Exemples

*Utilisons la représentation sous forme de graphes.
Les figures suivantes représentent :*

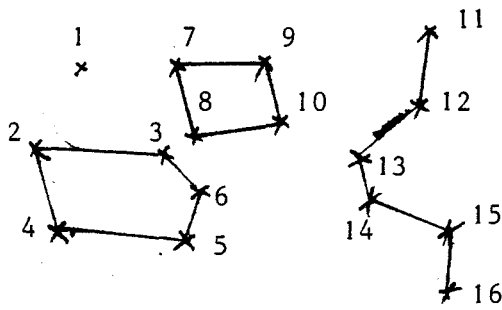


fig.1. une relation de base sur un ensemble de cardinal 16, comportant un point isolé, une chaîne ouverte et deux boucles.

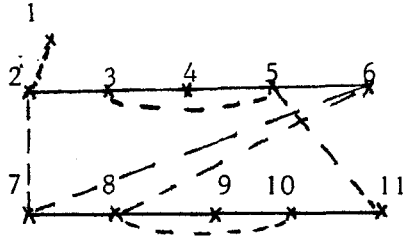


fig.2. deux relations de base n -complémentaires, sur E de cardinal 11.

ρ_1 représentée par des traits pleins a un point isolé : (1) et deux ρ -chaînes : (2, 3, 4, 5, 6) et (7, 8, 9, 10, 11.)

ρ_2 représentée par des traits tiretés a deux points isolés (4) et (9) et deux ρ -chaînes : (1, 2, 7, 6, 8, 10) et (3, 5, 11.)

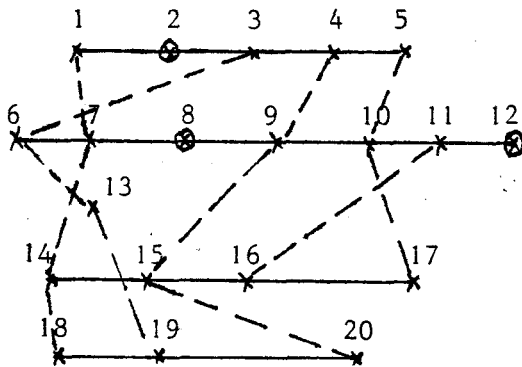


fig.3. sur E de cardinal 20.

ρ_1 en traits pleins a un point isolé 13 et quatre chaînes : (1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12), (14, 15, 16, 17) et (18, 19, 20).

ρ_2 en tiretés a trois points isolés : (2), (8) et 12 et cinq chaînes : (1, 7, 14, 18), (3, 6, 13, 19), (4, 9, 15, 20), (5, 10, 17) et (11, 16).

Les deux relations sont orthogonales.

4.2.2 Pas, chemins et connexité

On utilise, ici, les définitions usuelles :

Soit E muni de relations de base $\rho_1, \rho_2, \dots, \rho_n$, on dira qu'il y a un pas entre deux éléments de E : a et b s'il existe une relation au moins : ρ_i telle que $(a, b) \in \rho_i$.

Entre deux éléments a et $b \in E$, il existe un chemin s'il existe une suite $a, x_1, x_2, \dots, x_p, b$ et une suite de relations.

$\rho_{i_1}, \rho_{i_2}, \dots, \rho_{i_{p+1}}$ telles que

$$(a, x_1) \in \rho_{i_1}, (x_1, x_2) \in \rho_{i_2}, \dots (x_p, b) \in \rho_{i_{p+1}}$$

L'ensemble E sera dit connexe s'il existe un chemin entre tout couple de ses éléments

Première condition suffisante de connexité

Soit E muni de deux relations de base ρ_1 et ρ_2 . E sera connexe si l'une des relations (ρ_2 par exemple), a une ρ -chaîne ρ_s section complète de ρ_1 .

En effet, soient a et b \in E, il existe $\alpha \in \hat{\rho}_1(a)$ et $\alpha \in \rho_s$ et $\beta \in \hat{\rho}_1(b)$ et $\beta \in \rho_s$, donc un chemin sera constitué par les sous-chaînes : a, ..., α de ρ_1 , α , ..., β de ρ_2 , β ... b de ρ_1 .

Deuxième condition suffisante de connexité

Avec les mêmes notations pour E, ρ_1 et ρ_2 que précédemment, il suffit que pour tout couple de ρ_1 -chaînes (resp ρ_2 -chaînes) :

$\hat{\rho}_1(a)$ et $\hat{\rho}_1(b)$ distinctes, il existe $\alpha \in \hat{\rho}_1(a)$ et $\beta \in \hat{\rho}_1(b)$ tels que $(\alpha, \beta) \in \rho_2$.

La démonstration est du même type que la précédente.

Condition nécessaire et suffisante de connexité dans l'orthogonalité.

Supposons maintenant que ρ_1 et ρ_2 définis sur E soient orthogonales : $\rho_1 \perp \rho_2$. Construisons les deux ensembles de chaînes distinctes :

$$E_1 = \{ \hat{\rho}_1(a) \mid a \in E \} \text{ et } E_2 = \{ \hat{\rho}_2(a) \mid a \in E \}.$$

Soient e_{1_i} et e_{2_j} deux éléments quelconques de E_1 et E_2 , construisons la relation binaire $r \subset E_1 \times E_2$ définie par $(e_1, e_2) \in r$ les deux chaînes ρ -chaînes $\hat{\rho}_1(a) = e_1$ et $\hat{\rho}_2(b) = e_2$ ont un élément de E commun : $\hat{\rho}_1(a) \cap \hat{\rho}_2(b) = \{ \alpha \} \alpha \in E$.

La condition cherchée est :

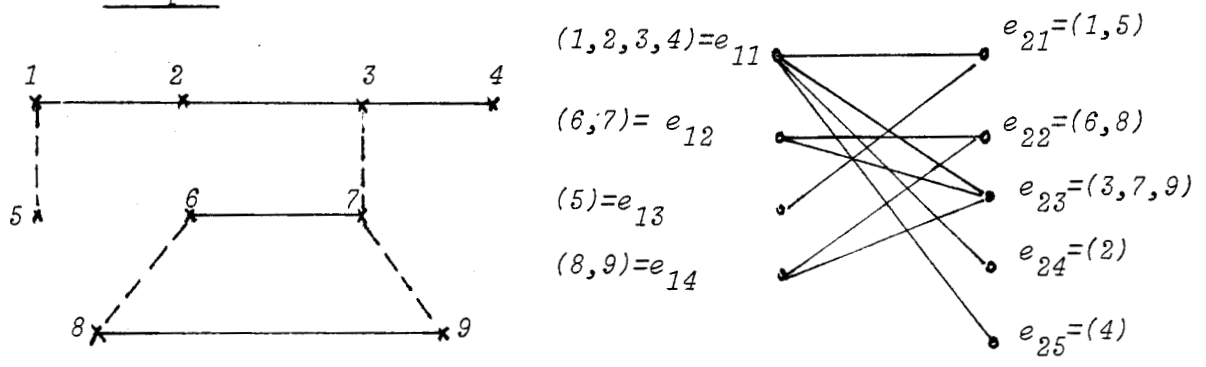
L'ensemble E est connexe par ρ_1 et ρ_2 si et seulement si l'ensemble $E_1 \cup E_2$ est connexe par r .

Supposons que $E_1 \cup E_2$ est connexe par r , alors, pour tout couple (a, b) d'éléments de E , il existe, d'une part, $\hat{\rho}_1(a)$ et $\hat{\rho}_2(b)$ éléments de E_1 . Si $\hat{\rho}_1(a) = \hat{\rho}_1(b)$ la propriété est immédiate, s'ils sont distincts, il existe un chemin :

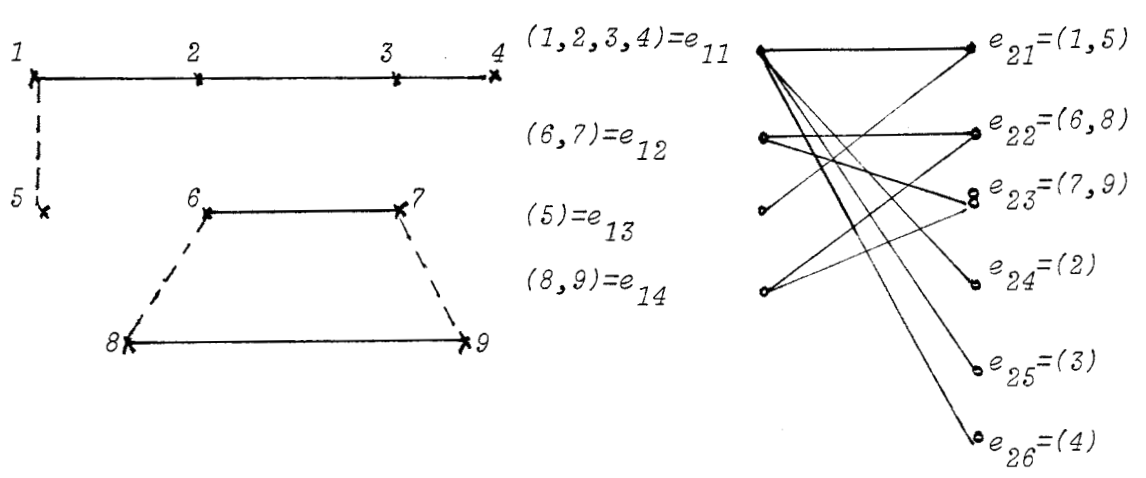
$e_{11} = \hat{\rho}_1(a), e_{21}, e_{12}, e_{22}, e_{13}, \dots, e_{1n} = \hat{\rho}_1(b)$ par la connexité de $E_1 \cup E_2$, et donc un chemin, concaténation des chemins partiels à construire sur les e_{i_j} qui joint a à b .

La réciproque est immédiate, puisque la connexité de E postule l'existence du chemin de a à b , donc la suite d'éléments de $E_1 \cup E_2$.

Exemple



Si l'on supprime l'élément $(3, 7) \in \rho_2$, les deux graphes perdent leur connexité



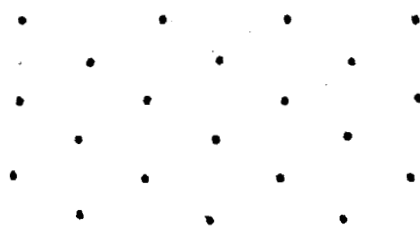
4.3. Dimension d'un espace discret

La notion de dimension d'un espace est essentielle pour établir ses propriétés topologiques. Toutefois, on constate que la discrétisation d'un espace rend caduques bien des définitions classiques dans le domaine continu et spécialement la notion de dimension. (Cf; notamment Rosenfeld [2]). Pour pallier ces difficultés, Mylopoulos et Pavlidis ont introduit dans [21] une notion de dimension d'un espace discret particulier construit sur un groupe abélien fini.

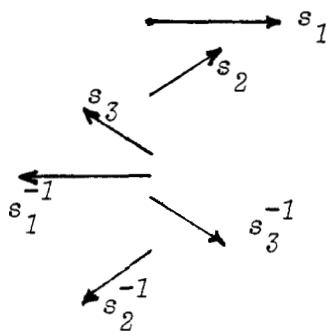
Rappelons l'essentiel de leur démarche ; pour ces auteurs :

Un espace discret est un groupe abélien fini à $2n$ générateurs : $s_1, s_2, \dots, s_n, s_1^{-1}, s_2^{-1}, \dots, s_n^{-1}$ auquel s'ajoutent des relations supplémentaires autres que la commutativité et l'inversion :

Exemple :



est engendré par :



On remarque que les groupes sans relations supplémentaires définissent les espaces euclidiens discrets classiques (Z^n).

La nécessité d'une définition directe de la dimension de tels espaces discrets apparaît aux auteurs du fait des contradictions entre les propriétés topologiques observables et la notion algébrique de dimension d'un groupe.

Exemple tiré de [21]) :

Le groupe $G_{1,2} = (s_1, s_2, s_1^{-1}, s_2^{-1} / s_1^9 s_2^{-10} = 1)$ a pour dimension algébrique 1. Or, le chemin (défini à partir d'un point quelconque par un mot du groupe)

$s_2 s_1^{-1} s_1^{-1} s_2^{-1} s_2^{-1} s_1 s_1 s_2 = 1$ "entoure" le point de départ et fait que $G_{1,2}$ ressemble topologiquement à un espace discret à deux dimensions.

Utilisant les définitions suivantes :

- le voisinage direct d'un point $x \in G$, $ND(x)$ est tout point $y \in G$ tel que $y = xg$ pour tout g générateur du groupe.
- le voisinage d'un point $N(x)$ est constitué par l'union de $ND(x)$ et de l'ensemble des points z tels qu'il existe y_1 et $y_2 \in ND(x)$ tels que le plus court chemin de y_1 à y_2 ne passant pas par x passe par z .
- $N^*(x) = N(x) - \{x\}$.
- un sous-ensemble F non vide de G est totalement non connexe si et seulement si $(ND(x) - \{x\}) \cap F = \emptyset$ pour tout $x \in F$, la famille de ces sous-ensembles est notée \mathcal{F}_0 .
- un sous-ensemble F non vide de G est linéairement connexe si et seulement si :

$$\text{card} [(ND(x) - \{x\}) \cap F] \leq 2 \text{ pour tout } x \in F \text{ et}$$

$$\text{card} [(ND(x) - \{x\}) \cap F] > 0 \text{ pour un } x \text{ au moins de } F.$$

La famille de ces sous-ensembles est notée \mathcal{F}_1 .

Les auteurs posent la définition suivante :

La dimension $\dim(F)$ d'un sous-ensemble arbitraire (an arbitrary set) $F \subseteq G$ est définie récursivement par :

$$\dim(F) = -1 \quad \text{pour} \quad F = \emptyset$$

$$\dim(F) = 0 \quad \text{pour} \quad F \in \mathcal{F}_0$$

$$\dim(F) = 1 \quad \text{pour} \quad F \in \mathcal{F}_1$$

$$\dim(F) = \max_{x \in F} [\dim(N^*(x) \cap F)] + 1, \text{ sinon}$$

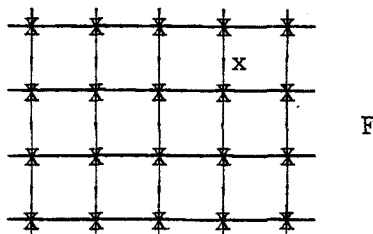
La cohérence interne de la définition est ensuite vérifiée et, par application aux groupes abéliens finis libres, les auteurs montrent que leur dimension en tant qu'espaces discrets est égale au nombre de leurs générateurs.

L'étude de cette définition nous a conduit à observer qu'elle donne des résultats incorrects :

si l'on prend pour sous-ensemble F un sous-ensemble non "compact" ou lacunaire et donc pour un ensemble structuré arbitrairement.

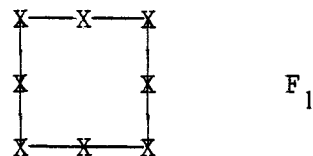
Vérifions tout d'abord la définition sur un exemple simple pour éclairer la démarche.

Soit F la partie figurée
ci-contre de G_2 :

$$\dim F = \max_{x \in F} [\dim (N^*(x) \cap F)] + 1$$


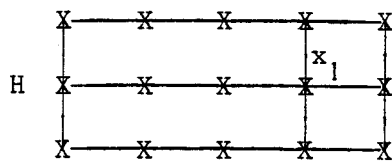
Soit x figuré sur la figure.

Posons $F_1 = N^*(x) \cap F$
 $\dim F_1 = 1$ puisque $F_1 \in \mathcal{H}_1$
 donc $\dim(F) = 2$.



Construisons maintenant un contre-exemple qui pourra être considéré à la fois comme une structure quelconque et comme un sous-ensemble de F et donc de G_2 , devant donc avoir une dimension inférieure ou égale à celle de F

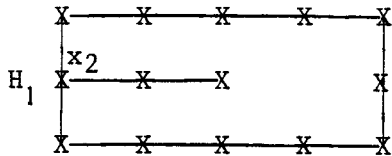
Contre-exemple : Soit H la structure ci-dessous extraite de F.



$$\dim(H) = \max_{x \in H} \dim(N^*(x) \cap H) + 1$$

$$\geq \dim(N^*(x_1) \cap H) + 1$$

$$\text{Nommons } H_1 = N^*(x_1) \cap H$$

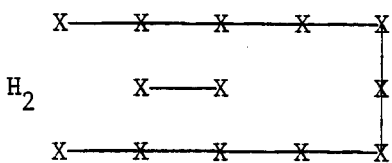


$$\dim(H_1) = \max_{x \in H_1} \dim(N^*(x) \cap H_1) + 1$$

$$\geq \dim(N^*(x_2) \cap H_1) + 1$$

Or, $N^*(x_2) \cap H_1 = H_2$ est linéairement

connexe comme on le voit ci-contre :



$$\text{donc : } \dim(H_2) = 1$$

$$\text{et } \dim(H) \geq 3. > \dim(F)$$

$$\text{Plus précisément } \dim(H) = 3$$

On perd, dans ce cas, la propriété essentielle de monotonie de la dimension qui veut qu'un sous-espace ait une dimension inférieure ou égale à celle de l'espace qui le contient.

Remarque : Nous ne sommes pas ici dans le cas d'une condition suffisante établie par Mylopoulos et Pavlidis : théorème 1 de [21]

En effet :

Prenant pour définition de la conservation des voisinages l'énoncé suivant :

Soient F_1 et $F_2 \subset G$. f , application de F_1 dans F_2 conserve les voisinages si :

$$\forall x \in F_1, \forall y \in ND(x) \text{ (resp } N(x) \text{ implique que si } \\ f(x) \text{ et } f(y) \text{ existent, alors } f(y) \in ND(f(x)) \\ \text{(resp } N(f(x))).$$

Les auteurs démontrent le théorème.

Théorème 1 :

Soit f une surjection d'un sous-ensemble de F_1 sur F_2 telle que f et f^{-1} conservent les voisinages, alors $\dim(F_1) \geq \dim F_2$.

Ici $f : F \longrightarrow H$ est une surjection qui ne conserve pas les voisinages de x'_1 par exemple.

Pour conserver l'intérêt de cette définition de la dimension dans les cas où elle s'applique, nous en proposons une extension valable pour des structures lacunaires (telle que H cidessus), pour des ensembles à structure algébrique formée de relations de base orthogonales deux à deux .

4.3.1 Dimension d'un espace, partie d'un groupe libre abélien fini.Structure lacunaire

Pour la définir, nous devons distinguer les éléments x , appelés points, des mots du groupe, suivant en cela la méthodologie employée dans les lots.

Sous-ensemble compact d'un groupe G .

Il est formé d'un sous-ensemble F de points de G , tel que $\forall x$ et $y \in F$ s'il existe g_i générateur de g avec $y = xg_i$ dans G , alors $y = xg_i$ dans F , sinon le sous-ensemble sera dit lacunaire.

Exemple : H est un sous-ensemble lacunaire de F . page 57.

Un corollaire immédiat du théorème 1 est que s'il existe une surjection $f : F_1 \longrightarrow F_2$ telle que f et f^{-1} conservent les voisinages, alors $\dim(F_1) = \dim(F_2)$.

Dimension d'un sous-ensemble lacunaire

Nous avons vu que l'application directe de la définition de la dimension de [21] peut conduire à $H \subset F$, $\dim(H) > \dim(F)$.

Remarque : La récursivité de la définition opère, pour x fixé dans le seul voisinage $N^*(x) \subseteq F$.

Soit x_1 le point de F tel que $\dim(N^*(x_1) \cap F)$ soit maximum, alors la suite x_2, x_3, \dots, x_N telle que :

$$N^*(x_1) \cap F = F_1 \text{ et } \dim(N^*(x_2) \cap F_1) \text{ max.}$$

$$N^*(x_{n-2}) \cap F = F_{n-1} \text{ et } \dim(N^*(x_n) \cap F) \text{ max}$$

a la propriété $x_i \in N^*(x_1) \quad i = 2, 3, \dots, N$

Appliquant la proposition 4 de (1) :

Pour tout $x \in G_n$ groupe abélien libre à n générateurs

$$N(x) = \left\{ x g_i g_j, x_{g_i} \mid g_i, g_j \text{ générateurs et } g_i \neq g_j \right\}$$

On en déduit :

Lemme 1 : Pour tout $\forall_x \in F$ partie compacte de G_n ,

$\forall_y \notin ND^*(x)$ et $y \in N^*(x)$, est sur un chemin de longueur 2, joignant deux points x_1 et x_2 de $ND^*(x)$.

En effet : soit $y = x g_i g_j$, $i \neq j$, il est sur le chemin $x_1 = x g_i$, $y = x_1 g_j$, $x_2 = y g_i^{-1} = x g_j$.

Définissons alors un voisinage fictif $\overline{N^*(x)}$ dans une partie lacunaire F de G_n et notons par l'indice F ou G les parties de F ou G prises dans F et G respectivement.

$\overline{N^*(x)}_F$ sera défini ainsi :

- 1) On ajoute dans $ND^*(x)_F$ les liaisons éventuellement manquantes entre deux points de F , on obtient ainsi $ND^*(x)_F$, l'ensemble des points n'est pas modifié.
- 2) On construit :

$$\overline{N^*(x)}_F = \overline{ND^*(x)}_F \cup Y_F \cup Z_F \text{ où :}$$

$$Y_F = \left\{ y \in F \mid y = x g_i g_j \text{ dans } F \right\} \text{ et ,}$$

Z est un sous-ensemble de points z fictifs, constituant entre deux points de $\overline{ND^*(x)}_F$ des chemins de longueur 2 qui remplacent, s'ils existaient des chemins minimaux de longueur > 2 ne passant pas par x, entre les deux mêmes points.

Exemples :

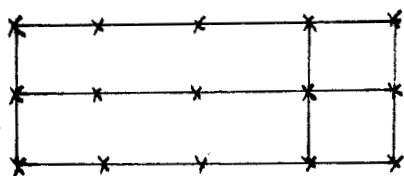
Soit F_1 une partie compacte de G_3



$\overline{ND^*(x)}_F$ devient



De même avec



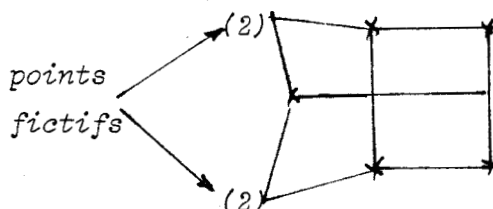
et F une partie lacunaire



et $\overline{N^*(x)}$



où $\overline{N^*(x)}$ devient



Soit maintenant un chemin minimal C ne passant pas par x de longueur supérieure à 2, entre deux points x_1 et x_2 de $\overline{ND^*(x)}_F$, chemin construit dans F partie lacunaire de G_n , munie de $\overline{ND^*(x)}_F$

Lemme 2 : Ce chemin ne peut être issu de la suppression d'un chemin de longueur 1 :

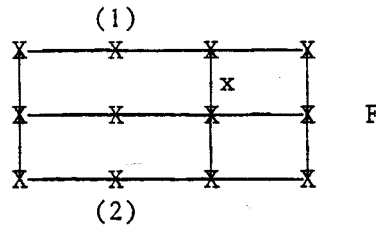
puisque ceux-ci ont été rétablis par le point 1) de la construction de $\overline{ND^*(x)}_F$

Lemme 3 : Tout chemin C est issu de la suppression d'un chemin minimal de longueur 2 entre les mêmes points pris dans G_n .

En effet, s'il n'en était pas ainsi, C serait minimal dans G_n ce qui est impossible d'après le lemme 1.

Corollaire : L'existence de C dans F muni de $\overline{ND^*(x)}$ provient de la suppression dans $\overline{N^*(x)}_F$ d'un y pris en tant qu'élément de $\overline{N^*(x)}_G$ (et non nécessairement en tant que point de F)

Exemple :



(1) et (2) ne figurent plus dans F en tant qu'éléments de $\overline{N^*(x)}_F$, alors qu'ils faisaient partie de $\overline{N^*(x)}_G$.

Théorème 2 :

Pour tout $x \in F$, partie lacunaire de G_n , il existe une surjection $f : \overline{N^*(x)}_G \longrightarrow \overline{N^*(x)}_F$ telle que f et f^{-1} conservent les voisinages.

Démonstration :

La surjection existe d'après la définition de Z et le lemme 3.

f et f^{-1} conservent les voisinages d'après la construction de $\overline{N^*(x)}_F = \overline{ND^*(x)}_F + Y_F + Z$ puisque la propriété est vraie pour $\overline{ND^*(x)}$, pour Y_F et pour Z d'après le lemme 3.

Nous pouvons donc appliquer le théorème (1) ci-dessus à :

$F_1 = G_n$ et $F_2 = F$ muni de $\overline{N^*(x)}_F$:

$\dim (F \text{ muni de } \overline{N^*(x)}_F) \leq \dim G_n$

En particulier, il suffira de construire $\overline{N^*(x)}$ au seul point x de F tel que $\dim(\overline{N^*(x)} \cap F)$ max d'après la Remarque 1.

Conclusion :

Un espace discret F, sous-ensemble quelconque d'un groupe abélien fini libre G_n : $F \subseteq G_n$ a pour dimension :

$$\begin{aligned} \dim(F) &= -1 & \text{si } F &= \emptyset \\ \dim(F) &= 0 & \text{si } F &\in \mathcal{F}_0 \\ \dim(F) &= 1 & \text{si } F &\in \mathcal{F}_1 \\ \dim(F) &= \max_{x \in F} \dim(N^*(x) \cap F) + 1 \end{aligned}$$

Si F est compact $N^*(x) = \overline{N^*(x)}$, tout $x \in F$, ce qui englobe la définition initiale.

4.3.2 Dimension d'un ensemble discret muni de relations de bases orthogonales

Soit (E, R_E) un support d'images où $E = \rho_1, \rho_2, \dots, \rho_n$

$\forall_i : \rho_i$ étant une relation de base et $\rho_i \perp \rho_j$ pour $i \neq j$.

Soient, pour tout $x \in E$:

$$ND(x) = \{y \mid (y, x) \in \rho_i \in R_E\}$$

$$ND^*(x) = ND(x) - \{x\}$$

$$N(x) = ND(x) \cup \{x\}$$

où Y est construit de la façon suivante :

Pour tout x_1 et $x_2 \in ND^*(x)$, $y \in Y$ existe si et seulement si il existe ρ_i et ρ_j $i \neq j$ tels que $(x_1, y) \in \rho_i$ et $(x_2, y) \in \rho_j$.

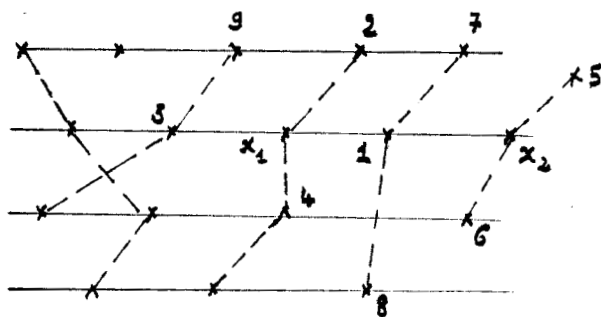
Exemple :

$$ND(x_1) = \{x_1, 1, 2, 3, 4\}$$

$$ND(x_2) = \{x_2, 5, 1, 6\}$$

$N(x_1) = \{x_1, 1, 2, 3, 4\} \cup \{7, 9\}$ 8 n'en fait pas partie puisque n'étant pas en relation en trait plein avec un voisin direct de x_1 par traits ponctués.

$$ND(x_2) = \{x_2, 5, 1, 6\} \cup \{\phi\}$$



Construisons maintenant une application $f : E \longrightarrow G_n$ ayant les propriétés suivantes :

$$f(x) \in G_n$$

$\forall y \in ND^*(x)$, $f(y)$ existe et $\in ND(f(x))$, ce qui est possible puisque $\text{card}(ND^*(x)) \leq 2n$ et $\text{card } ND(f(x)) = 2(n)$

Si y_1 et $y_2 \in \rho_i(x)$, alors $\exists g_i$ tel que $f(y_1) = f(x) g_i$
 et $f(y_2) = f(x) g_i^{-1}$

Ce qui est possible puisque $\text{card } R_E = n$

Soit maintenant $y \in N(x) - ND(x) = Y$

Par définition : $\exists x_1$ et $x_2 \in N(x)$ et ρ_i et $\rho_j \in R_E$ tels que $(x, y) \in \rho_i$ et $(x_2, y) \in \rho_j$.

Soient g_k et g_l tels que $k = l$, $f(x_1) = f(x) g_k$ et $f(x_2) = f(x) g_l$

Construisons $f(y)$ par $f(y) = f(x_1) g_l = f(x_2) g_k$

On peut alors énoncer :

Théorème 3 :

Pour tout $x \in E$, muni d'une structure algébrique composée de n relations de bases orthogonales deux à deux, il existe un voisinage lacunaire $F \subseteq N(x) \subset G_n$ groupe abélien libre fini, tel que $V(x)$ soit isomorphe à F .

Nommons \mathcal{F} la famille de voisinages $N(x) = F$ construite pour x parcourant E , on définira la dimension de E par :

$$\text{dim}(E) = \max_{F \in \mathcal{F}} (\text{dim}(F) \text{ prise dans } G_n)$$

On est assuré de l'invariance de cette dimension :

En effet si F_1 et F_2 sont isomorphes à $N(x)$ et si $\text{dim}(F_1) \neq \text{dim}(F_2)$, on a alors :

$$\dim(F_1) = \dim \overline{N}_1^*(z)_{F_1} \text{ et } \dim(F_2) = \dim \overline{N}_2^*(z)_{F_2}$$

Or, il existe une chaîne d'isomorphismes :

$$G_n \ni \overline{N}_1^*(z)_{F_1} \xrightarrow{\text{is.}} F_1 \xrightarrow{\text{is.}} V(x) \xrightarrow{\text{is.}} F_2 \xrightarrow{\text{is.}} \overline{N}_2^*(z)_{F_2} \in G_n$$

donc $\dim(\overline{N}_1^*(z)) \neq \dim(\overline{N}_2^*(z))$ est impossible d'après le corollaire 1.2 de [21] et $\dim(F_1) = \dim(F_2)$.

Applications immédiates

Affectant à un lot la dimension de son support :

Deux lots équivalents ont même dimension.

E est dit localement complet si et seulement si il existe un $x \in E$

au moins tel que :

$N(x)$ est isomorphe à $N(z)$ dans G_n .

Sa dimension est n .

Donc la dimension d'un lot (à relations de bases orthogonales deux à deux) est bornée par le nombre de ses relations.

4.4. Compatibilité et maillages

Relations compatibles

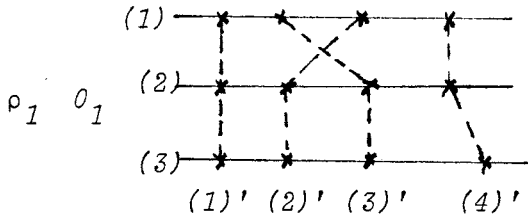
Un ordre total O sur une section initiale d'une relation de base induit un ordre total noté également O sur les ρ -chaînes de la relation et donc sur chaque section. Ceci nous permet de définir un cas particulier de l'orthogonalité de deux relations de base :

Compatibilité

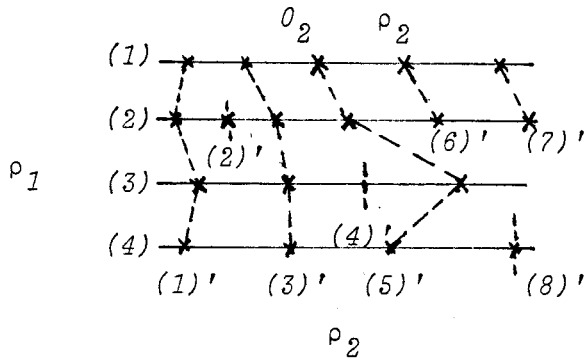
Deux relations de base définies sur un même ensemble, munies respectivement de deux ordres O_1 et O_2 seront dites compatibles si et seulement si :

- i) elles sont orthogonales,
- ii) les ordres O_1 et O_2 sont conservés sur toute section mutuelle définie par l'orthogonalité.

Exemple :



L'ordre (2)', (3)' n'est pas conservé sur la relation (1).
 ρ_1 et ρ_2 ne sont pas compatibles

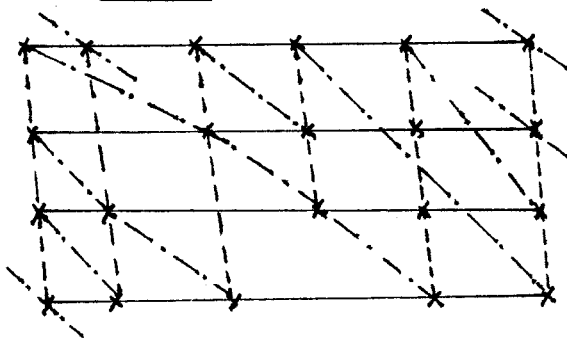


ρ_1 et ρ_2 sont compatibles, même si certains points forment à eux seuls une ρ -chaîne.

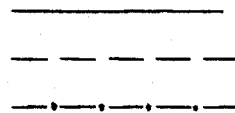
Conséquence : Toute ρ -chaîne d'un ensemble de deux relations compatibles est ordonnée.

Plus généralement, on dira que E est maillé si sa structure est formée de relations de base compatibles deux à deux.

Exemple :



Ensemble maillé à trois relations de base :



L'existence de lacunes dans le maillage, si elle n'est pas gênante pour le calcul de la dimension, n'est pas réaliste vis-à-vis des structures des images réelles. C'est pourquoi on imposera en général une condition restrictive aux maillages, quitte ultérieurement à considérer

un maillage lacunaire comme une partie d'un maillage régulier tel qu'on le définit maintenant.

Maillage régulier

Un maillage défini par $\rho_1, \rho_2, \dots, \rho_n$ sur un ensemble E est régulier si la condition suivante est réalisée :

$x \in E$, si $\text{card } \rho_i(x) = 1$, condition vide

si $\text{card } \rho_i(x) = 2$, $\rho_i(x) = \{x, a\}$, alors

x et a appartiennent à deux chaînes immédiatement successives pour toute relation ρ_j , $j \neq i$.

si $\text{card } \rho_i(x) = 3$, $\rho_i(x) = \{a, x, b\}$. a, x et b appartiennent trois chaînes immédiatement successives pour toute relation ρ_j , $j \neq i$.

Lemme

Sur un maillage régulier à deux relations de base, construit sur E , on peut construire dans le cas général une troisième relation de base maintenant le maillage régulier et ce, de deux façons seulement.

Soient ρ_1 et ρ_2 les deux relations de base, et x un point courant de E .

Dans le cas général : $\rho_1(x) = \{a, x, b\}$

$\rho_2(x) = \{c, x, d\}$

ensembles ordonnés par la compatibilité.

Par définition du maillage régulier, toute autre relation de base ρ_3 a trois chaînes successives passant par a, x et b d'une part, c, x et d d'autre part.

Soit $(\rho_3)_n$ celle qui passe par x , nommons $(\rho_3)_{n-1}$ celle qui passe par a et $(\rho_3)_{n+1}$ celle qui passe par b . Chacune de ces deux dernières

contient l'un des éléments c et d . On a donc deux solutions :

$$c \in (\rho_3)_{n-1} \quad \text{ou} \quad c \in (\rho_3)_{n+1}$$

$$d \in (\rho_3)_{n+1} \quad \text{ou} \quad d \in (\rho_3)_{n-1}$$

D'après la définition du maillage régulier, et, notamment, par la compatibilité des relations, on peut construire chacune de ces relations de proche en proche.

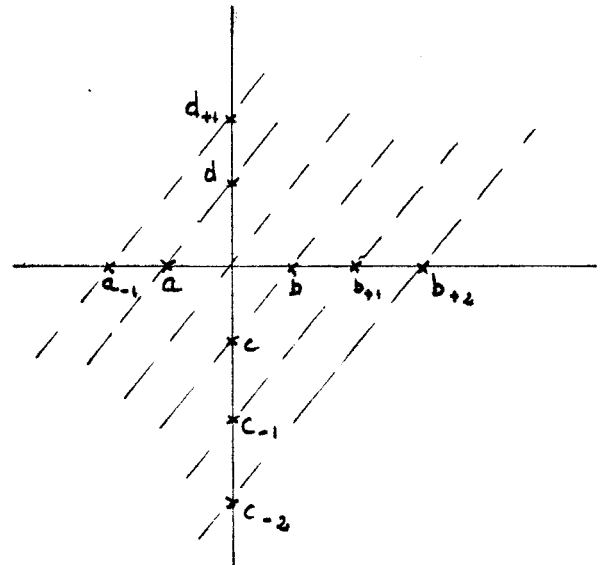
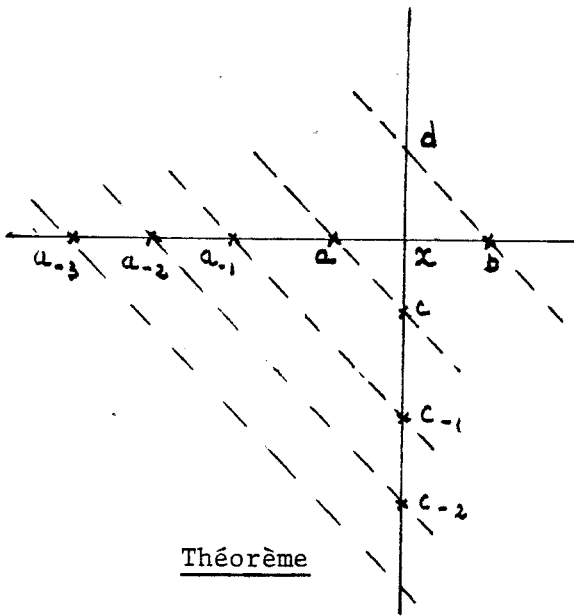
En effet : Soit ρ_3 tel que $c \in (\rho_3)_{n-1}$

$$d \in (\rho_3)_{n+1}$$

$$\begin{aligned} \rho_1(a) &= \{ a_{-1}, a, x \} \\ \rho_2(c) &= \{ c_{-1}, c, x \} \end{aligned} \quad (\rho_3)_{n-2} \text{ contiendra } a_{-1} \text{ et } c_{-1}$$

et ce, jusqu'au bout des ρ -chaînes.

Exemple :



Un maillage régulier contient, au plus, trois relations de base.

Supposons qu'il existe un maillage régulier à $n > 3$ relations de base, il est en partie régulier pour 4 quelconques de ces relations, soit ρ_1, ρ_2, ρ_3 et ρ_4 . Il est également régulier pour 3 et 2 relations respectivement ρ_1, ρ_2, ρ_3 et ρ_1, ρ_2 . Or, à partir de deux relations

ρ_1 et ρ_2 , il n'est possible d'en construire une troisième que de deux façons, ρ_3 est donc engendrée par une de ces solutions et ρ_4 formant maillage régulier avec ρ_1 et ρ_2 en particulier ne peut être que la deuxième solution. Or, ρ_3 et ρ_4 ne forment pas un maillage régulier, puisque, notamment, avec les conventions d'écriture précédentes :

a et c appartiennent à $(\rho_3)_{n-1}$ et c et a sont deux points successifs puisque $c \in \rho_3(a)$, et dans ρ_4 $a \in (\rho_4)_{n-1}$ et $c \in (\rho_4)_{n+1}$ ρ -chaînes non successives, q.e.d.

Groupes d'extension

Théorème

Un maillage régulier à deux relations de base est un espace discret de dimension 2 et est isomorphe à une partie compacte de G_2 .

Il suffit de montrer la deuxième partie de la proposition qui implique la première.

Pour cela, construisons un endomorphisme $f : E \rightarrow G$.

Soit $x \in E$, et $z = f(x)$.

$\rho_1(x) = \{a, x, b\}$ ordonné par O_2 , f peut être construit tel que $f(a) = f(x) g_1^{-1}$, $f(b) = f(x) \cdot g_1$.

De même : $\rho_2(x) = \{c, x, d\}$ ordonné par O_1 .

On construit f tel que $f(c) = f(x) g_2^{-1}$, $f(d) = f(x) \cdot g_2$.

Tout $y, \in \hat{\rho}_1(x)$ (resp $\hat{\rho}_2(x)$) est défini de façon unique par une suite : $x, x_1, x_2, \dots, x_p, y$ d'éléments de $\hat{\rho}_1(x)$ (resp $\hat{\rho}_2(x)$) et donc $f(y) = f(x) \cdot g_1^\alpha$ α entier $>> 0$ (resp $f(y) = f(x) g_2^\alpha$).

Soit maintenant y quelconque $\in E$, montrons que $f(y) = f(x) g_1^\alpha g_2^\beta$ où α et β sont uniques. S'ils ne l'étaient pas, on aurait :

$$f(y) = f(x) \cdot g_1^{\alpha_1} g_2^{\beta_1} \text{ avec } \alpha, \neq \alpha \text{ et } \beta_1 \neq \beta.$$

Par définition de g_1 et g_2 , il existerait deux chemins : de longueurs : $\alpha_1^{(\beta)}$ pas sur une ρ_1 -chaîne et $\beta_1^{(\beta)}$ pas sur ρ_2 -chaîne, ce qui est impossible puisque si $\alpha_1 \neq \alpha$, x_{k_1} à α_1 pas de y et x_k à α_p sur $\hat{\rho}_2(y)$ ne peuvent appartenir à la même ρ_1 -chaîne.

f est donc une injection conservant les voisinages f^{-1} préserve également les voisinages par construction et donc E est isomorphe à une partie compacte de G . et a pour dimension 2.

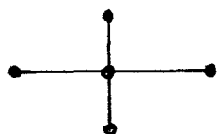
Théorème

Un maillage régulier à trois relations de base a pour dimension 3 et est isomorphe à une partie compacte de G_3 .

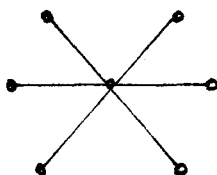
La démonstration se fait selon le même schéma que la précédente.

Corollaire : Une partie compacte de G_n , $n > 3$ ne forme pas un maillage régulier.

Exemples : Les systèmes de voisinages suivants forment un maillage régulier.

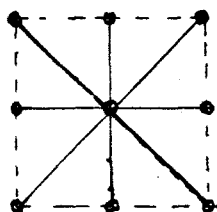


*voisinage de Von Neumann
dimension 2*



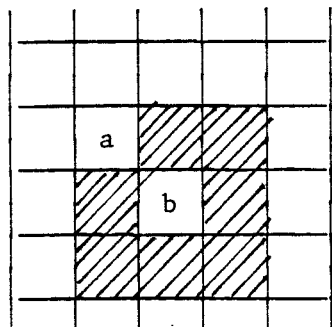
*voisinage hexagonal
dimension 2*

Le voisinage de Moore :

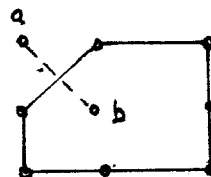


*ne forme pas un maillage régulier.
La figure qu'il engendre est de
dimension 3.*

La représentation plane qui en est faite engendre le paradoxe bien connu de Rosenfeld [2] : une figure fermée connexe peut avoir son intérieur connexe avec son extérieur :



soit :



Le paradoxe est facilement levé si l'on relève qu'il intervient entre deux relations de base ne formant pas maillage parfait, c'est à dire pour lesquelles la connexité ne joue pas : a et b voisins sur la diagonale qui les joint ne le sont pas au sens de deux chaînes successives de l'autre diagonale. On peut aussi dire que le sous-espace formé par ces deux relations ne forme pas un maillage parfait.

Si l'on ne considère que ce sous-espace, on constate qu'il est de dimension 3, et qu'il est formé lui-même de deux sous-espaces disjoints de dimension 3 également. Le paradoxe provient de ce que dans la figure considérée, on fait intervenir des voisinages de ces espaces disjoints.

4.5. Systèmes cellulaires

Un automate cellulaire est en général défini comme l'interconnexion de machines à nombre d'états finis, chacune étant placée sur un carré d'un damier infini : Butler [28]. Plus généralement, on pourra définir les structures mosaïques ou tabulaires par :

- Un support : collection d'éléments tous identiques, appelés points, représentatifs des positions des cellules.
- Une structure algébrique sur ce support, représentant le jeu d'interconnexions.

- Un ensemble d'informations attribuables aux points du support :
état interne et sorties des automates.

Cette définition fait d'un automate observé à un instant donné un lot d'informations.

Envisagé sous son aspect dynamique, il n'est plus un lot, mais un système de traitement de l'information. Pour cela, son emploi pour le traitement des images a souvent été envisagé [29] [30] [31], et il peut opérer comme un transformateur de lots, conservant la structure de support commune à lui et au lot d'image :

$$(I, E, R_E) \longrightarrow (I', E, R_E)$$

Ces transformations sont donc des opérations internes à la classe des lots isomorphes.

Il peut également opérer d'une façon quelconque : sommant, par exemple, en ligne et en colonne, les valeurs de quantification des points de l'image.

Remarque

Le fonctionnement de ces automates fait souvent intervenir des liaisons unidirectionnelles

On étend la dernière définition de la dimension d'un espace discret à ce cas, en observant que ces liaisons sont l'expression de relations de base orthogonales particulières : les relations de base orientées (cf. § 3).

CHAPITRE 5

REALISATION DES LOTS D'IMAGE

STRUCTURES D'INDEXATION

5.1. Rappels sur les structures algébriques

On rappelle les définitions de trois structures algébriques classiques [16]:

Un hypergroupe est une structure formée d'une seule opération binaire interne dans un même ensemble E .

Si T est l'opération, on notera indifféremment :

$$((x^*, y), z) \in T \quad \text{ou} \quad z \in x Ty \quad \text{ou} \quad z \in xy$$

L'opération n'est pas nécessairement toujours définie et son résultat unique.

Un monoïde est un hypergroupe, où le résultat de l'opération est unique et l'opération toujours définie et qui, de plus, possède les propriétés d'associativité et d'existence d'un élément neutre.

Un groupe est un monoïde où tout élément possède un inverse unique à gauche et à droite.

Un groupe abélien est un groupe où l'opération commute.

5.2. Structure générale par indexation.

Soit M_a l'ensemble des adresses d'une mémoire M .

Définissons $\tau \subseteq M_a \times M_a$ une relation binaire. τ est une opération

unaire non toujours définie : $\tau : M_a \longrightarrow M_a$ que nous appellerons transformation.

Considérons un ensemble $\mathcal{T} = \tau_1, \tau_2, \dots, \tau_n$ fini de transformations.

Nous pouvons définir une loi composition entre deux éléments quelconques de \mathcal{T} par :

$$\tau_j \circ \tau_i \subset M_a \times M_a,$$

où $\tau_j \circ \tau_i = \{(a, b) \mid \exists c, (a, c) \in \tau_i, (c, b) \in \tau_j\}$.

$\tau_j \circ \tau_i$ est évidemment une relation binaire interne, c'est à dire une transformation.

La loi \circ est donc interne et définit un hypergroupoïde sur l'ensemble \mathcal{T}^* des transformations .

Donc, d'après les propositions 6 et 7 de la première partie, chapitre 2 :

Théorème 1 :

La composition \circ des indexations a une structure d'hypergroupoïde.

Structure induite par des index multiples

Considérons un ensemble fini d'index, nombres entiers positifs

$$\mathcal{I} = p_1, p_2, \dots, p_n .$$

La transformation $\tau'_{p_i} : M_a \longrightarrow M_a$, telle que :

$$\forall p_i \in \mathcal{I}, \tau'_{p_i} = \{(a, a + p_i) \mid a \in M_a\}$$

est toujours définie si $\text{card } M_a$ est infini.

Si M_a est de cardinal fini, on conviendra de ne considérer que la restriction $\tau^*_{p_i}$ à M'_a , telle que $\forall a \in M'_a, \tau^*_{p_i} = \tau'_{p_i}$ est défini.

Comme dit page 28, ces égalités sont des congruences modulo $|M_a|$.

On dira, par la suite, comme indiqué au § 1.4 de la première partie que

p_i est localement définie.

Composition

Soient deux transformations τ_{p_i} et τ_{p_j} définies à partir de p_i et $p_j \in \mathcal{P}$ comme indiqué plus haut.

On définit $\tau_{p_i} * \tau_{p_j} = \tau_{p_i} \tau_{p_j}$ par :

$$\forall a (\tau_{p_i} * \tau_{p_j})(a) = \tau_{p_i}(\tau_{p_j}(a)) = (a + p_j) + p_i = a + p_j + p_i$$

appartenant à une restriction convenable de M_a .

Cette loi est évidemment localement associative :

$$\tau_{p_i}(\tau_{p_j} \tau_{p_k})(a) = \tau_{p_i}(\tau_{p_j}(\tau_{p_k}(a))) = a + p_k + p_j + p_i$$

Elle possède un élément neutre noté 1_τ : tel que :

$$1_\tau(a) = a + 0 \quad \forall a \in M_a$$

Elle est de plus commutative :

$$\forall a \tau_{p_i} \tau_{p_j}(a) = \tau_{p_j} \tau_{p_i}(a) = a + p_i + p_j$$

d'où :

Théorème 2 :

La loi $*$ dote un ensemble de transformations d'index d'une structure locale de monoïde commutatif.

Ambigüité et condition suffisante de non ambigüité

Les nombres $p_i \in \mathcal{G}$ étant pour l'instant quelconques, on peut trouver, pour deux suites :

$$\alpha_{i_1} \quad i_1 = 1, 2, \dots, n$$

$$\alpha_{i_2} \quad i_2 = 1, 2, \dots, n$$

distinctes :

$$b = \prod_{i_1} \tau_{p_i}^{\alpha_{i_1}}(a) = \prod_{i_2} \tau_{p_i}^{\alpha_{i_2}}(a). \text{ On dira alors qu'il y a ambigüité}$$

dans la décomposition de $b - a$.

Une condition suffisante de non ambiguïté que nous utiliserons par la suite est obtenue en :

(Cette C.S. est également valable pour le § 2.4. suivant)

i) Ordonnant \mathcal{Z} par valeurs croissantes, soit :

$$\mathcal{Z} = p_1, p_2, \dots, p_n$$

(observant que, comme on l'a déjà fait, la commutativité nous permet d'écrire tout mot du monoïde sous la forme :

$$\prod_i \tau_{p_i}^{\alpha_i}) ,$$

ii) imposant que $\forall 1 \leq i < n . \alpha_i$ soit borné par β tel que

$$\tau_{p_i}^{\beta}(a) < \tau_{p_{i+1}}(a)$$

La condition est bien suffisante comme on peut le vérifier par l'algorithme de décomposition d'une différence quelconque $b-a$, qui consiste à essayer successivement $(b-a) - \tau_{p_n}$, $(b-a) - 2 \tau_{p_n}$, ... et ensuite de même pour les τ_{p_i} dans l'ordre décroissant.

On trouve une réalisation analogue dans l'adressage relatif câblé : la pagination, usuellement à deux niveaux.

5.3. Index signés et structures locales

On appelle index signé un index affecté du signe + ou du signe - tel que, si p est valeur :

$$\tau_p(a) = a + p \quad \text{si le signe est +}$$

$$\tau_p(a) = a - p \quad \text{si le signe est -}$$

Soit la suite d'index $\mathcal{Z} = p_1, -p_2, p_2, -p_2, \dots, p_n, -p_n$.

et les transformations associées : $\tau_{p_1}, \tau_{-p_2}, \tau_{p_2}, \tau_{-p_2}, \dots, \tau_{p_n}, \tau_{-p_n}$

D'après le théorème 2, la loi $*$ construit un monoïde local sur \mathcal{L} mais ici, de plus :

$\tau_{p_i} * E_{-p_i} = \tau_{-p_i} * \tau_{p_i} = 1_\tau$, donc toute transformation a un inverse

et par commutativité :

$$\prod_i \tau_{p_i}^{\alpha_i} * \prod_i \tau_{-p_i}^{\alpha_i} = 1_\tau$$

Théorème 3 :

Un ensemble de paires opposées d'index signés est doté par la composition $*$ d'une structure locale de groupe abélien.

La mémoire ainsi structurée localement est un espace discret selon la définition du § [4.4] de dimension n .

5.4. Réalisation des lots d'images

5.4.1 Réalisation d'une relation binaire

Soit $R \in R_I$ une relation binaire élément de la structure algébrique du lot $L = (I, R_I)$, $R \subset I \times I$.

Soit M un support de mémoire universel d'après le théorème fondamental de la première partie § 2.3.

Considérons $\phi : I \longrightarrow M_a$ injection arbitraire, à tout couple (i_1, i_2) , ϕ fait correspondre un couple (a_1, a_2) avec $\phi(i_1) = a_1$ et $\phi(i_2) = a_2$.

On définit ainsi $\phi(R) = I \subset M_a \times M_a$, où τ établit une correspondance non toujours définie de M_a dans M_a .

L'implantation en mémoire consistera en :

- la définition de ϕ .
- la construction de la table de composition de l'hypergroupoïde défini par \circ sur τ .

5.4.2 Réalisation d'un ensemble de relations binaires

Avec les mêmes conventions d'écriture, remplaçant R par R_I et par \mathcal{E} ,

l'implantation en mémoire consistera en :

- la définition de ϕ .
- la construction des tables de composition \circ de l'hypergroupoïde \mathcal{E}^* .

Nous retrouverons ici les difficultés pratiques évoquées précédemment (première partie), pour la réalisation d'un lot quelconque. Nous allons voir que, dans le cas qui nous intéresse, celui des images numériques et spécialement quand le support de l'image est maillé, la réalisation locale bénéficie des propriétés de régularités de la structure algébrique du lot.

5.4.3 Réalisation des relations de base

Cas d'une ρ -chaîne isolée

Soit une structure ρ -chaîne définie par une relation de base sur I . Considérons sa représentation sous la forme d'une suite (nous supposons comme dit au § 1.4 deuxième partie, que les relations de base sont toutes à chaînes ouvertes).

Soit $\hat{\rho}(i)$ cette chaîne ouverte. Construisons ϕ tel que $\phi(i) = a$ quelconque, tel que les propriétés suivantes existent :

- Un index p étant donné, on dispose des deux transformations :

$$\tau_p \text{ et } \tau_{-p} = \tau_p^{-1} \text{ ainsi que de } 1_\tau.$$

- On associe successivement :

$$a_{-1} = \tau_p^{-1}(a) \quad \text{où } (i_{-1}, i_{+1}, i) = \hat{\rho}(i)$$

$$a_{+1} = \tau_p(a) \quad \text{ce qui définit } \phi \text{ sur } \rho(i) :$$

$$\phi(i) = a. \quad \phi(i_{-1}) = a_{-1} \quad \phi(i_{+1}) = a_{+1}$$

Cas d'un maillage parfait

Soit $a \in M_a$ la réalisation d'un élément $i \in I$, $\phi(i) = a$, dans le cas de deux relations ρ_1 et ρ_2 formant maillage parfait. La définition 14 deuxième partie et le § 3.3.2 précédent impliquent que l'on peut représenter ρ_1 et ρ_2 ainsi :

On choisit deux index p_1 et p_2 engendrant les transformations τ_{p_1} et τ_{p_2} et leurs inverses formant un groupe abélien local. On affecte au parcours sur une ρ_1 -chaîne la transformation τ_{p_1} (et $\tau_{p_1}^{-1}$) et au parcours sur une ρ_2 -chaîne la transformation τ_{p_2} (et $\tau_{p_2}^{-1}$).

Alors, à partir de a :

parcourir $\hat{\rho}_1(a)$ revient à se donner une suite de $\tau_{p_1}^{\alpha_1}$

parcourir $\hat{\rho}_2(a)$ revient à se donner une suite de $\tau_{p_2}^{\alpha_1}$

Trois relations de base formant maillage parfait seront de même

| décrites par | changements de chaîne |
|-------------------|--|
| ↓ | |
| $\hat{\rho}_1(a)$ | $\tau_{p_1}^{\alpha_1}$ |
| $\hat{\rho}_2(a)$ | $\tau_{p_2}^{\alpha_2}$ |
| $\hat{\rho}_3(a)$ | $(\tau_{p_1} + \tau_{p_2})^{\alpha_3}$ ou $(\tau_{p_1} - \tau_{p_2})^{\alpha_3}$ |

Quatre relations de base formant maillage, on obtient de même :

| | |
|-------------|--|
| $\rho_1(a)$ | $\tau_{p_1}^{\alpha_1}$ |
| $\rho_2(a)$ | $\tau_{p_2}^{\alpha_2}$ |
| $\rho_3(a)$ | $(\tau_{p_1} + \tau_{p_2})^{\alpha_3}$ |
| $\rho_4(a)$ | $(\tau_{p_1} - \tau_{p_2})^{\alpha_4}$ |

Toute ρ -chaîne étant finie, ϕ est construit sur $\hat{\rho}(i)$ et la ρ -chaîne est réalisée localement sur M_a .

Cas d'une relation de base unique

Une relation de base ρ sur I induit sur I une partition dont les éléments sont les ρ -chaînes $\hat{\rho}(a)$.

En vue de distinguer les ρ -chaînes entre elles, on emploiera deux transformations τ_1 et τ_2 de la façon suivante :

τ_1 opère à l'intérieur des ρ -chaînes comme indiqué au § précédent.

τ_2 opère entre les ρ -chaînes pour une de leurs extrémités au moins (cas des ρ -chaînes à un seul élément).

On munira la réalisation de la condition suffisante de non ambiguïté (§2.3 précédent). Pour cela, il est nécessaire de connaître $\max \text{card}(\hat{\rho}(i))$ et l'on prendra α borne liée à τ_1 égale à $\text{card}(\max \hat{\rho}(i)) + 1$ et $\tau_2(a) - a = \beta + 1$

Les éléments extrêmes des chaînes seront implicitement repérés par le fait qu'au delà l'atome de mémoire contient le code d'un signe blanc.

Les opérations locales sur les chaînes nécessitent également la disponibilité de τ_1^{-1} et de τ_2^{-1} . On peut donc énoncer :

La réalisation d'une chaîne nécessite au plus un groupe libre local de transformations à deux générateurs muni de la condition suffisante de non ambiguïté.

Cas de plusieurs relations de base

La réalisation de plusieurs relations de base présente les mêmes difficultés que celle d'un ensemble de relations binaires quelconques sauf pour l'une d'elles arbitrairement choisie. Les propriétés d'orthogonalité simple ou stricte n'introduisent que de très légères simplifications.

TROISIEME PARTIE

DEFINITION ET SIMULATION

D'UN SYSTEME DE TRAITEMENT NUMERIQUE DES IMAGES

On examine tout d'abord les méthodes utilisées pour définir les systèmes de traitement de l'information.

Les images ayant fait l'objet d'une modélisation de leur structure, non guidée par la spécification des traitements, on a été amené, pour définir l'architecture de machines informatiques adaptées à leur cas, à esquisser une méthode nouvelle présentée dans le § 6.3.

On l'applique aux images en 6.4.

Une simulation du biprocesseur obtenu est décrite ensuite.

CHAPITRE 6

ARCHITECTURE DU SYSTEME DE TRAITEMENT D'IMAGES

6.1. Les systèmes ou machines de traitement de l'information.

On ne peut pas donner a priori des définitions formelles de ces machines mais il est utile de classer les réalisations typiques et de les définir ainsi par exclusion.

Une machine de traitement de l'information est, dans le cas le plus général, un dispositif technique, c'est-à-dire construit par l'homme dans un but précis. Il reçoit et, par combinaison ou transformation, fournit des signaux informationnels, c'est-à-dire interprétables par l'homme à qui ils sont destinés. Ils peuvent être aussi transmis à un autre système quelconque.

Dans cette classe très vaste on distingue deux sous-classes, celle des systèmes analogiques et celle des systèmes discrets.

Dans les premiers les combinaisons ou transformations portent sur des grandeurs physiques continues, mécaniques ou électromagnétiques p.e.

Dans les seconds, elles portent sur des grandeurs quantifiées. Ces machines sont donc naturellement à nombre fini d'états.

Examinons plus avant cette deuxième classe. Ses différents représentants se distinguent par leur variété de fonctionnement :

- Celles qui ont un fonctionnement invariable : ne réalisant qu'une seule transformation : additionneur, multiplieur, transformateur de Fourier, etc...

- Celles qui sont en un certain sens universelles : ordinateurs à programme enregistré.

C'est, en général, à cette deuxième classe que l'on réserve le nom de machines informatiques. Toutefois cette distinction par variété de fonctionnement, ne peut pas servir à les classer toutes car entre les deux extrêmes de nombreuses réalisations ne peuvent être comparées entre elles. C'est ici qu'intervient la notion d'architecture et l'art de l'architecte. Il utilisera la spécificité de certains modules, l'universabilité de certains autres, pour assembler, méthodiquement pour une part, intuitivement pour une autre le système complet.

6.2. Méthodes de définition.

Nous distinguons trois voies.

6.2.1. Méthode de l'interprèteur.

Elle s'applique à la définition de l'unité de traitement, partie centrale de la machine, à laquelle seront ajoutés des organes divers, notamment pour la communication avec l'extérieur du système.

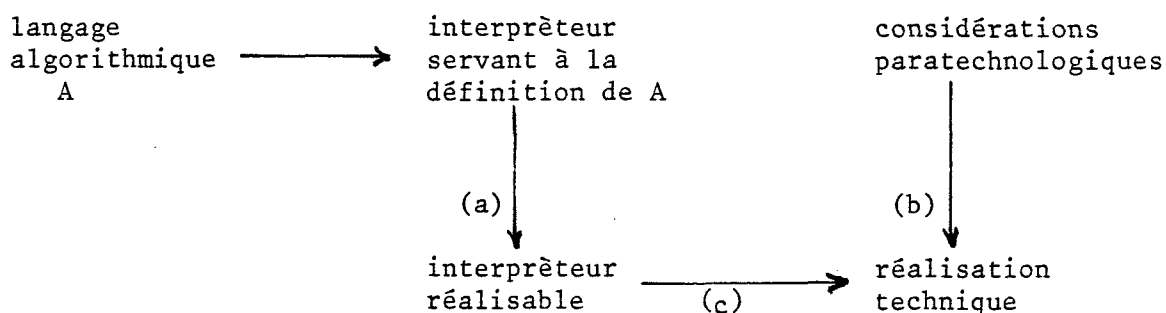
L'objectif posé au départ est de réaliser une unité qui exécutera une classe de traitements. Cette spécialisation n'est pas exclusive, elle doit s'entendre comme une meilleure adaptation de l'unité à la classe donnée.

La première étape consiste à formaliser cette classe de traitements à travers les algorithmes qui les réalisent et ensuite au moyen de la sémantique d'un langage qui permet de les exprimer.

Les traitements et les algorithmes associés peuvent porter aussi bien sur des calculs que sur des descriptions d'autres êtres, des machines par exemple.

La sémantique qui définit le langage doit être exprimée. On utilise pour cela un autre langage déjà connu et l'on décrit au moyen de ce dernier une machine de principe qui exécute les prescriptions du premier langage. Cette machine, appelée interpréteur a pour fonction la description exhaustive et non ambiguë du langage.

On peut alors envisager de passer à la réalisation technique de l'unité de traitement en partant de l'interpréteur. Ici intervient l'art de l'architecte comme le fait remarquer Anceau [13] qui résume les étapes de la définition par le schéma suivant :



Quand l'architecte intervient en (a), (b) et (c) deux options sont encore possibles. Dans la première, la machine réalisée exécutera les phrases du langage A sans intermédiaire. Dans la seconde, une machine de base sans rapport direct avec A recevra un ensemble de programmes interprétant A. Cette seconde option peut être démultipliée en : une machine de base, microprogrammée pour en faire une machine intermédiaire, elle-même programmée.

La réalisation se ramène alors à l'écriture de jeux de microprogrammes.

Exemples:

Le langage PL/1 décrit dans le langage VDL [32].

La machine Wang 2200 qui exécute directement les programmes écrits BASIC.

La machine Burroughs 6500 qui est déduite d'un interpréteur d'Algol.

6.2.2. La voie traditionnelle.

Elle ne mérite pas le nom de méthode. En effet elle se donne pour but une efficacité économique et consiste à améliorer l'existant. A partir d'une machine déjà réalisée, on essaye d'identifier d'une part les insuffisances de certaines fonctions, d'autre part l'acquis technique général. On en déduit alors empiriquement la définition d'une machine supérieure à la précédente dans ses performances.

Exemples :

Passage du CDC 6600 au Cyber 7.

Passage dans une même série d'un modèle à un modèle voisin, supérieur ou inférieur.

6.2.3. La voie analogique.

Alors que les deux précédentes sont bien connues, cette voie, représentée par de nombreuses réalisations, n'est pas présentée en tant que telle. Le nom que nous lui donnons est justifié par sa démarche.

Elle consiste à observer la structure d'un être déterminé : traitement ou données et à essayer d'en déduire la structure d'un système réalisable image du précédent. C'est en cela que l'on peut employer le mot analogique.

Elle a été plus particulièrement employée quand l'être d'origine était multidimensionnel : algorithmes parallèles, images bidimensionnelles, problèmes d'ordonnement p.e.

Les résultats qu'elle a permis d'obtenir ont toujours été limités car elle mêle deux projets de natures différentes : celui qui consiste à prendre une image de la structure donnée et celui d'un traitement très peu diversifié, voire unique. On aboutit ainsi très vite à une machine spécifique extrême dont la diffusion limitée interdit le plus souvent une industrialisation.

Exemples :

Machines de traitement d'images d'Unger | 29 |, d'Autechaud | 31 |, Processeur de Golay | 33 |, Machine Cybco | 34 | déduite de la machine réseau du Dr Sauvan | 30 |.

Parallélisation d'exécution dans les ordinateurs multiprocesseur Univac 1108, CII 10070 où des processeurs sont spécialisés dans des travaux différents : entrées et sorties, système d'exploitation, processeur de traitement.

Nous allons essayer dans le paragraphe suivant de systématiser la démarche analogique en précisant les bases qu'on lui imposera.

6.3. Essai de définition d'une voie systématique.

L'architecte tel qu'il est évoqué dans le §6.2.1. contribue non seulement à la définition de machines complètes mais aussi à celle d'unités de traitement isolées. Ce dernier travail est nécessaire mais non suffisant ; nécessaire car il convient que les unités de traitement soient toujours mieux adaptées à l'exécution des programmes ; non suffisant car un système informatique est constitué par l'assemblage de nombreux organes parmi lesquels l'unité de traitement.

L'évolution de l'intégration des composants électroniques fournit aujourd'hui des circuits complexes non dissociables dont on peut au plus utiliser une partie au lieu du tout. Parmi eux les unités de traitement ou processeurs, des organes d'entrée et de sortie, des mémoires. La relative inertie de ces productions dûe aux grandes séries nécessaires à l'amortissement des études amène à se poser de nouvelles questions telles que : comment choisir et assembler ces circuits dans un but déterminé. C'est à leur réponse que l'on réservera le terme d'oeuvre d'architecte.

Les mêmes questions se posent au niveau intermédiaire des circuits combinatoires et séquentiels classiques au delà des méthodes classiques de réduction et de synthèse.

Des travaux récents sont consacrés à l'étude de systèmes dotés d'une architecture a priori : hiérarchisés dans [13], étoilés comme les grands réseaux d'ordinateurs.

Il s'agit là essentiellement de mieux définir de telles architectures et d'en régler le fonctionnement.

Or nous avons vu que, dans la voie analogique, des architectures très diverses avaient été imaginées. Dès lors on peut se poser la question : quelle architecture et pourquoi celle-ci ?

Dans une optique plus restreinte, combien de processeurs et selon quels schémas d'interconnexion ?

Ces questions peuvent être posées pour une machine multiprocesseur aussi bien que dans une organisation (entreprise ou administration) sous les noms de décentralisation, informatique distribuée ou répartie.

Pour réduire la généralité des questions nous ferons trois conventions initiales :

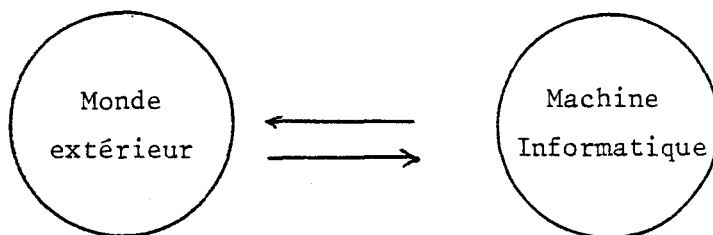
- 1) Les unités de traitement ou processeurs existent ainsi que d'autres organes identifiables, sinon ils peuvent être construits.
- 2) Bien que le fonctionnement d'une architecture soit fortement lié à sa définition, nous les séparerons.
La résolution conjointe des deux problèmes est encore non résolue et nécessitera probablement de nombreux travaux.
- 3) Les processeurs que l'on utilisera seront des organes physiques du type de Von Neumann.

6.3.1. Notion de système informationnel.

Depuis L. von Bertalanffy [35], les théories des systèmes se sont développées dans le sens du "management" (Ackoff [36]) et surtout, récemment vers l'auto organisation : Von Foerster [37], Atlan [38], et l'apprentissage : G. Pask [39] et les organismes vivants : Maturana et Varela [40].

Nous resterons dans une vue purement mécaniste suivant le premier auteur en insistant sur l'isomorphie des modèles. Après avoir établi les lignes directrices de la démarche nous en déduisons, à titre d'exemple l'architecture du système de traitement d'objets structurés que nous proposons.

La fonction des machines informatiques est de traiter des informations provenant et à destination de l'extérieur. Il est donc possible de modéliser très grossièrement leur position par le schéma :



La première tendance a été de choisir pour monde extérieur un système qui pouvait être le reste du monde. Dans les années 60 on a envisagé que quelques ordinateurs géants pourraient suffire aux besoins de calcul de la planète. Sans aller à cette extrémité un centre de calcul est toujours considéré comme "à tout faire". Or si cela est vrai pour des centres scientifiques et universitaires, l'observation montre que de très nombreux ordinateurs sont employés en liaison avec un système extérieur beaucoup mieux déterminé : une entreprise, une administration, un processus de production, voire un appareil seulement.

Apparaît ainsi une notion plus claire de couple "système extérieur"- "système de traitement".

La fonction du système informatique est alors de traiter et de gérer l'information de ce système extérieur.

On retrouve alors comme cas particulier, le point de vue classique de la machine "à tout faire". Dans des applications scientifiques par exemple, le système extérieur, constamment renouvelé, auquel est lié l'ordinateur est constitué par le modèle mathématique.

Le traitement consiste, au moyen d'un programme à réaliser une simulation de ce modèle. Les données sont les informations d'entrée.

Alors que dans ce point de vue, tout comme pour des applications de gestion ou d'autres, on peut considérer comme réaliste d'utiliser la méthode du § 6.2.1 et à défaut celle du 6.2.2, il n'en est plus de même si l'on prend pour point de vue celui de l'analyse du système extérieur.

Le point de départ n'est plus l'algorithme, spécification des traitements reportée au niveau de la conception des sous-systèmes et du fonctionnement de la machine, mais la structure informationnelle de ce système.

Pour la définir on ne peut plus guère s'aider d'une théorie puisqu'il s'agit d'un processus conduisant à un modèle-image ou modèle descriptif. Celui-ci doit contenir les éléments du système, significatifs vis-à-vis de l'information ainsi que les deux aspects fondamentaux du fonctionnement : synchronique et diachronique.

Dans ce qui suit nous allons donner l'esquisse d'une méthode sans qu'à ce stade de la recherche il soit possible d'en affirmer la généralité. Nous l'appliquerons ensuite au cas particulier du traitement des images.

La première modélisation est relative à celle des systèmes extérieurs (tels qu'ils ont été définis plus haut). On peut s'aider pour cela d'une méthode inspirée de l'analyse de gestion, celle des tableaux d'incidence qui contiennent les trois constituants mentionnés plus haut :

- éléments du système en lignes et en colonnes,
- perspective synchronique par le suivi d'une information dans le tableau,
- perspective diachronique par l'examen des origines et destinations des informations attachées à un élément.

Il est utile d'établir une typologie des éléments du système.

Celui-ci, considéré comme purement déterministe contiendra :

- des entrées par lesquelles transiteront les informations à destination du système, provenant du "reste du monde". La création d'une infor-

- mation en un point sera considérée comme une entrée auxiliaire,
- des points de transformation ou transformateurs,
 - des zones de stockage,
 - des puits où on pourra détruire des informations,
 - des supports de transmission,
 - des émetteurs vers l'extérieur.

La description précédente ne fournit pas directement l'architecture recherchée. Deux étapes au moins sont nécessaires.

6.3.2. Agrégation en processeurs :

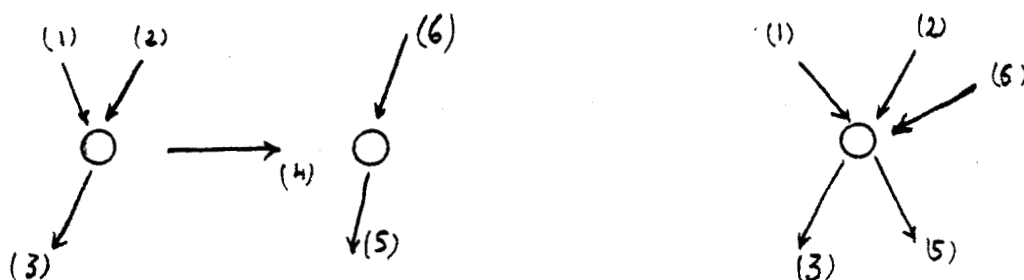
Nous nous intéressons ici aux points de transformation ou transformateurs. La fonction de chacun a été définie dans la première étape. Il s'agit ici de les spécifier en tant que processeurs par la tâche qui lui sera affectée sous la forme :

- des données d'entrée,
- de leurs conditions de validité,
- des données produites.

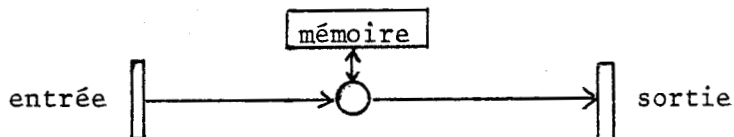
On a obtenu ainsi une architecture matérielle de principe. Selon la finesse de la description informationnelle de départ, elle est plus ou moins éclatée. Sa grande complexité peut la rendre incapable de s'adapter à des modifications des traitements; elle peut également la rendre économiquement non viable.

Pour la réduire on peut mettre en place une procédure de réduction. Elle consiste à affecter à un même organe : entrée, sortie, mémoire, processeurs, des fonctions différentes primitivement affectées à des organes distincts.

Si l'on représente l'architecture initiale par un graphe dont les noeuds sont les éléments du système et les flèches les liaisons informationnelles, un regroupement remplace deux noeuds de même nature par un noeud unique dont l'ensemble des arcs est l'union des ensembles des arcs des noeuds origine :



L'agrégation maximale fournit toujours le schéma de l'ordinateur unique à une entrée et une sortie :



Les critères d'agrégation sont multiples :

économiques pour réduire le coût du système,

logiques en regroupant des organes ayant mêmes ensembles d'arcs
ou des ensembles imbriqués,

organisationnels.

Leur choix est essentiellement affaire d'opportunité et de contraintes techno-économiques.

L'emploi de cette voie systématique ne résoud pas l'ensemble des problèmes posés.

Néanmoins elle nous paraît apporter des éléments de réponse aux questions telles que :

- centralisation ou distribution en matière de gestion ?
- quelle architecture multiprocesseur choisir pour une classe d'applications : réseau (ILLIAC IV), pipe-line (CDC STAR), arborescente, etc...

Ces questions se posent d'autant plus que de tels systèmes sont physiquement réalisables et que les théorèmes fondamentaux les classent comme tous équivalents. En effet, si une machine de principe (de Turing p.e.) est apte à exécuter tout algorithme d'une certaine classe, il en sera de même de toute construction équivalente (multiprocesseur p.e.), corollairement il n'est pas possible de choisir à l'intérieur de cette théorie entre ces schémas équivalents.

6.4. Architecture du système de traitement d'images numériques.

Nous sommes ici dans un cas particulier par rapport aux systèmes généraux des § 6.2 et 6.3. Les données sont organisées selon une structure précise et répétitive.

Les traitements prennent tous cette structure en compte. Quelle que soit la façon dont un point a été choisi, le traitement se ramène :

- au choix d'un voisin sur sa valeur et à un calcul sur celle-ci,
- à la combinaison des valeurs de points voisins.

Appliquant la méthode décrite dans les paragraphes précédents à la définition des images numériques des parties 1 et 2, on obtient :

une entrée : le capteur d'images,

une sortie : le restitueur,

un ensemble de $n \times p$ mots de mémoire ,

chacun contenant un point (n lignes et p colonnes).

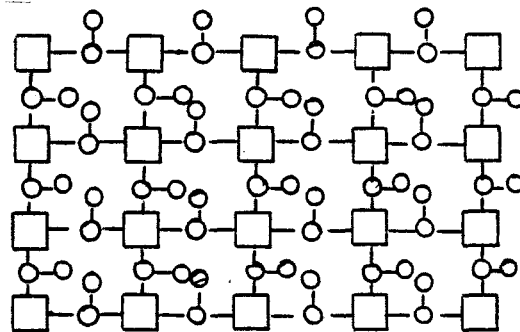
un ensemble de processeurs doubles : accès aux mots de mémoire et calcul.

Par exemple pour un voisinage 4 points : $P = n(p-1)+p(n-1)$

(figure 1) $= 2np-(n+p)$

A partir de là on peut faire jouer diverses procédures d'agrégation

Figures 1 à 4



$n = 4$

$p = 5$

□ mot de mémoire contenant la valeur d'un point.

○—○ processeurs de points accès et contenu

figure 1.

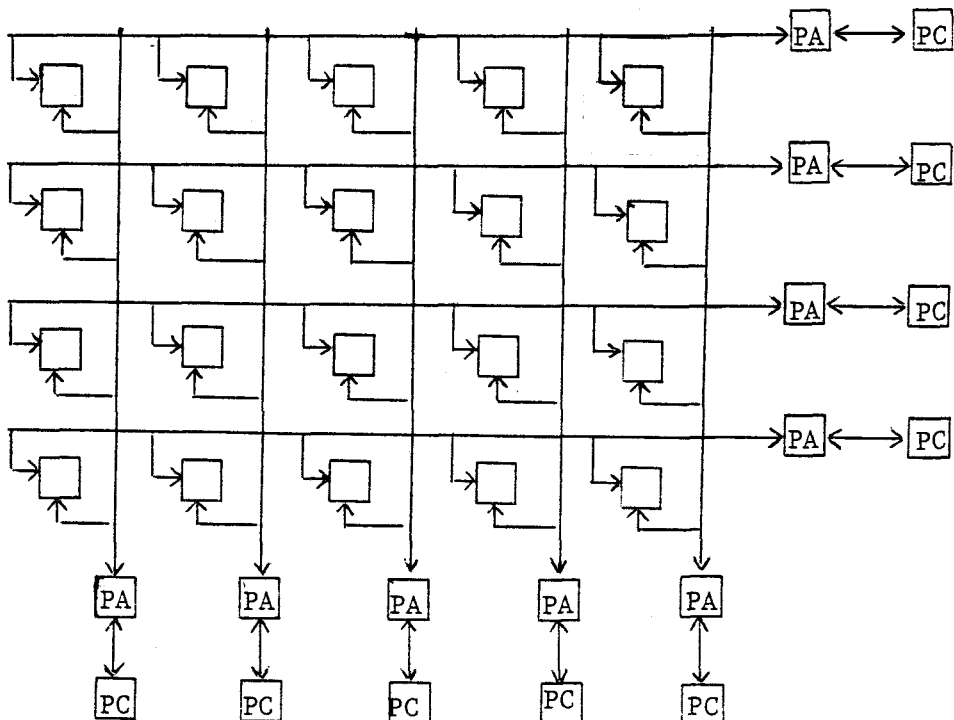
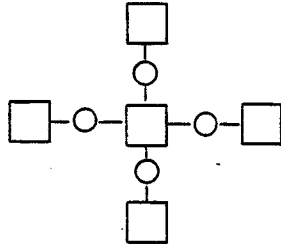


figure 2.

P.A.: processeur d'accès
P.C.: processeur de contenu



Processeur local de type fenêtre.

figure 3.

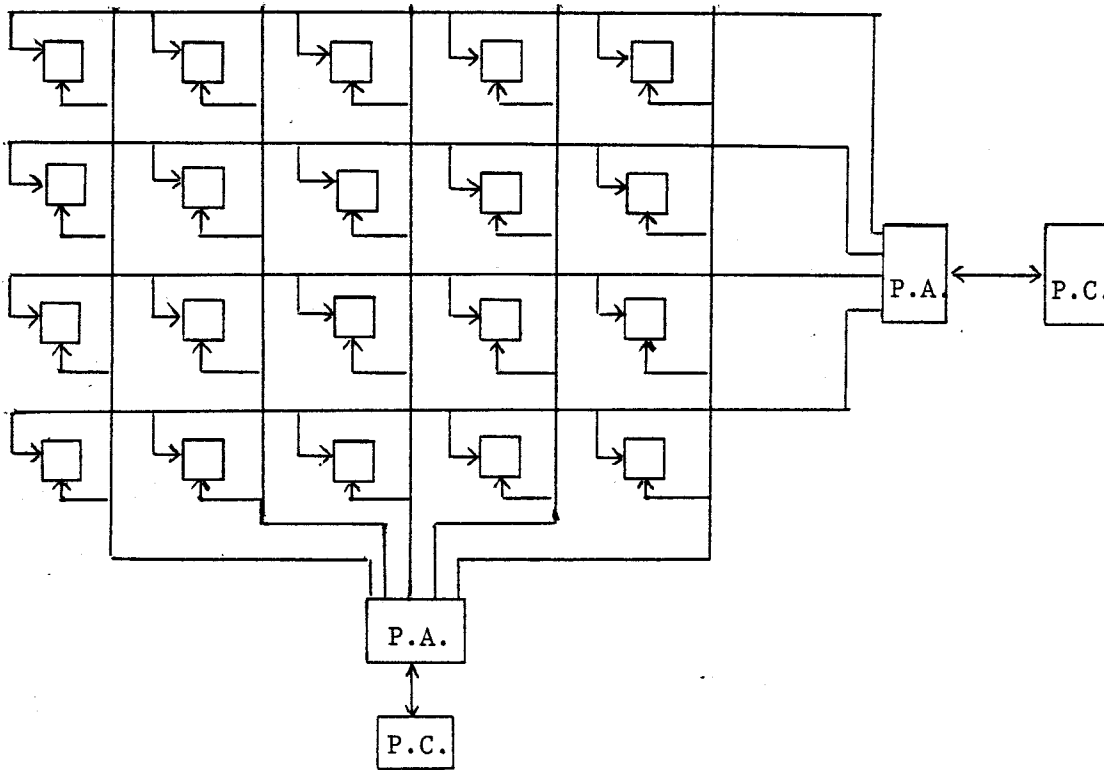


figure 4.



Agrégation par ligne et par colonne séparément (figure 2) toujours pour un voisinage de 4 points.

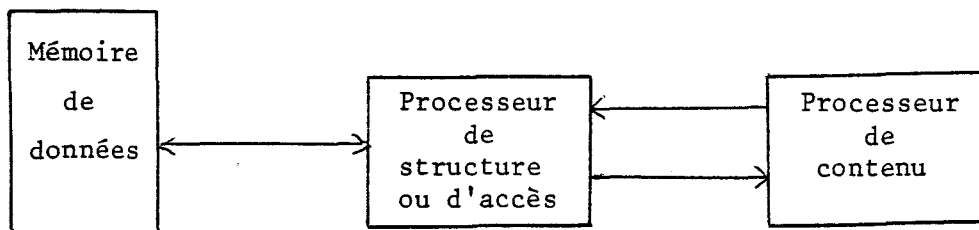
Agrégation de type fenêtre en partie comparable à celle d'Unger [29] ou d'Autechaud [31] figure 3.

Agrégation toutes lignes entre elles et toutes colonnes entre elles figure 4.

Nous voyons que les solutions sont nombreuses, et qu'un choix ne peut être fait que pour des raisons extérieures à l'informatique. Nous avons choisi de prendre une solution simple non encore explorée.

Le système se présente alors comme un biprocesseur muni d'une mémoire de données. Le processeur de structure assure le pointage sur un mot de mémoire et le changement de position selon une des possibilités de la structure du support. Il était donc possible dès lors, bien que ce ne soit pas imposé par la procédure d'adéquation de faire exécuter par ce processeur les opérations de lecture et d'écriture. Il manipule donc des nombres pour leur transmission.

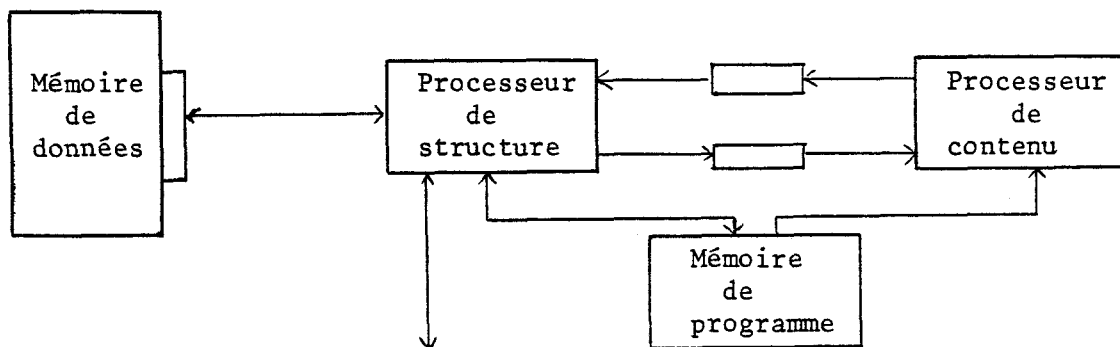
Un schéma partiel du système est alors le suivant :



Les organes suivants y sont ajoutés pour que le fonctionnement soit possible :

- un système d'accès à la mémoire
- une mémoire commune accessible aux deux processeurs
- deux files d'attente de communication entre processeurs.

D'où le schéma complet :



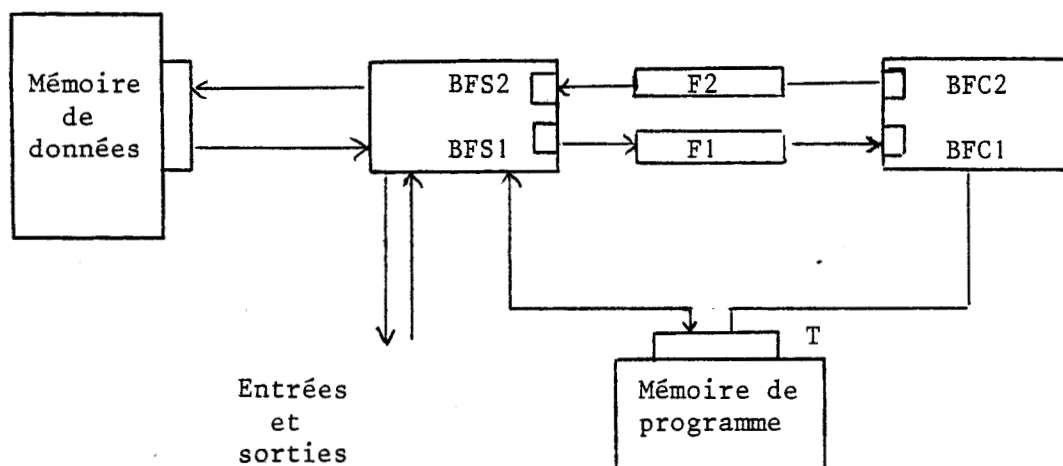
Entrées
Sorties

Dans la classification de Flynn [41] il s'agit d'un MIMD (multiple instruction stream, multiple data stream).

CHAPITRE 7

DESCRIPTION DETAILLEE ET SIMULATION DU SYSTEME

7.1. Description détaillée.



Les instructions logées dans la mémoire de programmes sont appelées sous le contrôle d'un automate T pointeur du programme. T contient un dispositif de décodage d'une marque spéciale indiquant à quel processeur l'instruction est destinée.

Sur ordre des instructions, le processeur de structure lit les contenus des mots en mémoire de données, et les transfère dans la file d'attente F_1 . Le processeur de contenu s'en saisit dans le même ordre, les transforme, et par une procédure utilisant la file d'attente F_2 les transmet au processeur de structure qui les inscrit en mémoire de données.

7.1.1. Processeur de structure.

Il dispose :

- de la mémoire de programme et de zones de travail

- de deux tampons contenant les données à transmettre et les résultats à recueillir du processeur de contenu : BFS 1 et BFS 2 respectivement chacun muni d'un indicateur : PFS 1 et PFS 2,
- de la mémoire de données,
- des index convenables selon les résultats du chapitre 5,
- du mécanisme d'accès à la mémoire de programme T.

7.1.2. Processeur de contenu.

Il dispose

- de la mémoire de programme et de zones de travail,
- de deux tampons contenant les données à recueillir et les résultats à transmettre au processeur de structure : BFC 1 et BFC 2, chacun muni d'un indicateur PFC 1 et PFC 2.

7.1.3. Les files d'attente.

Pour faciliter la plus grande simultanéité entre les tâches utiles exécutées dans les processeurs, on emploie deux files d'attente, du type premier entré, premier sorti. Elles sont constituées de registres à décalage à fonctionnement automatique. Elles transmettent des mots de longueur fixe.

Deux bascules d'état définissent l'état de chaque file d'attente vis-à-vis de l'extérieur.

SI = 1 accès autorisé

SI = 0 accès interdit

SO = 1 extraction autorisée

SO = 0 extraction interdite.

Elles seront nommées IF1 OF1 pour la file d'attente F1

IF2 OF2 pour la file d'attente F2

Elles sont attachées au premier et au dernier mot nommés BUF I et BUF O.

7.1.4. Eléments de programmation.

Les instructions sont de type classique. Les structures peuvent être microprogrammées. En régime séquentiel pur les instructions sont extraites une à une et dirigées vers le processeur d'exécution. Une anticipation d'un processeur à l'autre est donc possible. L'arrivée sur un branchement conditionnel ou incondtionnel provoque la suspension des possibilités d'anticipation jusqu'à obtention de son résultat qui est alors transféré au contrôleur T.

7.2. Simulation.

Le but visé est d'évaluer sur des cas réels les performances d'un tel système.

Nous supposons donc disposer de deux processeurs identiques. Le flux d'informations et les dénominations sont figurés sur le schéma de principe page précédente.

Les tâches de chacun des processeurs sont les suivantes, exécutées en priorités décroissantes pour éviter un engorgement des files d'attente :

PS : Tâche 1 rangement du résultat en mémoire

2 résultat chargé de la file d'attente dans un tampon

3 donnée transférée en file d'attente

4 donnée venant de la mémoire, chargée dans un tampon

5 tâche vide, d'attente

PS : Tâche 1 mise du résultat dans la file d'attente

2 calcul

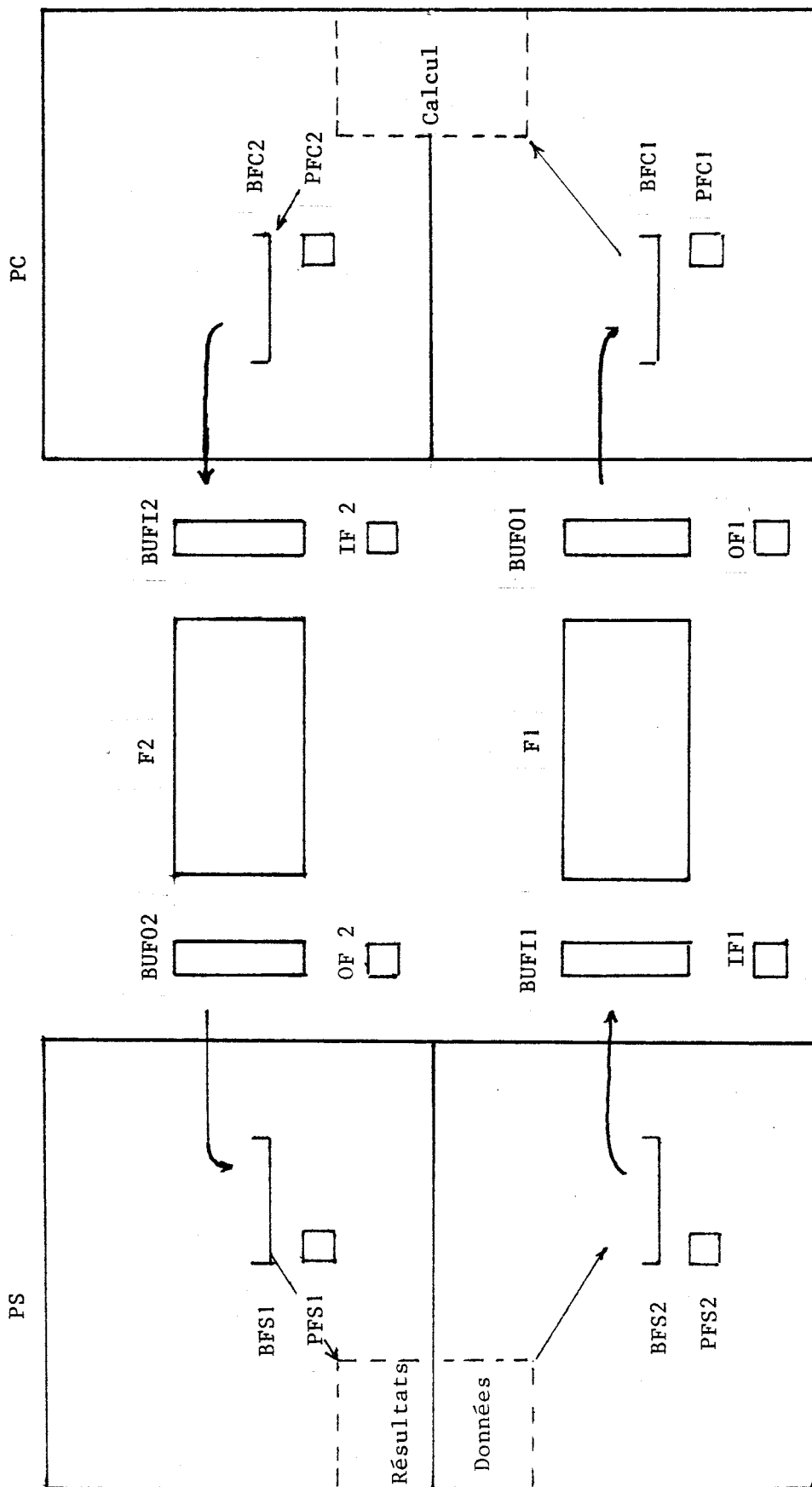
3 mise en tampon d'un mot de la file d'attente

4 tâche vide, d'attente.

F1 et F2 ont chacun une même tâche : après extraction d'un mot, faire avancer la file d'une unité.

SCHEMA GENERAL

Files d'attente



BNS LILLE

Le simulateur complet est organisé en deux niveaux. Le niveau supérieur est celui du moniteur qui sélectionne l'organe qui va exécuter un travail : l'un des processeurs ou une file d'attente. Ceci est réalisé par le test des horloges qui leur sont attachées.

Le niveau inférieur décomposé en autant de modules que d'organes. Ces modules choisissent la tâche qui va être exécutée, mettent à jour les horloges et informent les autres modules de la situation physique de la partie du système dont ils ont la charge (un mot a été rangé à tel ou tel endroit p.e.).

Les paragraphes suivants sont consacrés à la description de ce simulateur.

7.2.1. Moniteur.

Chacun des organes principaux, processeurs et files d'attente dispose d'une horloge qui comptabilise les temps de travail correspondant à chaque tâche élémentaire.

TS : temps du processeur de structure

$$TS = \sum_{i=1}^5 HS_i.$$

TC : temps du processeur de contenu

$$TC = \sum_{j=1}^4 HC_j.$$

TF1 et TF2, temps des files d'attente, somme des temps passés à opérer des décalages :

L'exécution d'un programme de traitement d'image est simulée de la façon suivante :

Au départ le programme est déjà rangé en mémoire ainsi que les données.

Les horloges sont mises à zéro.

Le moniteur est chargé de donner la main à chaque organe successivement selon une procédure double.

La main est donnée à l'organe dont le temps propre est le plus faible.

A égalité on choisit dans l'ordre de priorité décroissante :

Le processeur de structure,

Le processeur de contenu,

La file d'attente F2

La file d'attente F1

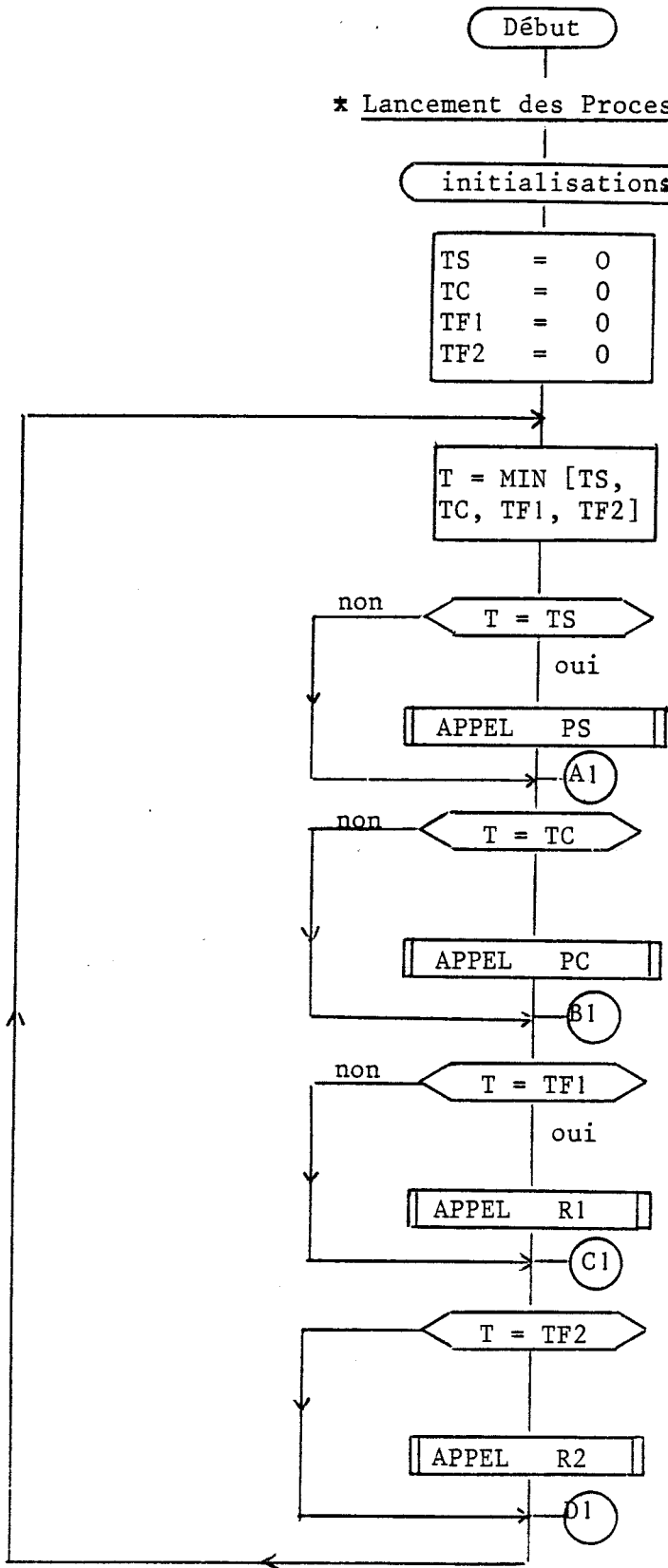
Quand un organe a la main, il choisit lui-même, au deuxième niveau, la tâche qu'il exécutera compte tenu de la situation des autres organes.

Une fois la tâche terminée, son horloge propre est mise à jour et la main est rendue au moniteur.

En fin d'exécution les horloges des processeurs marqueront la même heure, temps total :

$TT = TS = TC$. On pourra vérifier que $TF1 < TT$ et, $TF2 < TT$

L'organigramme du moniteur est présenté ci-dessous.



L'arrêt est provoqué par une intervention extérieure.

7.2.2. Processeur de structure.

Ayant reçu la main du moniteur, il tente d'exécuter les tâches possibles dans l'ordre de 1 à 5.

Tâche 1.

Elle ne peut être réalisée que si $PFS2 = 1$ indiquant qu'un mot est présent dans le tampon BFS2. Elle range ce mot en mémoire en appliquant la règle de calcul d'adresse.

Elle incrémente TS de DTS1.

Tâche 2.

Elle ne peut s'exécuter que si $PFS2 = 0$ et $OF2 = 1$.

Elle prend le mot disponible dans F2 et le range dans BFS2.

F2 est informé de l'opération par le signal SO2.

TS est incrémenté de DTS2.

Tâche 3.

Elle ne peut s'exécuter que si $PFS1 = 1$ et $IF1 = 0$.

Elle prend le mot disponible dans le tampon BFS1 et le met dans F1.

TS est incrémenté de DTS3.

Tâche 4.

Elle ne peut s'exécuter que si $PFS1 = 0$, indiquant que la tâche 3 a fourni un mot à F1.

Elle prend après calcul d'adresse un mot en mémoire de données et le met dans le tampon BFS1.

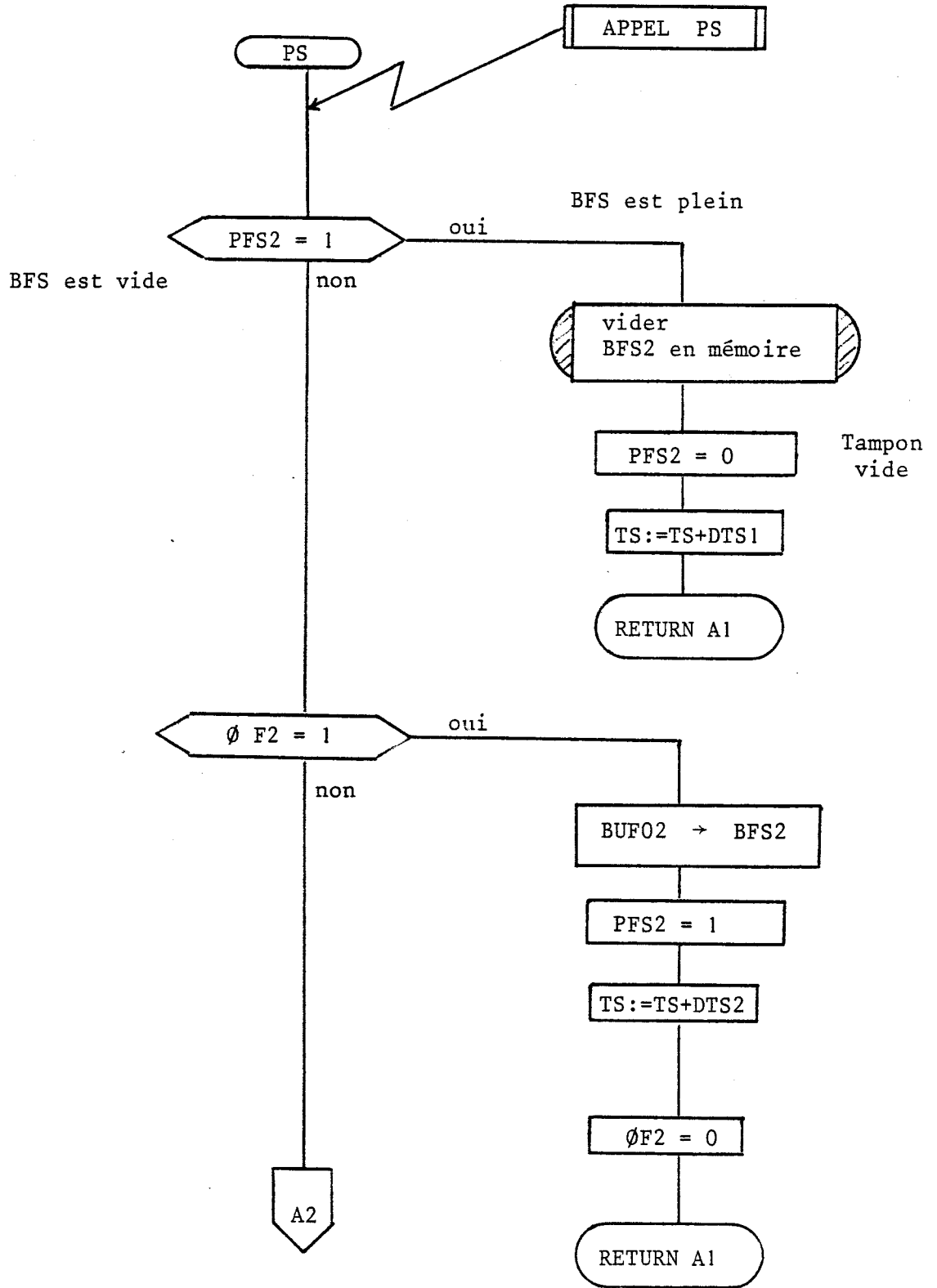
TS est incrémenté de DTS4.

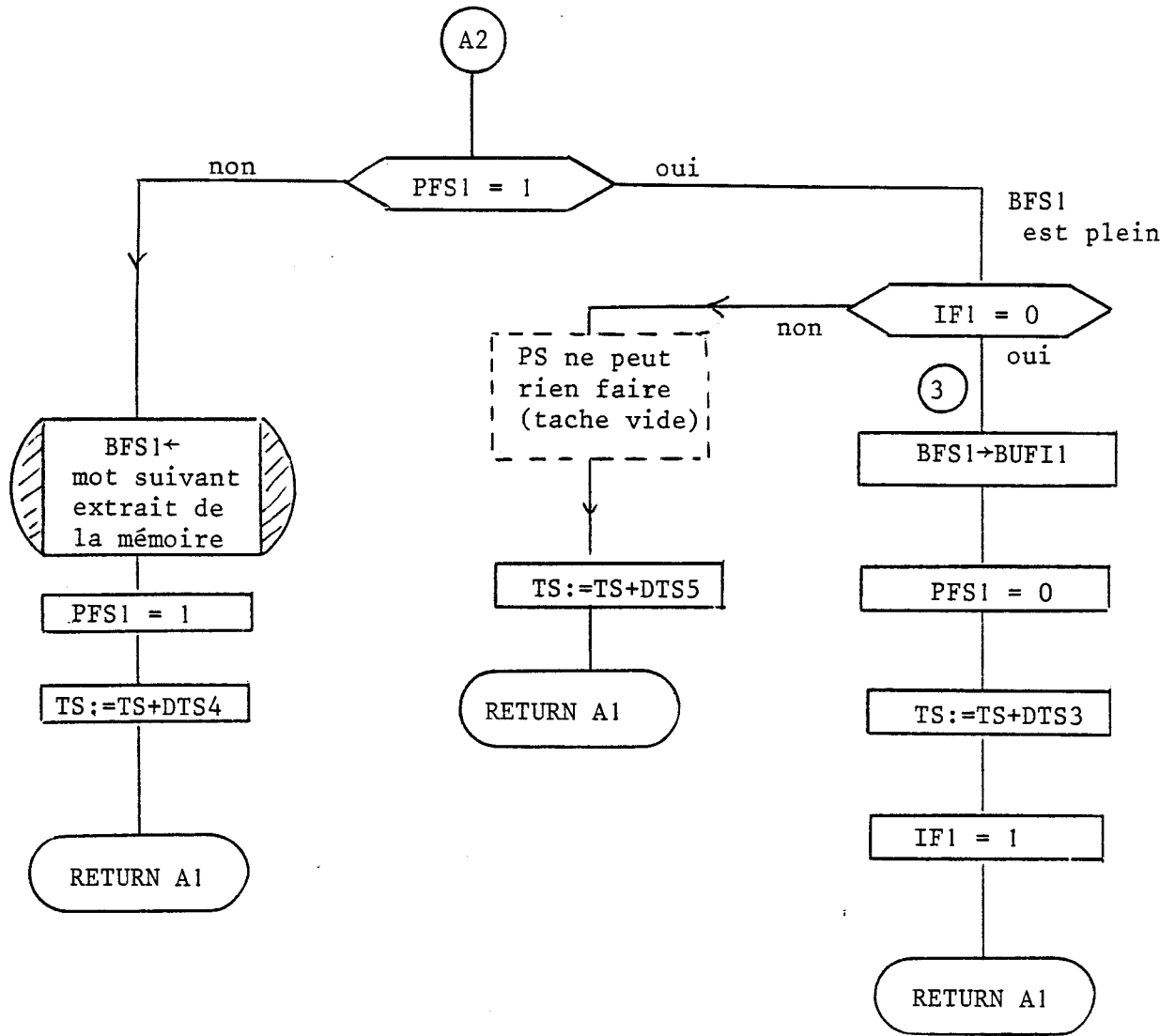
Tâche 5.

Tâche d'attente sans action.

TS est incrémenté de DTS5.

L'organigramme général est donné ci-dessous.





7.2.3. Processeur de contenu.

Ayant reçu la main du moniteur, il tente d'exécuter successivement les tâches 1 à 4.

Tâche 1.

Elle ne peut s'exécuter que si $PFC2 = 1$ et $IF2 = 0$ c'est-à-dire si le tampon BUF2 est vide et si la file d'attente F2 n'est pas pleine.

Elle prend le mot contenu dans BFC2 et le met dans F2.

TC est incrémenté de DTC1.

Tâche 2.

Elle ne peut s'exécuter que si $PFC1 = 1$ et $PFC2 = 0$ c'est-à-dire si la donnée est disponible dans le processeur et si le tampon de sortie est vide.

Elle exécute le calcul demandé.

TC est incrémenté de DTC2.

Tâche 3.

Elle ne peut s'exécuter que si $OF1 = 1$ et $PFC1 < X$.

Elle charge dans le processeur un mot de donnée.

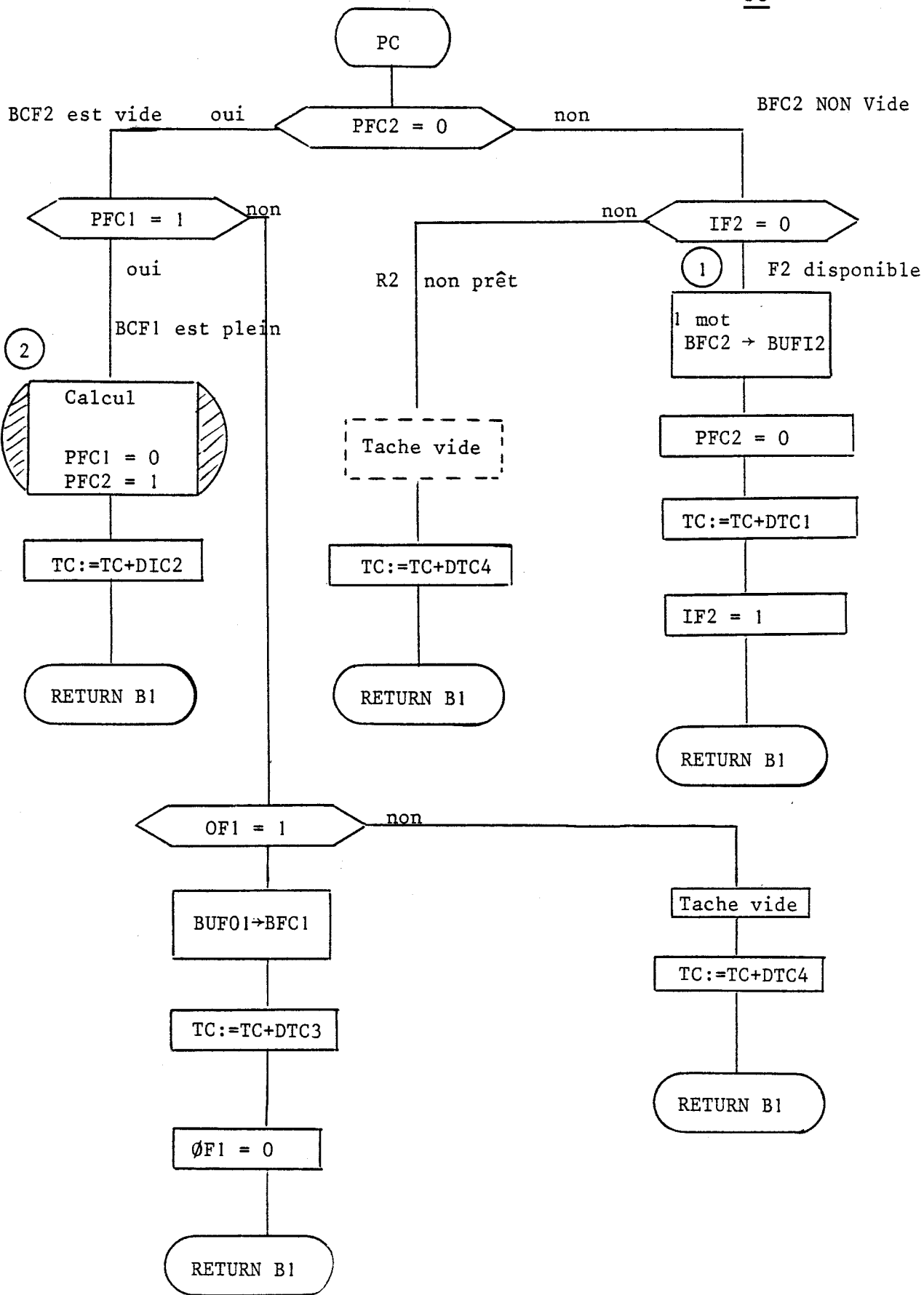
TC est incrémenté de DTC3.

Tâche 4.

Tache vide, sans condition ni action.

TC est incrémenté de DTC4.

L'organigramme général est donné ci-dessous.



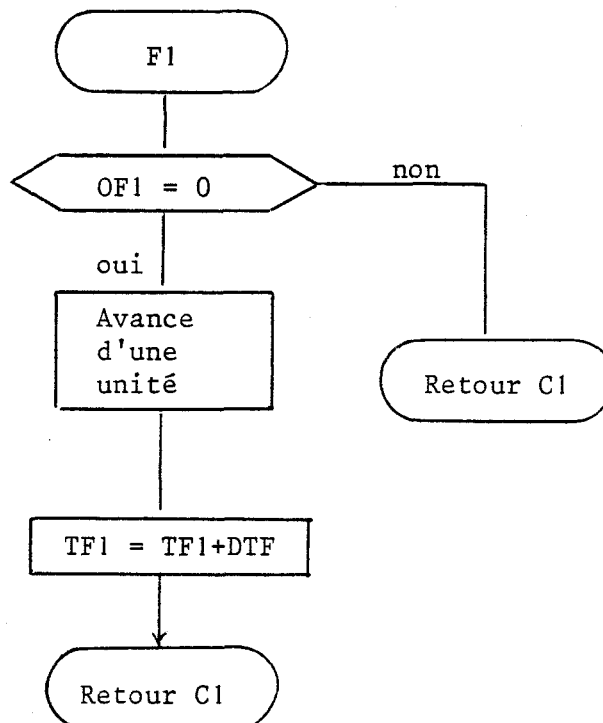
7.2.4. Files d'attente F1 et F2.

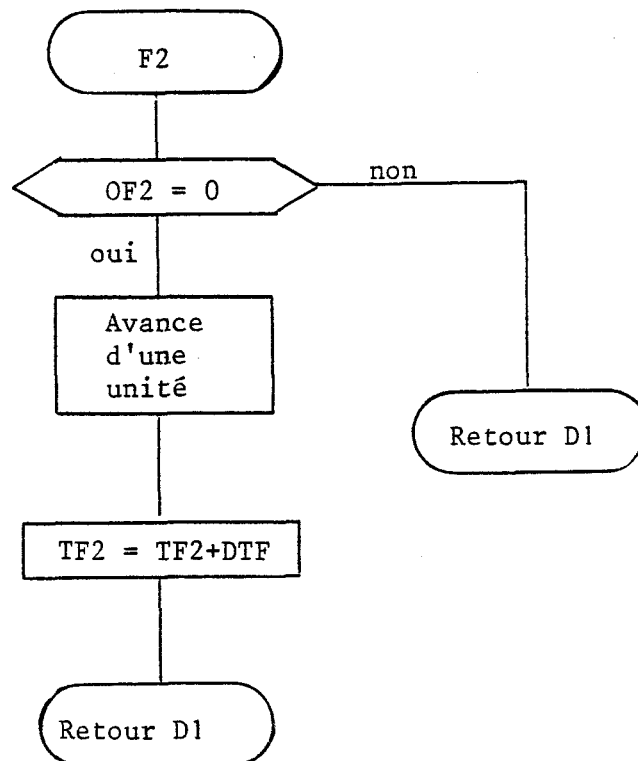
Leur fonctionnement comprend les possibilités suivantes :
extraction d'un mot, introduction d'un mot, décalage d'une unité.

Leurs horloges marquent le temps TF1 et TF2.

L'opération de décalage provoquée par l'extraction d'un mot, qui consiste à mettre en position d'un temps DTF, extraire le mot suivant incrément l'horloge.

Les deux organigrammes symétriques sont donnés ci-dessous.





7.2.5. Programmation et résultats.

On s'est placé dans la perspective de la réalisation du système présenté ci-dessus au moyen de micro processeurs IMP 16 C. Pour celà on a réalisé la simulation sur un ordinateur IMP 16. Les programmes écrits en langage d'assemblage sont fournis en annexe.

Les temps élémentaires DTS et DTC ont été relevés sur cet appareil, le temps DTF est celui d'un registre à décalage.....

On a appliqué la simulation aux cas suivants :

- 1) Lissage par voisinage à huit points.

La valeur d'un point est remplacée par la moyenne arithmétique des valeurs de ses huit voisins et de lui-même :

- 2) Lissage par voisinage à quatre points.

- 3) Nettoyage de figures à valeurs 0 ou 1 par remplacement des 0 isolés par des 1 et réciproquement.

On a relevé les résultats suivants, nommant

TT le temps total calculé par simulation sur le système
biprocasseur,

TM le temps relevé sur IMP 16 monoprocesseur et pour des images
traitées différentes IM1 à 4.

| <u>Cas 1</u> | IM1 | IM2 | IM3 | IM4 |
|-----------------|------|------|------|------|
| $\frac{TT}{TM}$ | 0,61 | 0,63 | 0,59 | 0,65 |

| <u>Cas 2</u> | IM1 | IM2 | IM3 | IM4 |
|-----------------|------|------|------|------|
| $\frac{TT}{TM}$ | 0,60 | 0,62 | 0,59 | 0,63 |

| <u>Cas 3</u> | IM1 | IM2 | IM3 | IM4 |
|-----------------|------|------|------|------|
| $\frac{TT}{TM}$ | 0,57 | 0,56 | 0,57 | 0,56 |

On peut donc conclure que le système proposé réduit très
sensiblement les temps de calcul dans une proportion comprise
entre un tiers et un demi.

La limite théorique : un demi ne pourrait être atteinte que
si dans chaque intervalle de temps les charges de travail des
deux processeurs étaient égales, ou mieux si aucun ne pouvait
être en attente des résultats de l'autre.

C O N C L U S I O N

Les thèmes traités dans les pages précédentes ressortissent tant de l'automatique que de l'informatique en passant par l'imagerie, qui accèdera peut-être au rang de science.

Nous avons voulu parcourir la chaîne qui va de la définition de l'objet (image numérique), jusqu'aux moyens de son traitement automatique.

Pour celà, la présentation a été faite en quatre points : définition générale, (lots d'information), représentation statique (en mémoire) et méthode d'accès, propriétés mathématiques des images numériques, architecture de traitement (aspect dynamique).

Pour chacun de ces points, si des réponses sont apportées, des questions restent ouvertes ou des voies de recherche peuvent être proposées.

- Sur les lots d'information. Un aspect n'est pas traité, celui de l'information qui leur est attachée, la structure elle-même est-elle informative ? Peut-on prolonger les travaux cités en [8], [9] et [10] ?

- Sur la représentation en mémoire. Ce mode d'accès a été choisi parmi les moyens classiques relatifs aux mémoires à accès direct. D'autres organisations sont envisageables : accès matriciels, défilements etc... qui pourraient mieux correspondre à une économie générale d'un système de traitement.

- Sur les propriétés mathématiques des structures d'image. La dimension d'un espace discret a été établie pour le caractériser par un invariant. L'isomorphisme entre la structure du support et l'adressage choisi rend applicable cette dimension aux mémoires et à leur mode d'accès. Des extensions à d'autres modes d'accès sont envisageables et notamment aux

systemes enchevêtrés de pointeurs.

- Sur l'architecture de traitement. Comme on l'a dit plus haut il s'agit d'une première tentative pour lever l'incertitude entre mono et multiprocesseurs. Ne pas avoir pris en compte le fonctionnement du système limite fortement ses possibilités d'emploi. Néanmoins cette limitation n'est pas intrinsèque et pourrait peut-être être levée par des moyens complémentaires réglant la synchronisation et les affectations.

B i b l i o g r a p h i e

1. R. THOM : La science malgré tout...in Oragnum, Encyclopaedia Universalis - 1973.
2. A. ROSENFELD : Picture Processing by Computer - Academic Press N.Y. - Londres - 1969.
3. J.L. KELLY : General Topology - Van Nostrand, Princeton - N.J. 1955.
4. J.P. BABARY : Contribution à l'étude des solutions de récurrences non linéaires - Thèse de l'Université de Toulouse - 1969.
5. A. ROSENFELD : Arcs and Curves in Digital Pictures. Journal of A.C.M., Vol. n° 1 - Janvier 1973.
6. STEFANELLI et ROSENFELD : Some parallel thinning algorithms. Journal of A.C.M., Vol. n° 18 - Avril 1971.
7. C. CASTORIADIS : Science moderne et interrogation philosophique, in Organum, Encyclopaedia Universalis - 1973.
8. D.M. Mac KAY : Quantal aspects of scientific information - Philosophical Magazine, 41 - 1950.
9. D.M. Mac KAY : Quantal aspects of scientific information, Institute of Radio Engineers, Report of Proceedings, Symposium on Information Theory - PGIT-1 - 1955.
10. B. PAYNE : The relations hip between judged complexity and amount of descriptive information for visual patterns generated from binary and ternary sequences Doctoral dissertation, University of Washington - 1962.
11. G.E. LOWITZ : La reconnaissance des structures par la théorie descriptive de l'information - Thèse Université de Paris - 1964.
12. J.LEE : Computer Semantics - Van Nostrand - 1972.

13. F. ANCEAU : Contribution à l'étude des systèmes hiérarchisés de ressources dans l'architecture des machines informatiques - Thèse, Université Scientifique et Médicale de Grenoble - 1974.
14. C. PAIR : Informatique non numérique - ENSI Nancy, 1976.
15. F.H. RAYMOND : Les principes des ordinateurs - P.U.F. - 1969.
16. J. LEVY-BRUHL : Introduction aux structures algébriques - Dunod - 1968.
17. D.E. KNUTH : The Art of programming, Vol. 2. Addison Wesley, Londres - 1969.
18. A. ROSENBERG : Data graphs and addressing schemes. Journal of Computer and System science, Vol. 5 - 1971.
19. Proceedings of the first international joint conference on pattern recognition - I.E.E.E. 1973.
Proceedings of the second international joint conference on pattern recognition - I.E.E.E. 1975.
sections picture processing.
20. ESTOURNET : Information d'images - Centre d'Edition de l'ENSTA - 1971.
21. J.P. MYLOPOULOS et T. PAVLIDIS : On the topological properties of quantized spaces, the notion of dimension - Journal of A.C.M. Vol. n° 18 - Avril 1971.
22. T. HUANG, W SCHREIBER et O. TRETIAK : Image processing - Proc. of I.E.E.E., Vol. n° 59 - Novembre 1971.
23. B. HUNT et J. BREEDLOVE : Scan and display considerations in processing images by digital computer I.E.E.E. Trans. on Computers - Août 1975.
24. J. KUNTZMANN : Théorie des Réseaux, graphes. Dunod, Paris - 1972.
25. P. OTA : Mosaic Grammars - Pattern Recognition, Vol. 7, n° 1/2 - 1975.
26. D. ERRERA : Sur les polyèdres réguliers de l'analysis situs. Académie Royale Belge - Collection 8 (2) - 1922.

27. A. ALTSHULER : Construction and enumeration of regular maps on the torus. Discrete Mathematics n° 4 - 1973.
28. J. BUTLER : A note on Cellular Automata Simulations. Information and Control 26 - 1974.
29. S.H. UNGER : A computer oriented toward spatial problems, Proceedings of I.R.E. Octobre 1958.
30. J. SAUVAN et J. BARTHELEMY : Procédé de reconnaissance des formes. Automatisme n° 9 - Septembre 1969.
31. J. AUTECHAUD : Etude d'un lecteur optique de caractères manuscrits. Thèse du Conservatoire National des Arts et Métiers - 1974.
32. P. WEGNER : The viésma definition language. A.C.M. Computing Surveys - Mars 1972.
33. GOLAY : I.E.E.E. : Trans on Computers. Vol. C 18 - Août 1969.
34. Brochures Commerciales.
35. L. Von BERTALANFFY : General System Theory. Groege Braziller, New-York - 1968.
36. R.L. ACKOFF : Toward a system of systems concepts. Management Science, Vol. 17 n°11 - Juillet 1971.
37. P.E.WESTON, H. Von FOERSTER : Artificial intelligence and machines that understand. Annual review of physical chemistry, Vol. 24 - Palo Alto - 1974.
38. H. ATLAN : L'organisation biologique et la théorie de l'information - Hermann - Paris 1972.
39. G. PASK : An approach to cybernetics. Hutchinson London, 3ème édition - 1972.
40. H. MATURANA, F. VARELA : Autopoietic systems. Rapport Facultad de Ciencias, Universidad de Chile, Santiago, Chile.
41. M.J. FLYNN : Some Computer Organizations and their effectiveness. I.E.E.E. Transactions on Computers - Septembre 1972.
42. J. RANCHIN : Informations structurées et supports. Revue du Cethedec, 2ème trimestre 1977.

43. J. RANCHIN : Définition d'architectures multiprocesseurs.
Séminaire de Reconnaissance des Formes. Belfort, Avril 1977
44. SHI - KUO CHANG : A model for distributed computer system
design. IEEE Trans. on Systems, Man, and Cybernetics ,
vol.SMC 5 n° 6, mai 1976.
45. R. BORRIN : Système de traitement d'images, Thèse C.N.A.M.1975.
46. J. RANCHIN : Un système d'acquisition, traitement et resti-
tution d'images. Colloque National du GRETSI, Nice, 1975.
-

Plans de réalisation

1) Choix techniques.

La réalisation projetée suit exactement la description détaillée du chapitre 7. Les processeurs choisis sont identiques pour des raisons de simplicité de mise en oeuvre et de programmation au niveau d'une maquette ; ce sont deux IMP 16 (National Semi Conductor). Cette identité permet de plus de vérifier les résultats obtenus par simulation sur les vitesses d'exécution.

Les files d'attente sont du type FIFO AM 3341, Chacune est constituée par registres de 4 bits chacun d'une FIFO. Leur diagramme logique est donné en figure 1.

Leurs signaux de contrôle obéissent aux chronogrammes de la figure 2.

2) Cartes de réalisation.

L'ensemble est organisé sur 2 cartes :

Carte 1 de l'IMP 1 vers l'IMP 2. fig 3,3b, 3t

Carte 2 de l'IMP 2 vers l'IMP 1. fig 4,4b, 4t

Chacune des cartes est munie d'un connecteur,

Connecteur 1 (fig 5)

Connecteur 2 (fig 6)

Un dispositif de visualisation (fig 7) permet de repérer l'état des extrémités des files d'attente.

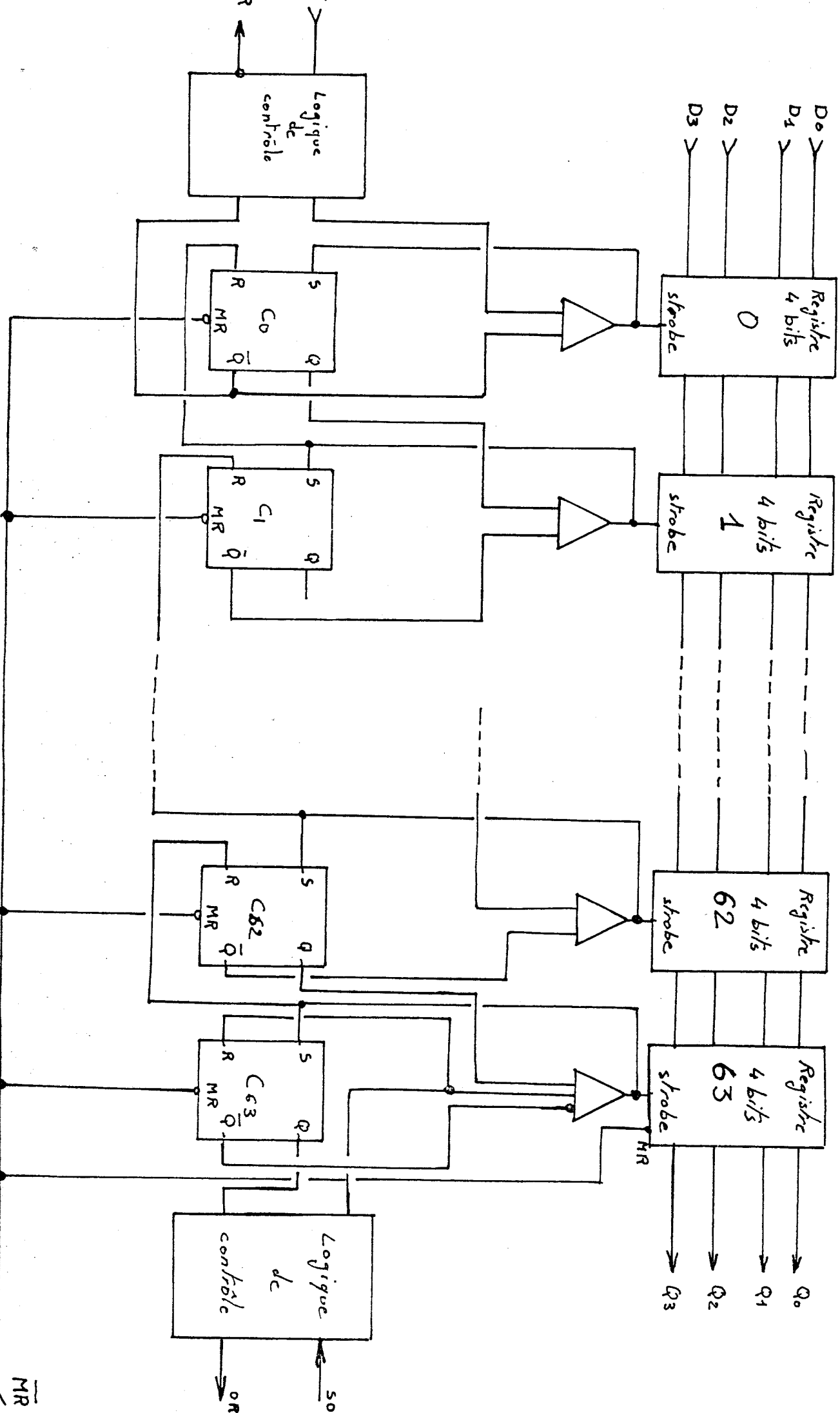
Enfin vers chacun des IMP se trouve une prise

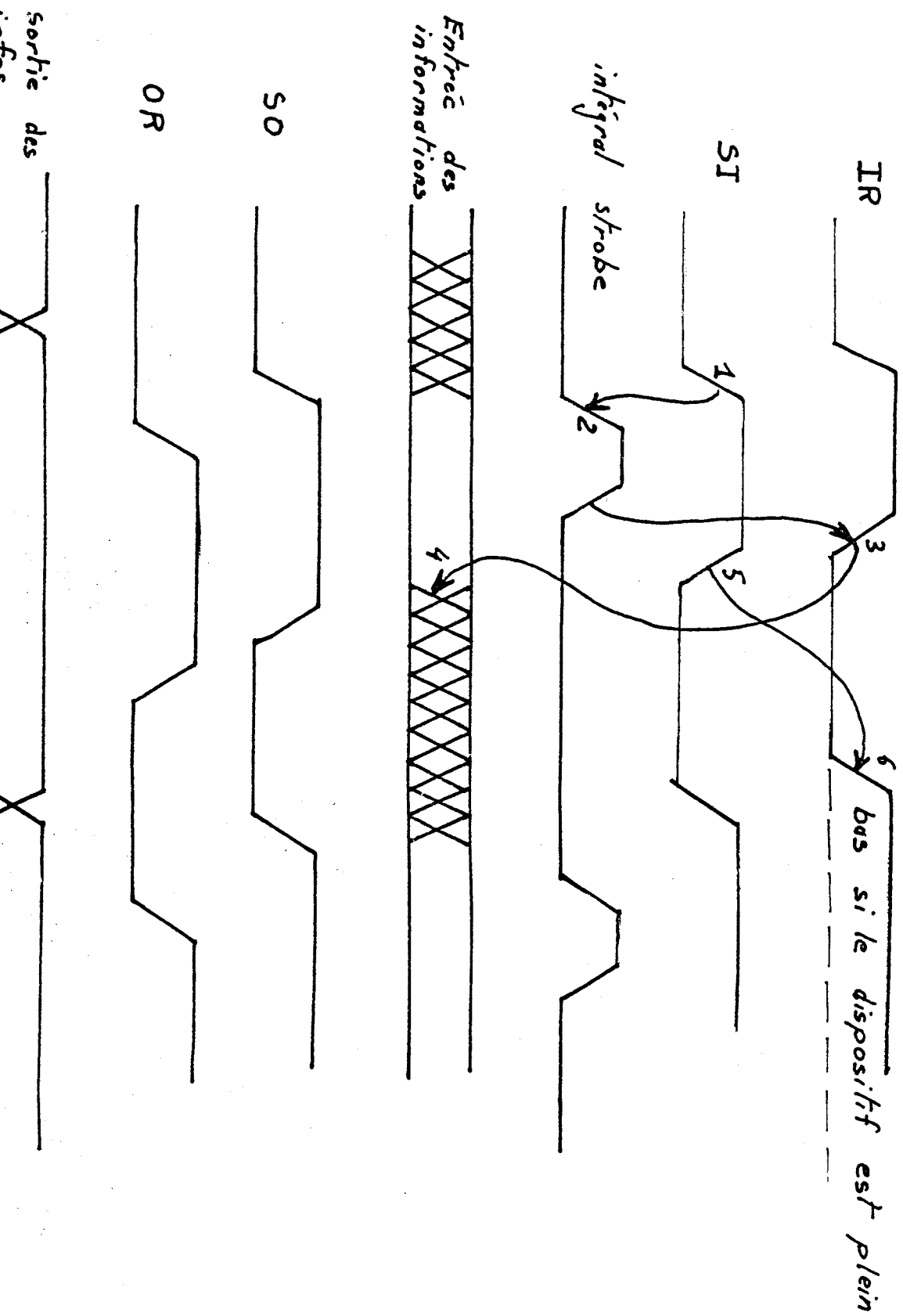
Prise P 1 vers IMP 1 fig 8

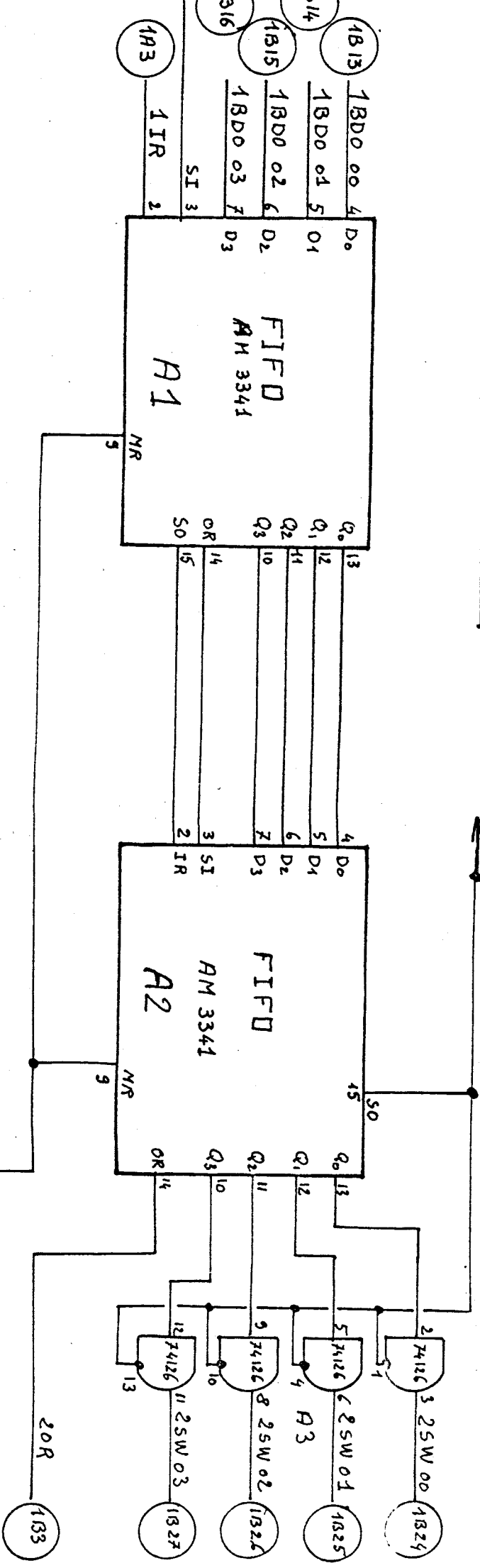
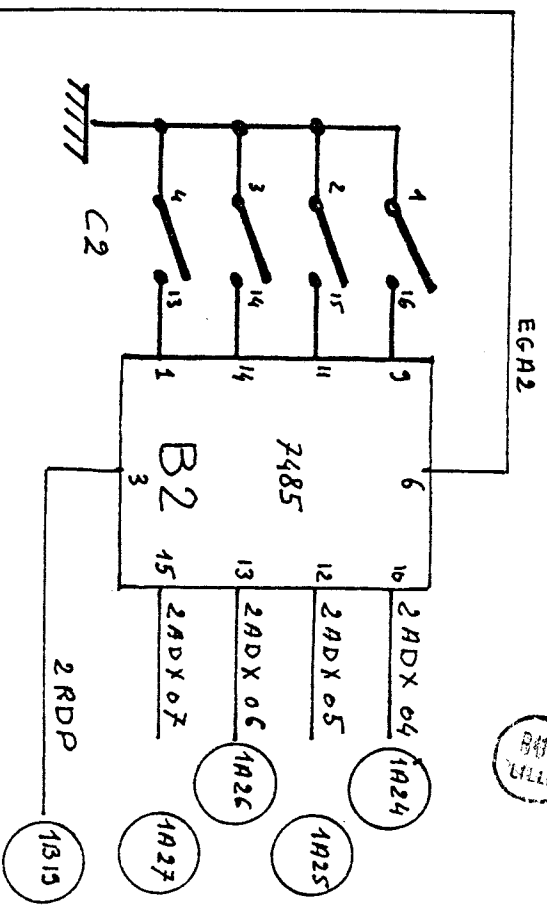
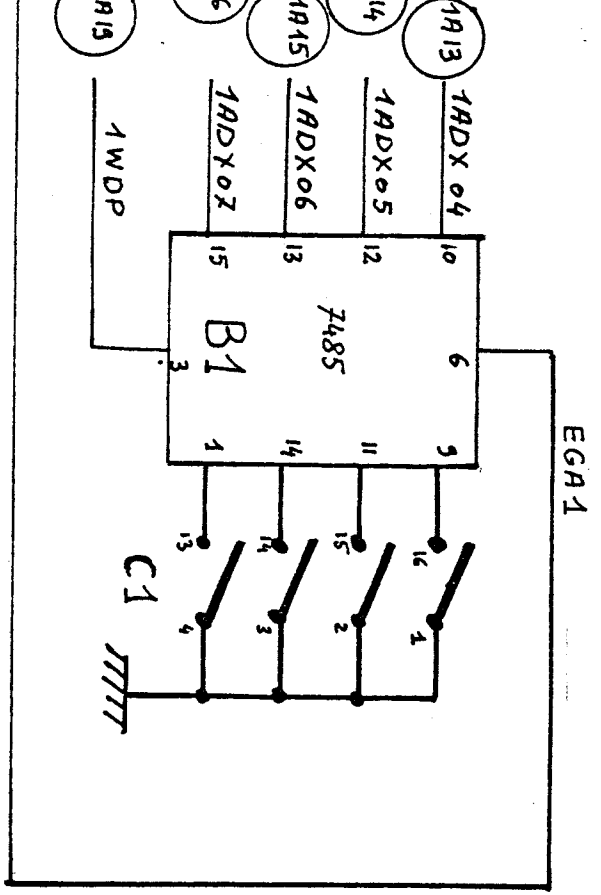
Prise P 2 vers IMP 2 fig 9

3) Schéma d'ensemble, fig 10

Il reprend les éléments d'entrée et de sortie précédemment réalisés : écran, cathodique et caméra (45) et (46).







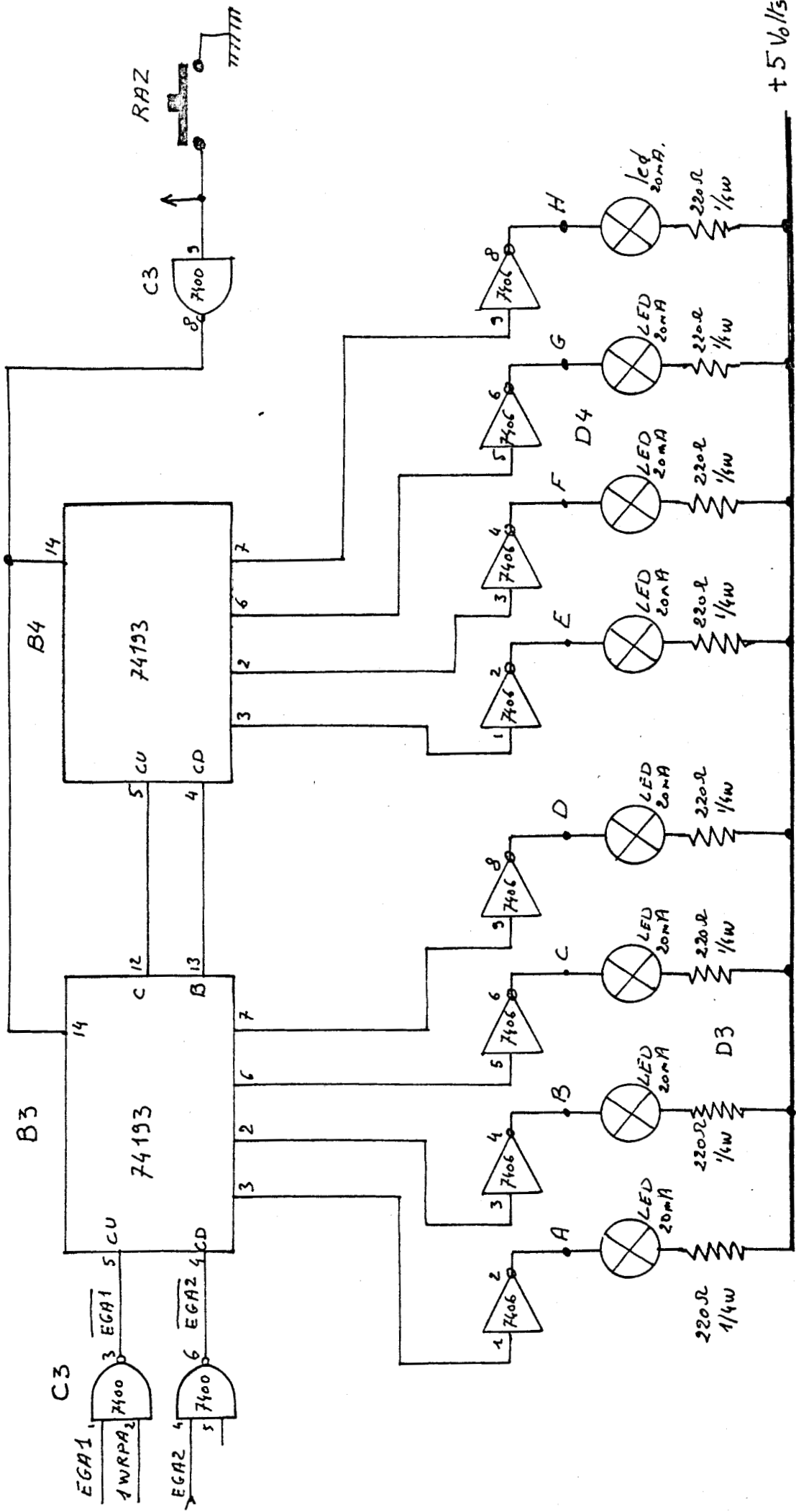


Figure 3 bis

Carte 1

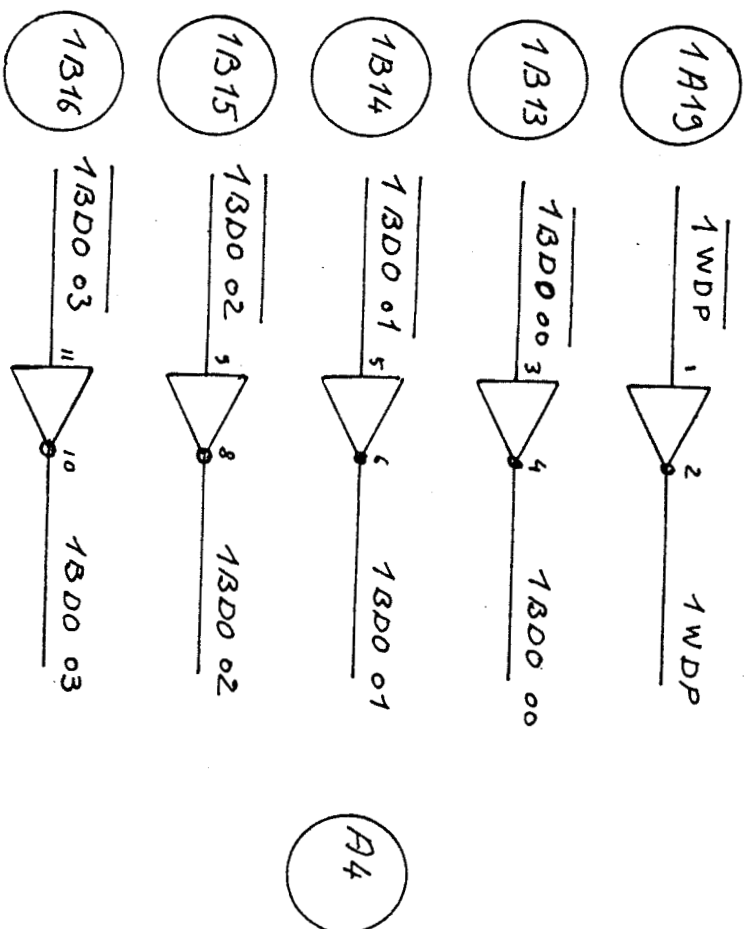
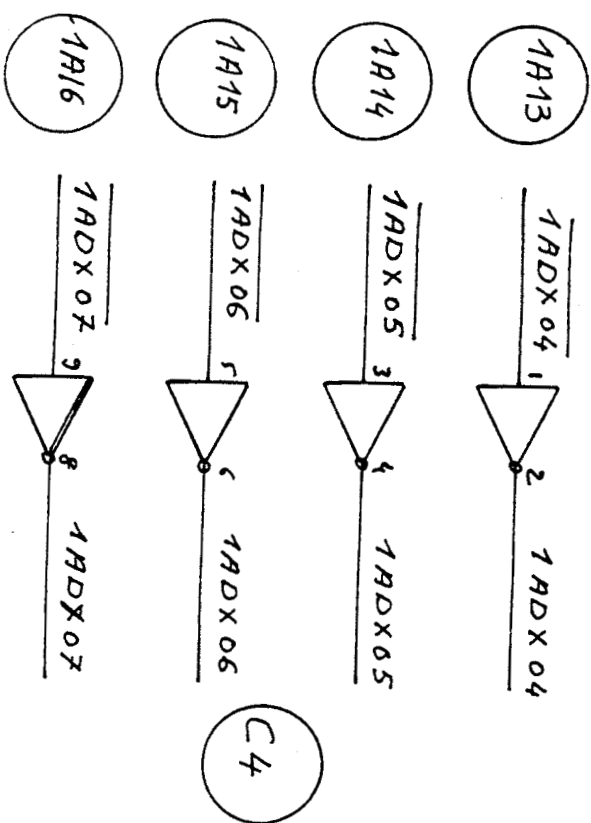
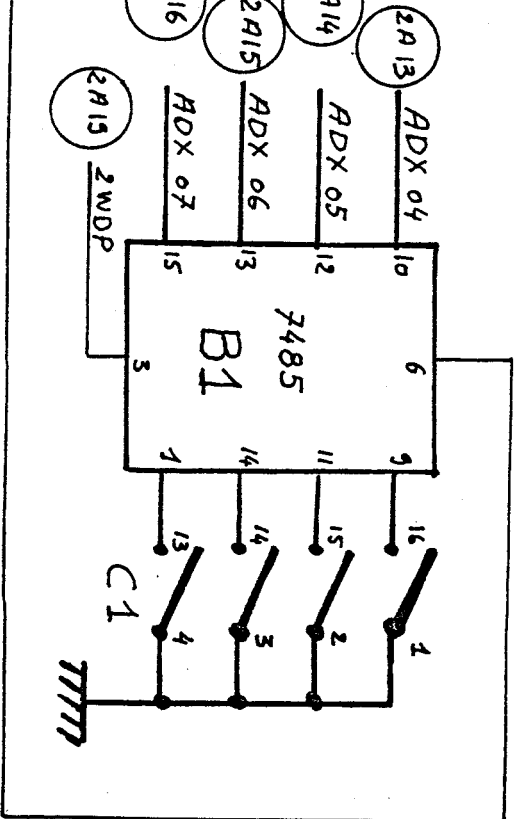


Figure 3 ter

EGA3

Carte N°2



EGA 4

BUS LITTLE

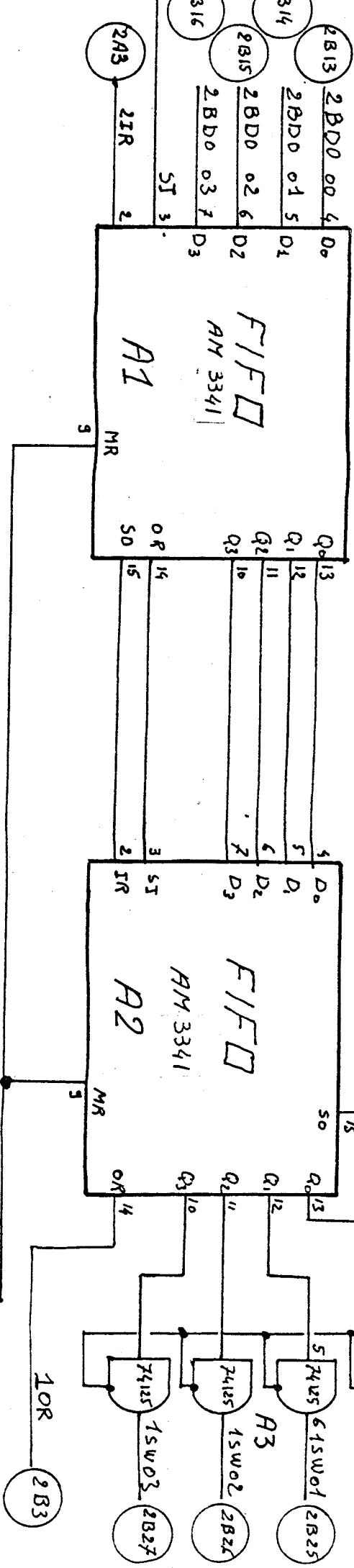
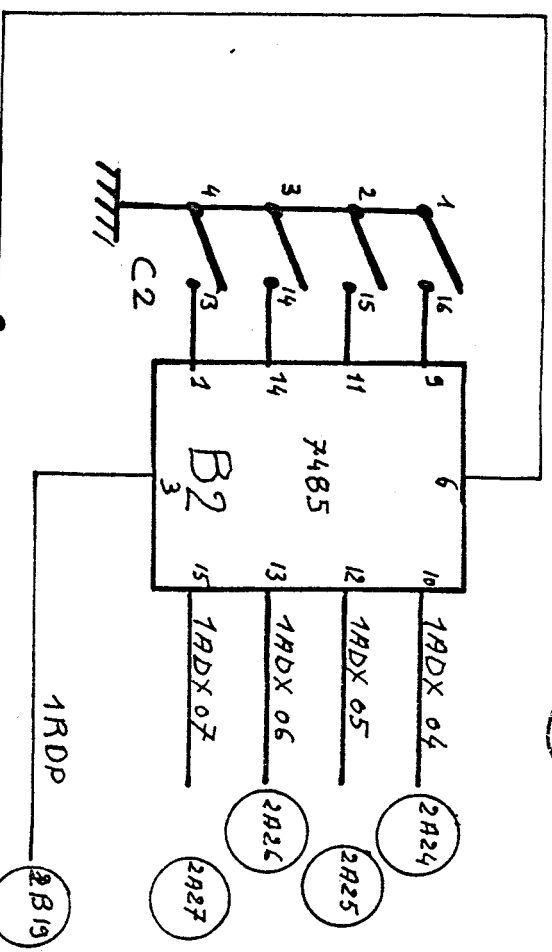


Figure 4

carte N° 2

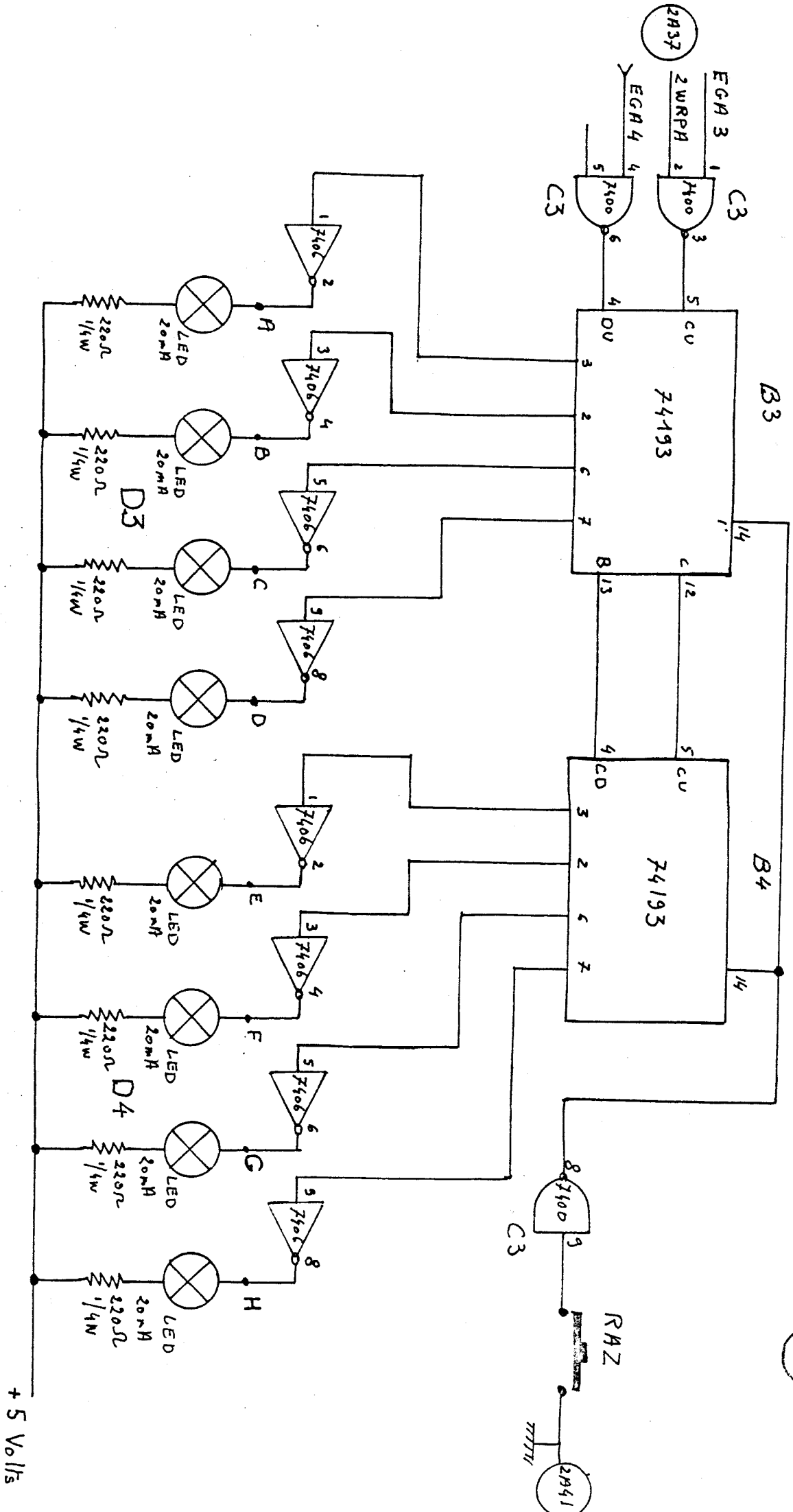


Figure 4 Bis

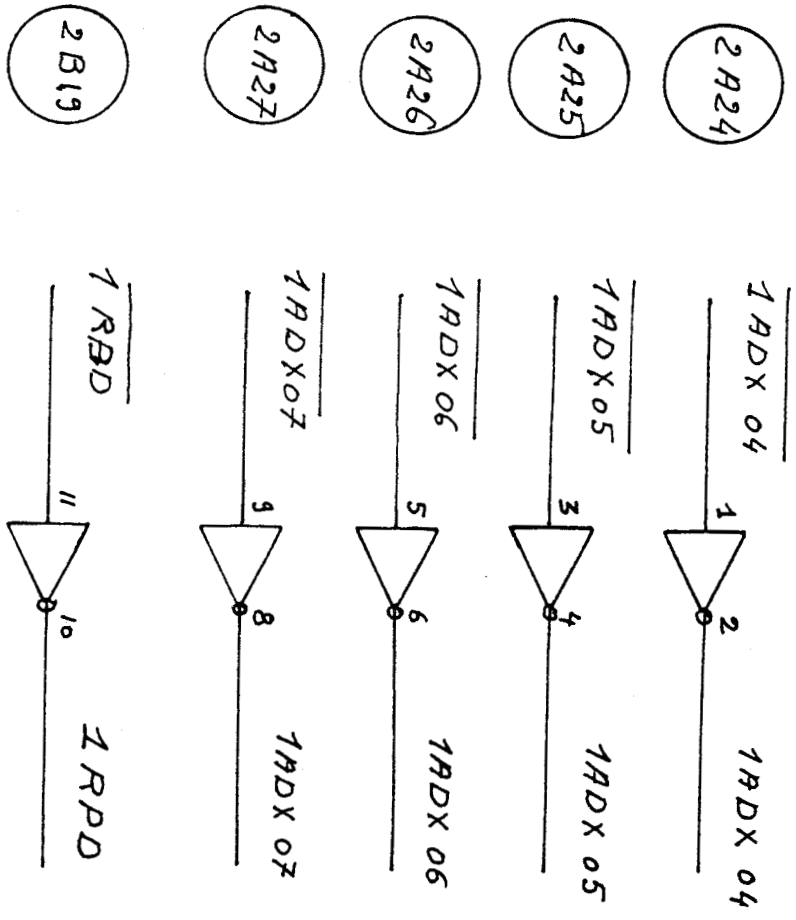


Figure 4 Ter

CONNECTEUR 1

| <u>Côté A</u> | | <u>Côté B</u> | |
|---------------|------------------------|---------------|------------|
| 1 | masse | 1 | masse |
| 2 | | 2 | |
| 3 | 1 IR | 3 | 2 OR |
| 4 | A1 | 4 | E1 |
| 5 | B1 | 5 | F1 |
| 6 | C1 | 6 | G1 |
| 7 | D1 | 7 | H1 |
| 8 | | 8 | |
| 9 | | 9 | |
| 10 | | 10 | |
| 11 | + 5 volts | 11 | + 5 volts |
| 12 | | 12 | |
| 13 | 1 ADX04 | 13 | 1 BDO 00 |
| 14 | 1 ADX05 | 14 | 1 BDO 01 |
| 15 | 1 ADX06 | 15 | 1 BDO 02 |
| 16 | 1 ADX07 | 16 | 1 BDO 03 |
| 17 | | 17 | |
| 18 | | 18 | |
| 19 | 1 WDP | 19 | 2 RDP |
| 20 | | 20 | |
| 21 | | 21 | |
| 22 | masse | 22 | masse |
| 23 | | 23 | |
| 24 | 2 ADX04 | 24 | 2 SW 00 |
| 25 | 2 ADX05 | 25 | 2 SW 01 |
| 26 | 2 ADX06 | 26 | 2 SW 02 |
| 27 | 2 ADX07 | 27 | 2 SW 03 |
| 28 | | 28 | |
| 29 | | 29 | |
| 30 | - 12 volts | 30 | - 12 volts |
| 31 | | 31 | |
| 32 | | 32 | |
| 33 | + 5 volts | 33 | + 5 volts |
| 34 | | 34 | |
| 35 | | 35 | |
| 36 | | 36 | |
| 37 | 1 WRPA | 37 | |
| 38 | | 38 | |
| 39 | | 39 | |
| 40 | | 40 | |
| 41 | RAZ (du panneau avant) | 41 | |
| 42 | | 42 | |
| 43 | masse | 43 | masse |

Figure 5

CONNECTEUR 2

| <u>Côté A</u> | | <u>Côté B</u> | |
|---------------|------------------------|---------------|------------|
| 1 | masse | 1 | masse |
| 2 | | 2 | |
| 3 | 2 IR | 3 | 1 OR |
| 4 | A2 | 4 | E2 |
| 5 | B2 | 5 | F2 |
| 6 | C2 | 6 | G2 |
| 7 | D2 | 7 | H2 |
| 8 | | 8 | |
| 9 | | 9 | |
| 10 | | 10 | |
| 11 | + 5 volts | 11 | + 5 volts |
| 12 | | 12 | |
| 13 | 2 ADX 04 | 13 | 2 BDO 00 |
| 14 | 2 ADX 05 | 14 | 2 BDO 01 |
| 15 | 2 ADX 06 | 15 | 2 BDO 02 |
| 16 | 2 ADX 07 | 16 | 2 BDO 03 |
| 17 | | 17 | |
| 18 | | 18 | |
| 19 | 2 WDP | 19 | 1 RDP |
| 20 | | 20 | |
| 21 | | 21 | |
| 22 | masse | 22 | masse |
| 23 | | 23 | |
| 24 | 1 ADX 04 | 24 | 1 SW 00 |
| 25 | 1 ADX 05 | 25 | 1 SW 01 |
| 26 | 1 ADX 06 | 26 | 1 SW 02 |
| 27 | 1 ADX 07 | 27 | 1 SW 03 |
| 28 | | 28 | |
| 29 | | 29 | |
| 30 | - 12 volts | 30 | - 12 volts |
| 31 | | 31 | |
| 32 | | 32 | |
| 33 | + 5 volts | 33 | + 5 volts |
| 34 | | 34 | |
| 35 | | 35 | |
| 36 | | 36 | |
| 37 | 2 WRPA | 37 | |
| 38 | | 38 | |
| 39 | | 39 | |
| 40 | | 40 | |
| 41 | RAZ (du panneau avant) | 41 | |
| 42 | | 42 | |
| 43 | masse | 43 | masse |

Figure 6

FACE AVANT

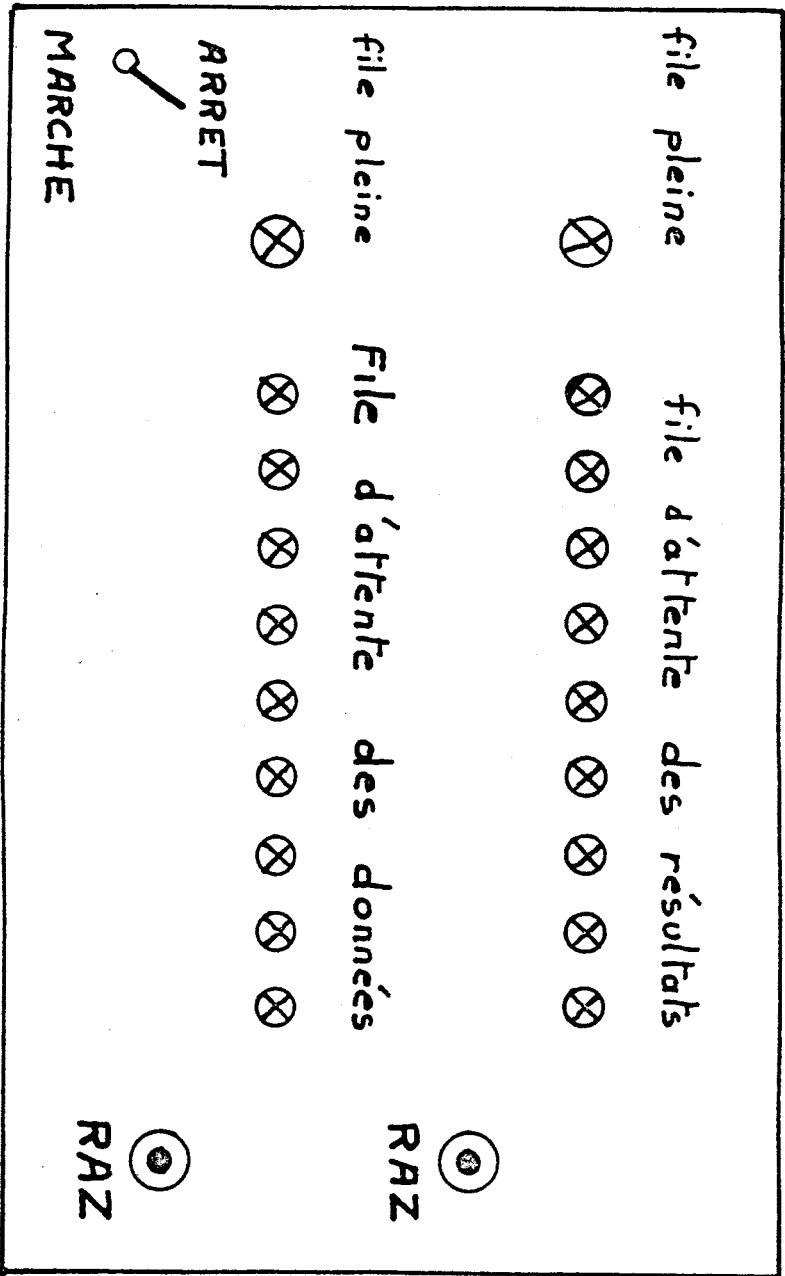


Figure 7

| N° PIN | SIGNAL | DIRECTION |
|--------|----------|----------------------|
| 1 | masse | |
| 2 | 1 IR | C1 - A 3 |
| 3 | masse | |
| 4 | 1 ADX 04 | C1 - A 13, C2 - A 24 |
| 5 | masse | |
| 6 | 1 ADX 05 | C1 - A 14, C2 - A 25 |
| 7 | masse | |
| 8 | 1 ADX 06 | C1 - A 15, C2 - A 26 |
| 9 | masse | |
| 10 | 1 ADX 07 | C1 - A 16, C2 - A 27 |
| 11 | | |
| 12 | | |
| 13 | | |
| 14 | | |
| 15 | | |
| 16 | masse | |
| 17 | 1 WDP | C1 - A 19 |
| 18 | masse | |
| 19 | 1 BDO 00 | C1 - B 13 |
| 20 | masse | |
| 21 | 1 BDO 01 | C1 - B 14 |
| 22 | masse | |
| 23 | 1 BDO 02 | C1 - B 15 |
| 24 | masse | |
| 25 | 1 BDO 03 | C1 - B 16 |
| 26 | | |
| 27 | | |
| 28 | | |
| 29 | masse | |
| 30 | 1 RDP | C2 - B 19 |
| 31 | | |
| 32 | | |
| 33 | | |
| 34 | masse | |
| 35 | 1 SW 00 | C2 - B 24 |
| 36 | masse | |
| 37 | 1 SW 01 | C2 - B 25 |
| 38 | masse | |
| 39 | 1 SW 02 | C2 - B 26 |
| 40 | masse | |
| 41 | 1 SW03 | C2 - B 27 |
| 42 | | |
| 43 | | |
| 44 | | |
| 45 | masse | |
| 46 | 1 OR | C2 - B 3 |
| 47 | | |
| 48 | | |

Figure 8

| N° PIN | SIGNAL | DIRECTION |
|--------|--------------|----------------------|
| 1 | masse | |
| 2 | 2 IR (CJ 12) | C2 - A 3 |
| 3 | masse | |
| 4 | 2 ADX 04 | C2 - A 13, C1 - A 24 |
| 5 | masse | |
| 6 | 2 ADX 05 | C2 - A 14, C1 - A 25 |
| 7 | masse | |
| 8 | 2 ADX 06 | C2 - A 15, C1 - A 26 |
| 9 | masse | |
| 10 | 2 ADX 07 | C2 - A 16, C1 - A 27 |
| 11 | | |
| 12 | | |
| 13 | | |
| 14 | | |
| 15 | | |
| 16 | masse | |
| 17 | 2 WDP | C2 - A 19 |
| 18 | masse | |
| 19 | 2 BDO 00 | C2 - B 13 |
| 20 | masse | |
| 21 | 2 BDO 01 | C2 - B 14 |
| 22 | masse | |
| 23 | 2 BDO 02 | C2 - B 15 |
| 24 | masse | |
| 25 | 2 BDO 03 | C2 - B 16 |
| 26 | | |
| 27 | | |
| 28 | masse | |
| 29 | 2 RDP | C1 - B 19 |
| 30 | | |
| 31 | | |
| 32 | | |
| 33 | | |
| 34 | masse | |
| 35 | 2 SW 00 | C1 - B 24 |
| 36 | masse | |
| 37 | 2 SW 01 | C1 - B 25 |
| 38 | masse | |
| 39 | 2 SW 02 | C1 - B 26 |
| 40 | masse | |
| 41 | 2 SW 03 | C1 - B 27 |
| 42 | | |
| 43 | | |
| 44 | | |
| 45 | masse | |
| 46 | 2 OR CJ 13 | C1 - B 3 |
| 47 | | |
| 48 | | |

Figure 9

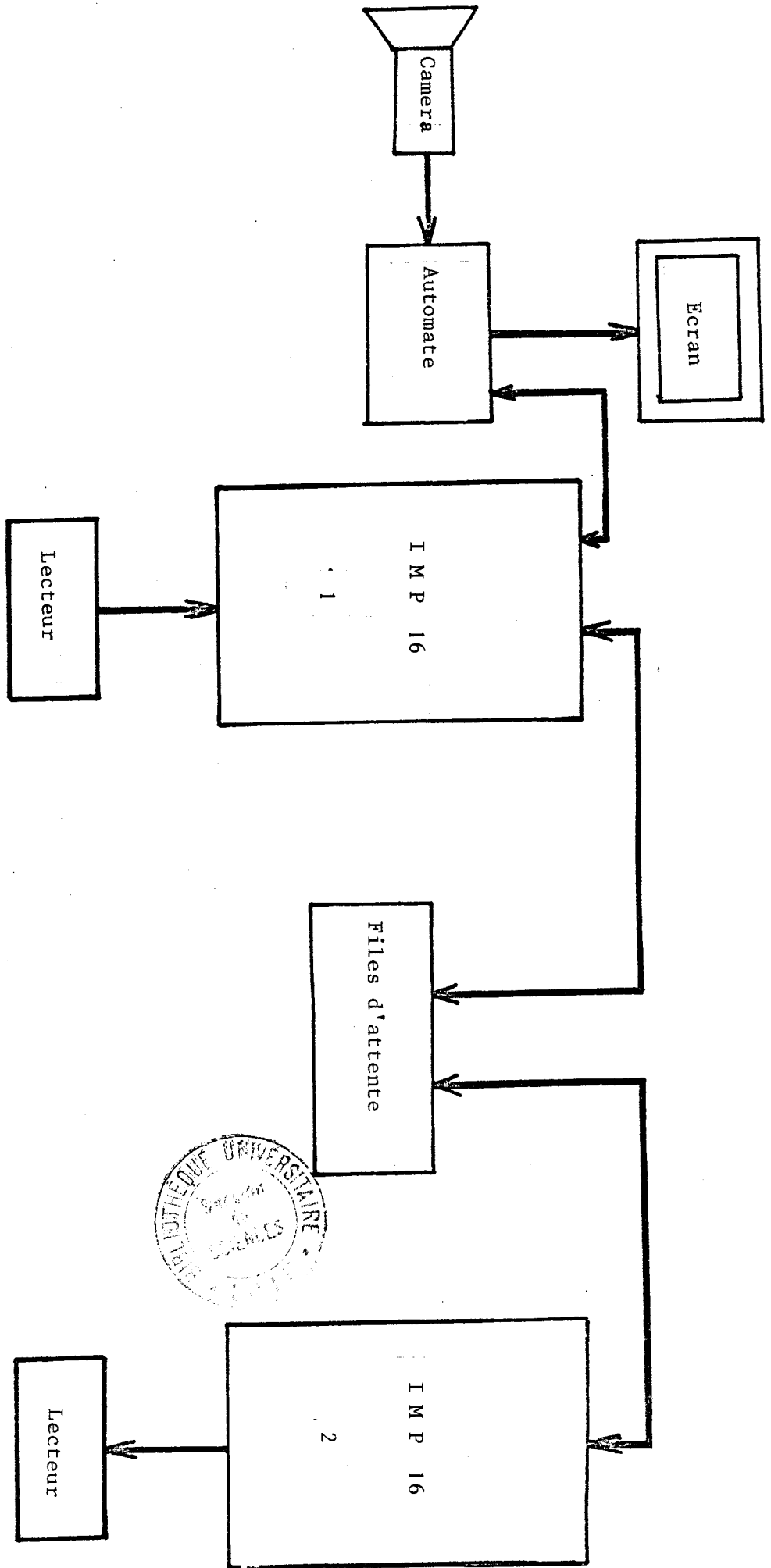


Figure 10

