

INFORMATIQUE
FONDAMENTALE
ET
APPLICATIONS

Comité de
rédaction:

E. Bianco

R. Cusin

P. Isoardi

J.P. Lehmann

R. Stutzmann

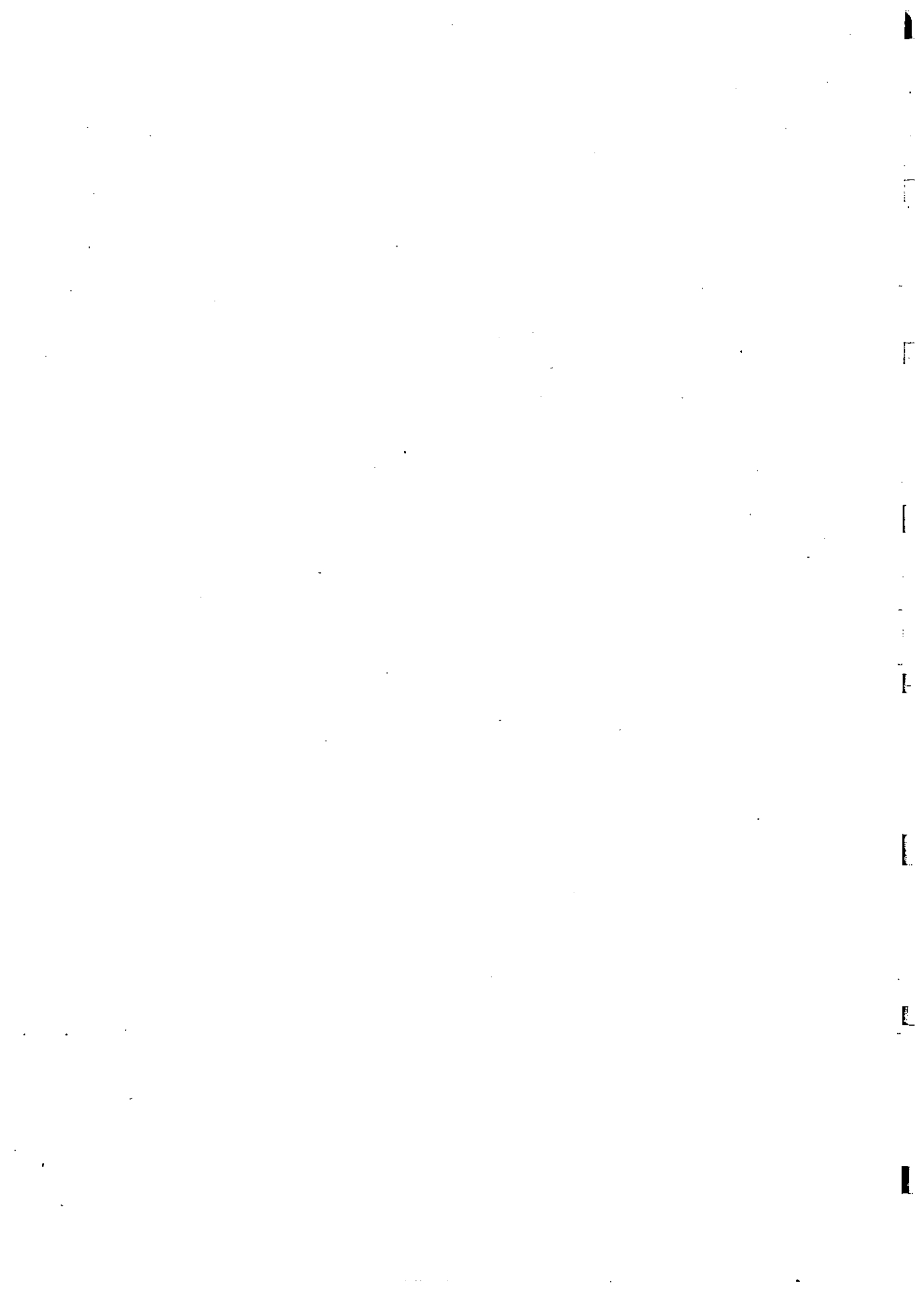
Dépositaire:

G. Ambard

Sommaire

- P 1 -EDITORIAL: Informatique et sport.
- P 5 -Analyse de première approche de langage naturel, dans le cadre de l'étude d'un système support de thésaurus.
- P 25 -Un théorème important de l'informatique.
- P 37 -Langage de traitement de données et de communication.
- P 48 -Glissement d'un hyperpavé dans un champ mémoire de dimension N.
- P 69 -L'ECONOMIQUE ET L'INTELLIGENCE ARTIFICIELLE.
- Première Conférence Internationale

Juin 1985



Editorial

e. bianco

Informatique et Sport.

Soixante mille fois deux cent. Trois millions fois six cent. Voilà de la sémantique à l'état pur. Et la sémantique à l'état pur c'est comme les beaux cristaux, le diamant par exemple, pour en rendre l'éclat soutenable, il faut mettre un peu de chair autour, une jolie femme si vous voulez.

Vous habillez ainsi de quarante morts, le premier cristal de sémantique et vous obtenez le plus beau match de foot du siècle. Vous accolez le mot "avortement" au second et voilà l'une des plus belles affaires commerciales de la médecine moderne officielle.

Le plus beau match de foot ? Le reportage "sportif" m'exaspère car, pour moi, le sport se pratique, il ne se raconte pas, et encore moins ne se commente. Trouver de l'intérêt à apprendre qu'un croquet de quarante-quatre vient de heurter la sphère élastique me paraît suspect, il y a forcément autre-chose au milieu, du non-dit. Mais je reconnais que ce soir-là j'ai écouté l'émission avec intérêt, enfin il se passait quelque chose d'humain. L'affolement presque sincère des commentateurs, leurs sanglots savamment étouffés, et sachant que derrière, là-bas, ça saigne, voilà du vrai spectacle taillé dans le vif.

Il faut bien comprendre la situation, et cela a été suffisamment répété sur les ondes: ce soir-là c'était la fête du football (et du totocalcio, bien sûr). Toute la Police était présente en habit de soirée: le hallebardier s'est tenu fort correctement. Ne confondons pas la soirée de fête avec une simple campagne du maintien de l'ordre comme par exemple les grèves de mineurs des années cinquante, d'abord on n'était pas en tenue de soirée, et puis comme ça ne suffisait pas, les chars de l'armée sont intervenus au canon contre les carreaux des mines. Les mineurs n'avaient aucune chance de mettre à feu et à sang Lille ni même Bruxelles. C'était le bon temps où les ministres du dedans s'appelaient Jules Moch (cela ça ne s'invente pas), et ils étaient déjà roses.

Quand au cristal des avortements, des journalistes incompetents mais pourtant romantiques, ont pu essayer d'expliquer qu'il s'agit d'une mesure humanitaire, que rendre légal l'avortement rendrait également le bonheur dans les familles de la France profonde...

Non: une seule phrase dit toute la vérité et rien que la vérité:

Trois millions fois six cent

et je laisse de côté la TVA.

Madame Veil a tout compris de l'informatique, Monsieur Michel Debré, lui, est du siècle des petits profits individuels.

Mais je ne voudrais pas que l'on puisse croire que je tiens à ranimer de vieux débats éculés. Restons modernes et à l'écoute du Monde, le Monde moderne veut du gâteau et des jeux de cirque, et il veut participer aux jeux du cirque. Etre un peu acteur, quoi!

Or, plus ça lui coûte cher moins il participe, perché dans des gradins qui ont déjà servi et pourront servir encore à lui faire goûter un autre genre de spectacle.

Voir les autres se faire bouffer par les lions est un spectacle d'une rare délicatesse qui procure des frissons et d'horribles cauchemars délicieux.

L'ennui pour les esclaves c'est qu'ils rêvent la liberté, qu'ils chantent la liberté, parfois avec de belles voix sur de belles paroles, mais le jour où ils croient la détenir, ils n'ont d'autre hâte que de la remettre entre des mains plus expertes estiment-ils. Et alors toutes les Jeanne-d'Arc à petite moustache et mèche sur le côté sont les bienvenues qui savent montrer d'un doigt vengeur les véritables fautifs.

La liberté n'est pas un produit manufacturé, et je crains fort que les petites libertés de grande série qu'on nous offre à propos d'un petit maillot féminin fort seyant, d'un voyage aux Baléares, d'un herbicide-qui-enfin-a-raison-du-liseron-qui-rendait-fou-mon-Grand-Père, du parti de l'Opposition qui a enfin solution à nos maux: "regardez mon sourire, Moi enfin Je vous écoute", ne soient que de tout petits expédients pour tenter d'équilibrer notre balance commerciale.

La force de l'illusion n'est-elle pas un baume capable d'atténuer -de guérir diraient les médocastres- ces douleurs qui nous rongent.

Que l'ancienne minorité ait pu étaler outrageusement son incompetence laisse-t-il encore la place à un quelconque placebo ? Allons nous voila prêts pour pénétrer de plein pied dans cette grande illusion de liberté: les vacances.

ANALYSE DE PREMIERE APPROCHE DU LANGAGE NATUREL
DANS LE CADRE DE L'ETUDE D'UN SYSTEME SUPPORT DE THESAURUS

*m.t. laskri
ph galand*

C.R. Subject classification informatics : H31

Résumé

Dans le cadre de la conception de système de communication homme-machine en langage naturel, cet article présente l'étude de la première partie d'un automate permettant d'analyser toute phrase contenue dans un texte figurant dans un cadre thématique déterminé. Cette analyse a pour but de générer une représentation intermédiaire de toute forme de phrase possible afin de constituer le thesaurus correspondant.

ANALYSE DE PREMIERE APPROCHE DU LANGAGE NATUREL
DANS LE CADRE DE L'ETUDE D'UN SYSTEME SUPPORT DE THESAURUS

INTRODUCTION

Il s'agit d'un projet de conception d'un système de communication homme-machine en langage naturel qui servira de support pour la construction de thesaurus utilisables dans différents domaines d'application.

Ce système peut se partager entre plusieurs tâches, en commençant par la phase d'analyse d'un texte et son enregistrement pour constituer un "fonds d'informations", jusqu'à l'exploitation de celui-ci comme l'établissement d'un dialogue questions-réponses.

L'objet de cet article est d'esquisser la méthode d'analyse d'un texte en la replaçant dans son cadre général, c'est-à-dire la construction d'un thesaurus. La conception d'une telle construction présentera deux phases qui permettront d'aboutir à un instrument "d'enrichissement réciproque" :

- une phase de "réflexion" où il faudra se mettre à la place du spécialiste du domaine concerné et se créer le souci de sa "recherche" ;
- et une phase de "réalisation" comprenant principalement un travail d'informaticien visant à une mise en oeuvre et une application souple.

L'étude envisagée nous a d'abord amenés à réfléchir sur la notion de thesaurus telle qu'elle a été définie dans la norme NF Z47-100 [1]. Cette norme existant dans un cadre restreint, celui de la recherche documentaire, il nous a paru intéressant de nous pencher sur l'apport du langage naturel comme support d'informations. Nous nous sommes donc posé le problème de la systématisation de prise en compte de la langue vivante. Pour cela, il fallait appréhender les processus d'analyse et de compréhension d'un texte. Le problème de reconnaissance de la signification d'un texte n'est pas facile. Un énoncé n'est pas en relation univoque avec son sens. Seul, l'usage de la langue le laisse pressentir. Sans devoir définir une sorte de langage usuel minimum, il s'agit de constituer une base pour le traitement de l'information dans des textes écrits en français. Le présent exposé se contentera d'en définir une partie. Celle-ci sera constituée de :

- la présentation générale du processus de constitution du fonds sémantique dans un cadre thématique déterminé (chap. I)
- et l'analyse de première approche (incluant quelques exemples d'algorithmes) (chap. II).

NOTION DE THESAURUS

Elle a été définie en 1978 par l'AFNOR [1] mais celle-ci n'a pour but que la documentation automatique [2]. La documentation automatique porte sur des objets tels que "livres", "articles", "revues", "bulletins"... Elle nécessite la constitution d'un fonds documentaire et sa gestion dans le but d'une consultation du fonds. L'informatisation est favorisée par une normalisation. Ce qui a conduit à l'établissement d'une norme de thesaurus, indépendante de tout système documentaire, qui consiste à décrire les principes d'une indexation automatique :

- analyse d'un document par appréhension de son contenu, identification des concepts représentant ce contenu et sélection des concepts nécessaires à une recherche ultérieure ;
- organisation de l'information dans le langage documentaire par la reconnaissance des descripteurs permettant de représenter fidèlement les concepts sélectionnés dans les documents.

Le thesaurus se définit comme un moyen de réaliser cette indexation à l'aide d'un vocabulaire contrôlé et dynamique de termes (descripteurs) obéissant à des règles terminologiques propres et reliés entre eux par des relations sémantiques. Le thesaurus permet donc de traduire en termes d'indexation ou en termes de recherche tout concept devant entrer ou sortir d'un système documentaire donné.

La recherche dans la chaîne documentaire s'effectue aujourd'hui dynamiquement par indexation de la demande, s'il s'agit d'un thesaurus, ou par mots-clés (c'est-à-dire des clés d'accès permettant de retrouver le ou les documents cherchés), dans le cas d'un lexique ou dictionnaire organisé.

On le voit, celle-ci reste limitée. L'intérêt de notre étude s'est porté sur la notion générale d'information pour aboutir à celle de langue vivante.

Langage naturel

Le langage naturel ou plus commodément la langue vivante est un support intéressant mais à prise difficile. En effet, la notion d'information dans une langue morte ou langue technique reste figée, donc contraignante. Par contre, la langue vivante présente l'inconvénient important d'être particulière à chaque individu. D'autre part, sa maîtrise reste délicate par le fait du manque d'un métalangage permettant de l'analyser. Le problème était donc la prise en compte systématique, dans un cadre thématique déterminé, de la langue vivante.

Analyse et compréhension

"Analyser ne veut pas dire comprendre". Analyser une phrase consiste à dégager les concepts des mots de la phrase au delà d'une simple lecture car, en effet, la correspondance entre le sens d'un énoncé et sa formulation est très généralement complexe.

Comprendre une phrase consiste à transformer une phrase écrite en langage naturel en une représentation interne adaptée à une base de connaissance (thesaurus) et manipulable au moyen d'algorithmes.

On dira qu'un système informatique a compris un texte s'il est capable de répondre à toutes sortes de questions le concernant [3].

CHAPITRE I - PRESENTATION GENERALE DU PROCESSUS
DE CONSTITUTION DU FONDS SEMANTIQUE

Avant de présenter sommairement la phase de constitution du "fonds sémantique" (Cf I.1-k), donnons d'abord quelques définitions très utiles pour la suite.

I.1 - DEFINITIONS

- a) Système support de thesaurus : un système support de thesaurus est une structure et un algorithme permettant de l'explorer et pouvant la faire évoluer dans un contexte interactif.

- b) Analyse de première approche : c'est l'analyse des phrases à enregistrer dans le système en tenant compte de leurs structurations sémantiques.

- c) Thesaurus primaire : le thesaurus primaire représente l'ensemble des listes (liste des noms propres au domaine d'application, liste des verbes, etc...), c'est-à-dire le lexique intervenant dans l'analyse de première approche.

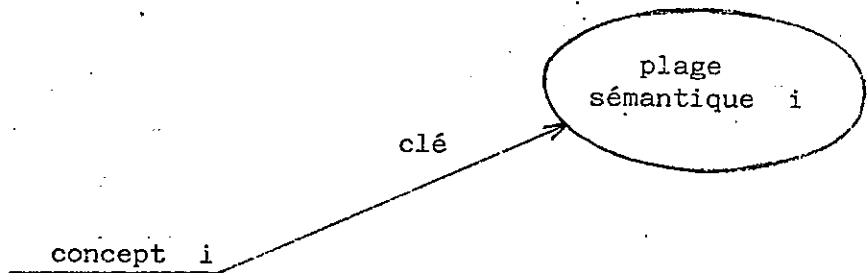
- d) Schéma intermédiaire : c'est la représentation résultante de l'analyse de première approche d'une phrase à analyser.

e) Plage sémantique :

élément de connaissance et de perception, isolable et cernable par des jeux de concepts plus ou moins complexes.

f) Concept :

un concept représente toutes les notions appartenant à une même plage sémantique ; est utilisé comme une clé d'accès à la plage sémantique correspondante.



Remarque : les concepts sont tirés des mots mais représenteront des combinaisons de termes (ex. : "bonne affaire", "séjour d'altitude").

g) Lien sémantique :

un lien sémantique établit une relation entre deux concepts

concept i Rij concept j

h) Chemin sémantique :

un chemin sémantique représente l'assemblage ordonné de liens sémantiques

concept i Rij concept j Rjk concept k ...

- i) Thesaurus secondaire : Le thesaurus secondaire se compose des différents concepts nécessaires pour la réalisation de tous les chemins sémantiques.
- j) Analyse de deuxième approche : c'est l'analyse conçue pour la réalisation du chemin sémantique correspondant au schéma intermédiaire dérivé de l'analyse de première approche.
- k) Fonds sémantique : le fonds sémantique se compose de tous les chemins sémantiques constitués par l'analyse de deuxième approche, c'est-à-dire c'est le stock d'informations exploitable dans un cadre thématique déterminé.

I.2 - CONSTITUTION DU FONDS SEMANTIQUE

En soumettant une phrase rédigée en langage naturel au système, nous lui appliquons l'analyse de première approche qui fait référence au thesaurus primaire pour construire le schéma intermédiaire associé.

L'analyse de deuxième approche appliquée à ce schéma intermédiaire, et en faisant appel au thesaurus secondaire, nous donne le chemin sémantique correspondant.

Ensuite, il ne reste plus qu'à enregistrer ce chemin sémantique dans le fonds sémantique.

Le schéma suivant représente la constitution du fonds sémantique :

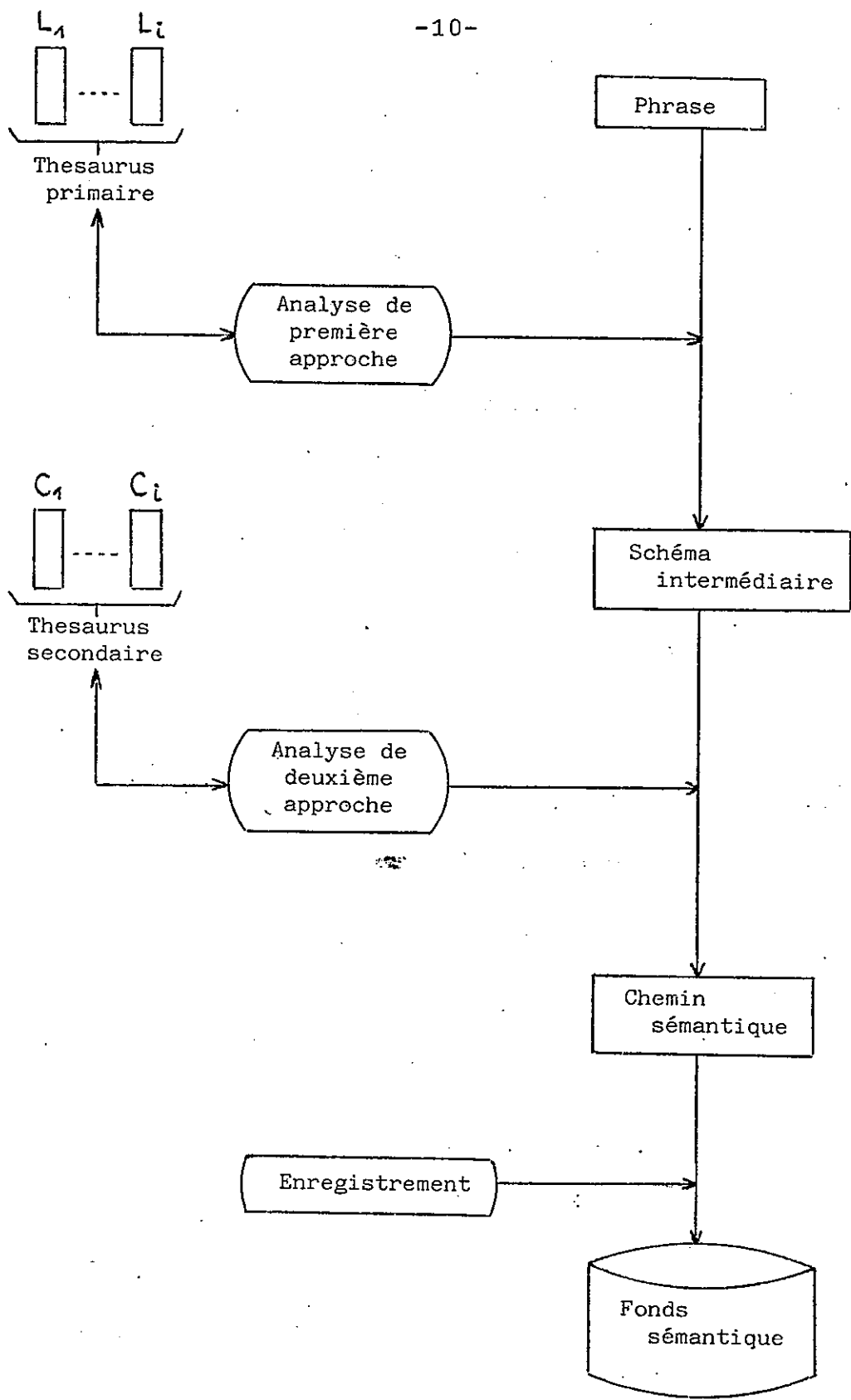


fig I.2 : CONSTITUTION DU FONDS SEMANTIQUE

CHAPITRE II - ANALYSE DE PREMIERE APPROCHE

Un nombre important de recherches en analyse automatique du langage naturel se distingue par l'utilisation de la théorie des grammaires génératives [4]. Cependant, il est à constater que se pose le problème de l'intégration du contexte pour guider l'analyse.

II.1 - EXPOSE DE LA METHODE

Afin d'analyser toute phrase formulée en langage naturel, nous avons défini une représentation assez générale permettant d'engendrer son schéma intermédiaire. Cette représentation générale n'est autre que le schéma intermédiaire le plus complet pouvant représenter une phrase comportant toutes les formes de phrases que nous avons pu dégager.

a) Formes des phrases et exemples (*)

En général, une phrase du langage naturel se compose d'un groupe nominal suivi d'un groupe verbal-complément. Cependant, le groupe nominal peut être plus ou moins complexe :

- il peut contenir une phrase incidente précédant le groupe nominal réel telle que :
"Composée de treize membres nommés par le gouvernement, elle..."
"Changeant d'ailleurs de politique, le Royaume Uni..."
- il peut commencer par un adverbe tel que :
"Enfin, la Cour de Justice de Luxembourg..."
"Mais, inquiet du dynamisme des Six, il..."
"Quand les cours du marché mondial..."

(*) les exemples sont tirés de [5]

- ou il peut contenir une phrase incidente succédant au groupe nominal réel telle que :

"Le Traité de Rome, signé par les Six le 25 mars 1957,..."

"La Commission, qui provient de la fusion..., ..."

Le groupe verbal-complément peut se composer :

- d'un groupe verbal suivi éventuellement d'un ou plusieurs groupes compléments tel que :

"La création d'un marché unique permet d'accroître considéra-
GV GC1

blement la puissance sidérurgique."
GC2

"La préférence communautaire favorise la vente des productions
GV GC

européennes."

- de plusieurs groupes verbal-complément tels que :

"Le Marché Commun a pour but de créer une union douanière,
GV+GC1

d'harmoniser les politiques économiques, et de faire disparaître
GV+GC2

les inégalités sociales".
GV+GC3

Bien entendu, le groupe complément peut contenir à son tour :

- une phrase incidente précédant le groupe complément réel ;

- une phrase incidente succédant au groupe complément réel.

Enfin, une phrase peut contenir plusieurs phrases articulées par des ponctuations, autres que le point final, ou par des coordinations :

- "Un désarmement douanier, aussi complet que celui des Six, était prévu mais chacun conservait l'entière liberté de ses tarifs à l'égard du tiers."

Ainsi, nous dégageons une représentation générale pour toute phrase quelle que soit sa forme discutée ci-dessus.

D'où le schéma intermédiaire le plus général pouvant être généré par l'analyse de première approche pour représenter les phrases.

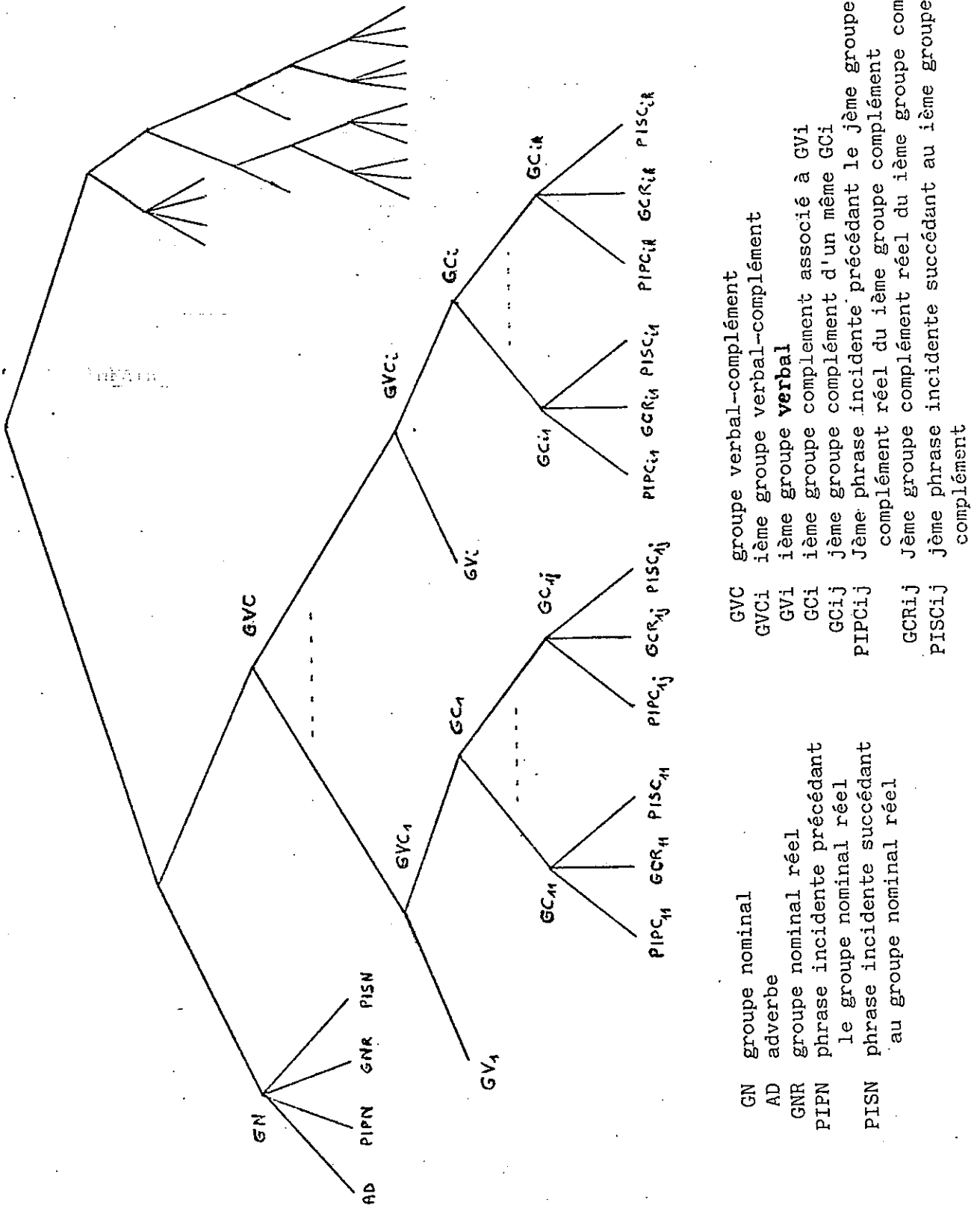


fig II.1 : SCHEMA INTERMEDIAIRE GENERAL

b) Fonctions de l'analyse de première approche

Analyser correctement une phrase revient donc à générer sans ambiguïté son schéma intermédiaire. Ainsi, lors de cette analyse, il est raisonnable de prévoir que l'automate que nous construisons peut se tromper de temps en temps. Mais nous le construisons de manière à ce qu'il puisse se corriger ; pour cela, il faut lui apporter une quantité fantastique d'informations.

Cet automate communique avec le thesaurus primaire pour définir le sens des termes de la phrase à analyser pour spécifier qu'il s'agit d'un terme propre au domaine d'application, d'un verbe, d'un pronom personnel, d'un adjectif possessif, etc... Cette analyse nous permet donc de savoir si on traite un terme du groupe nominal, du groupe verbal ou du groupe complément afin de générer correctement le schéma intermédiaire (cf fig. II.1).

En plus de la fonction de la formulation des phrases et de la fonction de génération de leurs schémas intermédiaires, l'analyse de première approche tient compte de leurs structurations sémantiques.

Cette analyse transforme certains types de phrases en une forme dont la représentation est prévue dans le schéma intermédiaire général. Ainsi, la phrase suivante :

"C'est en 1972 que les Six de l'Europe Occidentale constituent la Communauté Economique Européenne"

sera transformée en :

"Les Six de l'Europe Occidentale constituent la Communauté Economique Européenne en 1972."

Remarque : le fait de changer l'expression d'une telle phrase modifie bien entendu son contenu, mais d'une manière telle que nous pouvons considérer qu'elle soit imperceptible si l'on veut prendre en compte la totalité du texte.

Pour lever l'ambiguïté de référence par un pronom personnel, un adjectif possessif ou un adjectif démonstratif, cette analyse a la charge de dupliquer le mot ou le groupe de mots auquel fait référence le pronom personnel, l'adjectif possessif ou l'adjectif démonstratif.

Exemple :

"Changeant d'ailleurs de politique, le Royaume Uni n'hésita pas à solliciter son adhésion au Marché Commun."

deviendra :

"Changeant d'ailleurs de politique, le Royaume Uni n'hésita pas à solliciter l'adhésion du Royaume Uni au Marché Commun."

Pour lever l'ambiguïté dans la structuration des phrases, par exemple, pour reconnaître qu'un groupe complément appartient à la phrase incidente succédant au groupe complément principal, ou à la phrase principale, l'automate, lorsqu'il rencontre ce cas, avance dans l'analyse afin de reconnaître ce qui suit pour qu'il puisse prendre une décision. Ensuite, il revient à l'endroit où il a rencontré l'ambiguïté pour continuer normalement.

Enfin, pour fournir une extensibilité au système, l'automate réalisant l'analyse de première approche, permet d'enregistrer les termes qui n'existent pas encore dans le thesaurus primaire et dans le thesaurus secondaire, et de reprendre ensuite l'analyse ; cette fonction se réalisant avec l'accord de l'utilisateur (nous ne tenons pas encore compte des fautes d'orthographe).

II 2 - ALGORITHMES REALISANT CETTE ANALYSE

Nous donnons ici quelques algorithmes qui réalisent cette analyse de première approche pour le groupe nominal. Ces algorithmes sont écrits dans un langage conçu pour la spécification de l'analyse fonctionnelle des applications [6].

PROC PR-groupe-nominal * traitement d'un groupe nominal général *

FAIRE

SELON

CAS * On reconnaît un adverbe * PR-adverbe

CAS * On débute le groupe nominal réel par une incidence le
précédant * PR-incidence-préd
PR-signaler-gnr
* signaler l'attente d'un groupe
nominal réel *

CAS * On débute le groupe nominal réel *
PR-groupe-nominal-réel.

CAS * On débute une incidence succédant au groupe nominal
réel par "qui", "que", "dont"... *
PR-incidence-succ
PR-signaler-gvc
* signaler l'attente d'un groupe verbal-
complément *

FINSELON

SI * il y a eu arrêt de l'analyse * ALORS SORTIR

SINON SI * fin du groupe nominal général * ALORS SORTIR

SINON REFAIRE

FINSI

FINSI

FINFAIRE

FINPROC

PROC PR-incidence-préd * traitement d'une incidence précédant soit le
groupe nominal réel soit un groupe complément *

FAIRE

SELON

CAS * il s'agit d'un adjectif possessif *
PR-trait-possessif

CAS * il s'agit d'un adjectif démonstratif *
PR-trait-démonstratif

CAS * terme inconnu * /* ne rien faire */

AUTRE-CAS * il s'agit d'un autre terme *

PR-accrocher-terme-schéma-courant

* placer le terme courant dans le schéma
intermédiaire courant à son emplacement *

PR-acquérir-terme-suivant

* acquisition du terme suivant *

PR-analyser-sens-terme

SI * le terme est inconnu *

ALORS PR-signaler-arrêt-analyse

FINSI

FINSELON

SI * il y a eu arrêt de l'analyse * ALORS SORTIR

SINON SI * fin de l'incidence * ALORS SORTIR

SINON REFAIRE

FINSI

FINSI

FINFAIRE

FINPROC

PROC PR-incidence-succ * traitement d'une incidence succédant au groupe nominal réel ou au groupe complément *

FAIRE

SELON

CAS * on est dans le cas où on a une incidence qui commence par un adverbe d'incidence, par un participe passé, par un participe présent *

PR-accrocher-terme-schéma

PR-acquérir-terme-suivant

PR-analyser-sens-terme

SI * le terme est inconnu *

ALORS PR-signaler-arrêt-analyse

SINON PR-incidence-pred

FINSI

CAS * l'incidence commence par un "qui" *

PR-accrocher-terme-schéma

PR-acquérir-terme-suivant

PR-analyser-sens-terme

SI * le terme est inconnu *

ALORS PR-signaler-arrêt-analyse

SINON PR-groupe-verbal-complément

FINSI

CAS * l'incidence commence par un "que" *

PR-accrocher-terme-schéma

PR-acquérir-terme-suivant

PR-signaler-sens-terme

SI * terme inconnu *

ALORS PR-signaler-arrêt-analyse

SINON PR-groupe-verbal-complément-groupe-nominal

FINSI

CAS * l'incidence commence par un "dont" *

PR-accrocher-terme-schéma

PR-acquérir-terme-suivant

PR-analyser-sens-terme

SI * terme inconnu *

ALORS PR-signaler-arrêt-analyse

SINON PR-phrase-complète

FINSI

FINSELON

FINFAIRE

FINPROC

PROC PR-groupe-nominal-réel

FAIRE

SELON

CAS * il s'agit d'un pronom personnel sujet ou d'un adjectif
démonstratif sujet tel que "celui-ci" *

PR-trait-pronom-personnel

CAS * il s'agit d'un possessif *

PR-trait-possessif

CAS * il s'agit d'un démonstratif *

PR-trait-démonstratif

CAS * terme inconnu * /* ne rien faire */

AUTRE-CAS * il s'agit d'un autre terme *

PR-accrocher-terme-schéma

PR-acquérir-terme-courant

PR-analyser-sens-terme

SI * terme inconnu *

ALORS PR-signaler-arrêt-analyse

FINSI

FINSELON

SI * il y a eu un arrêt de l'analyse * ALORS SORTIR

SINON SI * fin du groupe nominal réel *

ALORS

SELON

CAS * terme courant est un verbe *

PR-signaler-GVC

CAS * début d'une incidence *

PR-signaler-incidence-succ

FINSELON

SORTIR

SINON REFAIRE

FINSI

FINSI

FINFAIRE

FINPROC

II.3 - EXEMPLE D'APPLICATION

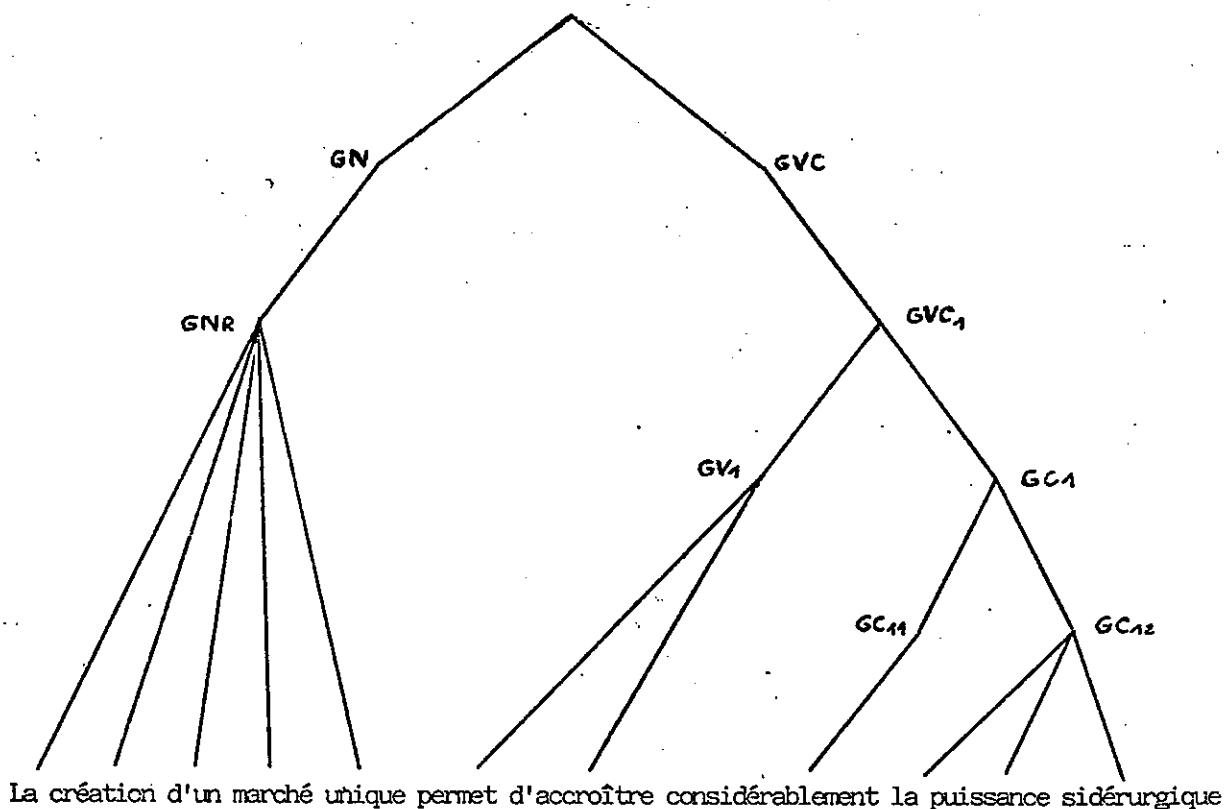
Essayons maintenant d'appliquer cette analyse de première approche à la phrase suivante :

"La création d'un marché unique permet d'accroître considérablement la puissance sidérurgique."

A travers notre analyse, nous pouvons découper cette phrase de la façon suivante :

"La création d'un marché unique" constitue le groupe nominal
"permet d'accroître" constitue le groupe verbal
"considérablement" représente le premier groupe complément
et "la puissance sidérurgique" constitue le deuxième groupe complément.

D'où le schéma intermédiaire généré par l'analyse :



BIBLIOGRAPHIE

- [1] Association Française de Normalisation AFNOR 1981
NF Z47-100 "Règles d'établissement de thesaurus monolingues"
AFNOR 1978
NF Z47-102 "Principes généraux pour l'indexation des documents"
- [2] Georges VAN SLYPE, Michel VAN DYK, Marcel GUILLOT
"Systèmes documentaires et ordinateur"
Les Editions d'Organisation Paris 1973
- [3] J. FERBER
"La compréhension automatique de textes"
Artefact Microsystèmes n° 46 1984
- [4] M. ICHBIAH, H. VALA BREGUE, LEVERY
"Analyse syntaxique automatique des phrases françaises"
Informatique et Gestion pp 127-136 1970
- [5] "Les Grandes Puissances Economiques"
Librairie Armand Colin Paris 1978
- [6] Institut de Formation et de Conseil en Informatique
"Guide pour la rédaction du Cahier d'Expressions des Besoins"
SYDAM 1982

THEORIE DE L'INDEX

e. bianco

C.R. Subject classification informatics: D31

Résumé.

Il est montré comment simuler le jeu d'un index avec une machine élémentaire qui n'en possède pas.

UN THEOREME IMPORTANT

DE

L'INFORMATIQUE

Dans nos Sociétés hautement policées, une branche de la connaissance habillée de mondanités, et qui ne comporterait pas quelques brassées de Théorèmes, Vérités Universelles, comme autant de boutons d'or et de nacre, ne ferait pas sérieux.

L'Informatique a trop longtemps baigné dans le cambouis de l'intuition, vêtue d'une salopette bleue de chauffe, avec ses traces de mains noires sur les hanches, pour qu'on ne fasse pas l'effort de la débarbouiller un peu avant de l'amener au bal. Et puisqu'il s'agit du calcul sur le nom, montrons comment est apparue la particule.

THEORIE DE L'INDEX.

L'invention de la machine à cases adressables (Machine de Nolin) présente l'avantage par rapport à la machine de Turing, de permettre l'accès direct à la case. Le gain obtenu, essentiellement d'ordre algorithmique, se retrouve dans l'économie de ces machines de Turing indispensables pour voyager, aller-retour, d'un paquet d'information à un autre.

L'introduction d'une notion supplémentaire: la notion de procédure, montre qu'en passant de la machine de Turing à la machine à cases adressables, on a toutefois perdu quelque chose de bien

commode: l'indépendance de l'algorithme par rapport à la localisation de la configuration. C'est pour rattraper cet avantage sans perdre pour autant celui qu'apporte la machine à cases adressables, qu'on introduit une nouvelle machine: la procédure formelle.

Je rappelle rapidement ce qu'est la procédure formelle. La mémoire centrale sur laquelle on travaille est constituée de k cases, auxquelles il faut ajouter la case w et la case I. La case I ne diffère que par le rôle qu'elle joue, des autres cases de la mémoire, et leur contenu est au maximum de k. Les instructions qui travaillent exclusivement sur ces cases-là, et ce sont les seules qui nous intéressent ici ont pour structure:

x := 0

x := y

si y = 0 vers ei

x := y +_k 1

vers ej

Nous verrons qu'il est possible de rajouter tout de suite:

x := y Δ y

si y \mathbb{R} y vers ei

Les opérandes x ont la forme possible:

$$x \left\{ \begin{array}{l} \alpha \\ \alpha, \beta \\ I \\ \omega \end{array} \right.$$

Les opérandes y ont la forme x, et en plus, peuvent être une constante.

En bref, à la machine à cases adressables, j'ajoute une case de plus dont le rôle est spécial: son contenu est le nom d'une case. Ceci impose une condition: si je veux calculer sur I comme sur toute autre case, son contenu doit être de même nature.

Si k est le contenu maximum de ces cases et K le nombre de cases

standard, il faut évidemment que:

$$k \geq K$$

pour pouvoir atteindre toutes les cases.

Alors se pose le problème de l'équivalence de ces deux machines. On voit sans peine que la procédure formelle a, au moins, la puissance de la machine à cases adressables. Il suffit de figer I à zéro, et tout algorithme de la machine à cases adressables possède aisément un équivalent en termes de procédure formelle. La réciproque est moins évidente.

Je me donne une procédure formelle avec une mémoire de K cases. Je construis en face, une machine à cases adressables de k cases avec: $k = K + 6$, et je réserve aux 6 cases supplémentaires les rôles suivants:

case $k - 1$	Contient la valeur de I de la PF.
$k - 2$	Contient successivement α et β .
$k - 3$	Calcul de l'adresse.
$k - 4$	1 ^{ère} valeur
$k - 5$	2 ^{ème} valeur
$k - 6$	3 ^{ème} valeur
$K - 1$	Dernière case, image de la case $K-1$ de la procédure formelle.

Je vais d'abord montrer que je sais construire un algorithme qui établit une correspondance entre le contenu d'une case et le nom d'une autre case. En case $(k - 3)$ je place une valeur quelconque que je veux considérer comme une adresse. Je vais essayer d'atteindre le contenu de la case qui a pour nom la valeur en question et je le place en $(k - 4)$. Bien sûr, j'ignore a priori le contenu de $(k - 3)$.

<u>si</u> $c(k-3) = 0$ <u>vers</u> e_0	$e_0 : c(k-4) := c_0$
$c(k-3) := c(k-3) -_k 1$	<u>vers</u> ef
<u>si</u> $c(k-3) = 0$ <u>vers</u> e_1	$e_1 : c(k-4) := c_1$
$c(k-3) := c(k-3) -_k 1$	<u>vers</u> ef
<u>si</u> $c(k-3) = 0$ <u>vers</u> e_2	$e_2 : c(k-4) := c_2$
---	<u>vers</u> ef
---	---
$c(k-3) := c(k-3) -_k 1$	<u>vers</u> ef
<u>si</u> $c(k-3) = 0$ <u>vers</u> e_{k-1}	$e_{k-1} : c(k-4) := c(k-1)$
<u>vers</u> ef	$ef : suite...$

Le principe de cet algorithme est simple, on explore tous les contenus possibles de la case (k-3), et pour chaque valeur trouvée on effectue le transfert en k-4 du contenu de la case dont le nom a pour valeur celle précisément qu'on vient de trouver. Bien sûr un tel algorithme peut toujours être écrit dans la mesure où le nombre de valeurs que contient une case est borné.

Je désigne par: $\overset{k-3}{x \rightarrow k-4}$ cet algorithme.

Je peux construire également des algorithmes qui construisent des constantes sur le principe suivant:

```

cx := 0
cx := cx +_k 1
cx := cx +_k 1
cx := cx +_k 1
    
```

cet exemple met la constante 3 dans la case N°x. Je désigne un tel algorithme par: K_α^x .

Par exemple: K_5^{k-2} qui place 5 en (k-2).

Je peux maintenant écrire l'algorithme image de $[_{\alpha, \beta}$. Je rappelle que cette forme représente le contenu de la case dont l'adres-

se est ainsi obtenue: en adresse $[I + \alpha]$, on récupère une valeur, mettons γ , qui est considérée comme une adresse, et cette valeur corrigée de β donne à partir de l'origine $[I]$ l'adresse de la case dont on prend le contenu.

Par construction $0 \leq \alpha, \beta \leq K-1$ donc:

K_α^{k-2} , $c(k-3) := c(k-1) +_k c(k-2)$, calcule $[I + \alpha]$
 $r_{x \rightarrow (k-3)}^{(k-3)}$, K_β^{k-2} , γ dans $(k-3)$
 $c(k-3) := c(k-3) +_k c(k-1)$,
 $c(k-3) := c(k-3) +_k c(k-2)$, $[I + \gamma + \beta]$ dans $(k-3)$
 $r_{x \rightarrow (k-4)}^{k-3}$ place le $[_{\alpha, \beta}$ dans $(k-4)$

Je désigne alors plus généralement l'algorithme que je viens d'écrire par la forme: $T_{k-4}^{\alpha, \beta}$. Sachant que cela représente un transfert de ce qui est représenté par $[_{\alpha, \beta}$ dans la case $k-4$. Il doit être clair que, quel que soit le triplet $\alpha, \beta, (k-4)$, je sais construire l'algorithme T qui lui correspond, réciproquement pour chaque jeu de valeurs de ce triplet il faut bâtir un T particulier.

Dès lors il m'est possible de construire l'image de

$$[_{\alpha, \beta} := [_{\alpha, \beta} +_k 1$$

$$T_{(k-4)}^{\alpha, \beta}$$

$$c(k-4) := c(k-4) +_k 1,$$

$$c(k-4) := c(k-4) +_k 1,$$

$$c(k-4) := c(k-4) +_k 1,$$

$$c(k-4) := c(k-4) +_k 1,$$

$$c(k-4) := c(k-4) +_k 1,$$

```

c(k-4) := c(k-4) +k 1 ,
c(k-5) := cω ,
si c(k-5) = 0 vers e1 ,
c(k-4) := 0 ,
vers e2 ,
e1 : c(k-4) := c(k-4) -k 1 ,
c(k-4) := c(k-4) -k 1 ,
c(k-4) := c(k-4) -k 1 ,
c(k-4) := c(k-4) -k 1 ,
c(k-4) := c(k-4) -k 1 ,
c(k-4) := c(k-4) -k 1 ,
e2 : cω := c(k-5) .

```



Il me reste à construire un algorithme $R_{k-4}^{\alpha, \beta}$ qui envoie le contenu de (k-4) dans la case désignée par α, β . Pour cela il suffit de changer peu de choses à $\begin{matrix} k-3 \\ x \rightarrow (k-4) \end{matrix}$:

```

e'0 : c0 := c(k-4)
vers ef'
e'1 : c1 := c(k-4)
vers ef'
e'2 : c2 := c(k-4)
vers ef'
---      ---      ---
---      ---      ---
e'k-1 : c(k-1) := c(k-4)

```

A partir de là on constate que les instructions:

```

x := 0
x := y

```


si $y = 0$ vers e_i
vers e_j

obtiennent sans difficulté des images en terme de machine à cases adressables. L'exemple ci-après montre que les images de:

$$x := y \text{ A } y$$

et si $y \text{ R } y$ vers e_i

découlent de la même manière, il suffit de construire les algorithmes correspondants en termes de procédure formelle, et ça on sait le faire, et de les traduire ensuite en machine à cases adressables en utilisant les formes algorithmiques que nous venons de voir. Je n'insiste pas sur la forme $[_{\alpha}$ qui est plus simple à traduire que la forme $[_{\alpha, \beta}$.

Prenons un exemple pour illustrer la chose. Une instruction d'affectation quelconque fera l'affaire:

$$I_{2,3} := I_{5,10} +_k I_{4,0}$$

En terme de machine à cases adressables j'obtiens l'équivalent suivant:

$$T_{5,10}^{k-4} \quad , \quad T_{4,0}^{k-5} \quad , \quad c(k-6) := c(k-4) +_k c(k-5) \quad , \\ c(k-4) := c_w \quad , \quad R_{2,3}^{k-6} \quad , \quad c_w := c(k-4) \quad .$$

DISCUSSION.

Il me faut revenir sur les hypothèses afin de bien cerner le résultat obtenu.

- 1) Je pars d'une machine formelle MF dont la mémoire a K cases avec K contenu maximum par case.
- 2) Je me donne une machine à cases adressables MCA de $k=K+6$ cases, avec k contenu maximum de chaque case.
- 3) Je me donne un algorithme quelconque en terme de procédure

formelle, et je me demande si je vais pouvoir construire au moins un algorithme équivalent en terme de MCA, c'est-à-dire, tel que, partant du même jeu de données, et pour tout jeu possible de données, me donne le même jeu de résultats.

4) J'établis donc une correspondance entre les cases de la MF et certaines cases de la MCA: les cases 0 à K-1 de la MF sont mises en correspondance, chacune avec les cases 0 à k-7 de la MCA. Pour toute instruction de la MF qui agit sur certaines de ses cases, je construis un algorithme de MCA qui fait évoluer identiquement les cases de la MCA qui leur correspondent.

5) Pour les noms des cases, les algorithmes T et R règlent la question, pour les contenus il faut voir que $k = K + 6$ (Selon l'hypothèse exprimée en 2), or, seule l'addition fait évoluer les contenus, c'est donc dans l'image de $[_{\alpha, \beta} := [_{\alpha, \beta} + K \ 1$ qu'il faut prévoir le débordement à k-6, afin de respecter l'identité des calculs. Et c'est ce que fait la partie D de cet algorithme.

D'où l'énoncé du théorème:

Il n'existe pas d'algorithme de la PF, qui n'ait au moins un équivalent en termes de la machine de Nolin.

Il faut bien voir que cette propriété est une propriété de structure. Le fait de posséder une case spéciale telle que I ne permet pas de réaliser des calculs qui seraient impossibles dans la machine de Nolin. Par contre, dans cette dernière, il nous a fallu spécialiser six cases banales. Il est tout à fait possible de réduire ce nombre par divers moyens comme l'utilisation d'une file externe ou autre, mais impossible de l'annuler pour le type d'algorithme choisi ici, et il est connu par ailleurs que la puis-

sance d'une MCA est indépendante du nombre de ses cases.

On peut réfléchir sur les mécanismes de cette démonstration. Très tôt dans l'histoire de l'ordinateur, le calcul sur le nom de la variable a pris de l'importance. Pour manipuler un tableau de valeurs il était bien commode de disposer d'un nom commun à l'ensemble, et d'obtenir le nom de chaque élément par un calcul appliqué au nom commun. L'avantage est de n'avoir à transporter qu'une seule valeur: celle du nom commun. C'est un algorithme unique, valable pour tous les tableaux, qui permet d'atteindre l'élément.

Plus tardive mais de même nature est apparue la notion de translatabilité: un paquet d'information pouvait être implanté non pas en adresse effective, mais par rapport à une origine variable, flottante. Ce n'est qu'au moment de la mise en place en mémoire qu'on décide de la valeur de l'origine, qui devient le nom commun du paquet d'information.

D'un point de vue technique il faut souligner qu'en termes de machine de Nolin, le pivot du calcul sur le nom est l'instruction:

si ca = 0 vers ei

qui assure la reconnaissance de la valeur par sa position dans l'algorithme.

Du point de vue du principe, la construction de l'algorithme $T_{K-4}^{\alpha, \beta}$ montre réellement le rôle que joue un index, puisqu'on est bien là forcé de le simuler et avec des outils élémentaires.

Enfin du point de vue des idées fondamentales, si je me donne deux systèmes formels: la machine de Turing et une construction formelle proche des ordinateurs modernes, la procédure formelle, et que je me propose de définir ce qui est commun à ces systèmes disons le champ sémantique recouvert, il m'est indispensable de

me définir des systèmes intermédiaires de transition. Et c'est à ce moment que des notions surgissent d'elles-mêmes:

le concept de ruban borné aux deux bouts qui fait la mémoire centrale, et qui permet de créer la notion d'adresse, puis le concept d'index, ces deux concepts marquant les deux premières étapes.

LANGAGE D'ECHANGE
LANGAGE DE COMMUNICATION

p. sanchez

C.R. Subject classification informatics: D.3.1 D.3.2 D.3.3

Résumé

On part d'un langage de programmation comme le LAC. Afin de rendre ce dernier utilisable, il faut se donner deux autres sortes de langages: l'un va servir à manipuler des données, il fait le pendant du LAC qui lui, ne traite que de programme, l'autre, le langage de communication sert à manipuler des programmes tout construits et des jeux de données tout prêts. Les idées générales dégagées à cette occasion sont utilisables pour les autres langages de programmation.

LANGAGE DE TRAITEMENT

DE DONNEES

GENERALITES

Dans tout traitement dont l'aboutissement conduit à un résultat, il est nécessaire de fournir au préalable des données au programme, données qui pourront être dans le corps du programme en tant que variables locales ou, en tant que paramètres formels dont les valeurs effectives apparaîtront au cours des insertions de procédures.

Les programmes étant conçus pour travailler sur des données nous pouvons envisager le cas où la première procédure possède un nombre élevé de paramètres formels. Dans ces conditions, il serait préférable de posséder un langage adapté au maniement de ces objets. Ce langage pourra constituer des stocks de données dans lesquels les programmes puiseront des valeurs et entasseront des résultats, tout en assurant la communication avec l'extérieur. Le langage de traitement des données étant fonction du langage de programmation, nous aurons un langage de traitement des données pour chacun des langages de programmation.

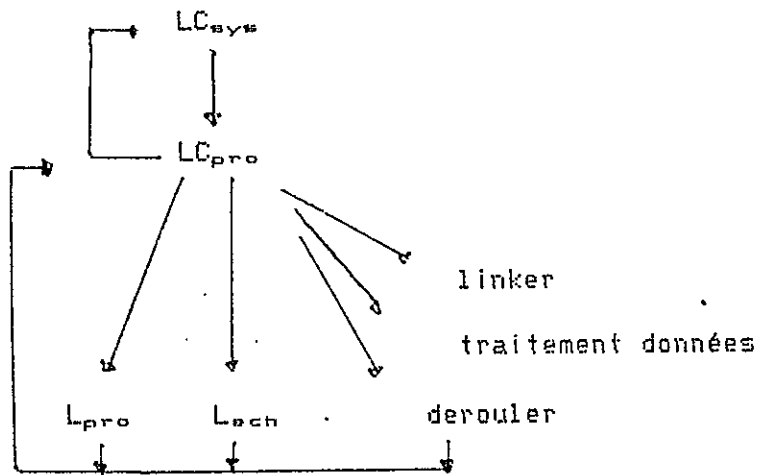
L'étude de ce langage se fera pour l'exemple particulier du langage de programmation LAC, développé dans les bulletins précédents.

Il ne faut pas oublier que dans le système que nous étudions, chacun des compilateurs qui le compose est "autojectif", donc la structure des données doit être finie et bornées. Chaque paramètre doit figurer par son nom, son type et sa valeur. Mais il faudra faire la différence entre les paramètres qui sont des données et ceux qui sont des résultats pour le programme. Dans l'instruction de création de la zone de données devra figurer la liste de tous les paramètres effectifs et le langage permettra certaines commodités de dialogue avec l'utilisateur. En effet, c'est à ce niveau de relation entre l'utilisateur et le système que l'on juge la commodité du langage de communication, il faudra donc lui faciliter la tâche en augmentant les performances de dialogue. Comme cela demande une étude énorme, nous aborderons ici quelques exemples quitte à la compléter par la suite au moment de la réalisation, la liste des commandes pouvant faire partie d'un manuel d'utilisation.

SYNOPTIQUE GENERAL DU SYSTEME

Lors de la mise en route d'un système, l'utilisateur se trouve confronté au langage de communication du système (LC_{sys}), qui est le langage le plus externe et qui dirigera l'utilisateur vers l'un des langages de programmation (L_{pro}) proposé. Mais cela se fera par l'intermédiaire d'un nouveau langage de communication (LC_{pro}), particulier au langage de programmation désiré, qui mettra à la disposition de l'utilisateur un langage de programmation (L_{pro}), un langage d'échange (L_{ech}), et diverses possibilités de construire ses programmes, dérouler avec plusieurs

options, gerer les procédures, les données, linker...



Nous nous occuperons plus particulièrement des commandes du langage de communication (`LCpro`) qui feront appel à un compilateur spécialisé dans le traitement des données.

CREATION DE LA CONFIGURATION DE DONNEES

créer <ident>, <n1> etage, <n2> ;

< ident > := nom de la configuration de données.

< n1 > := < entier > := nb. d'étages de la configuration.

< n2 > := < entier > := nb. d'éléments minimum de la configuration.

Exemple: creer CONF, 5 etage , 1000 ;

La zone de configuration de données ainsi définie est formée de n2 éléments minimum, un élément minimum étant par exemple l'octet, et de n1 paquets d'informations représentant chacun des données pour un programme que l'on appellera "etage". Chaque étage aura la même structure.

DEFINITION DE L'ETAGE

La longueur d'un étage étant définie lors de la création de la configuration, il faut maintenant décrire sa structure, c'est à dire donner les noms des paramètres et leurs types. Cela se fera par la phrase suivante:

etage < ident >, (type < ident >[(< n1 >)][,]) ;

nom de la configuration := < ident >

nom des paramètres := < ident >

n1 := < nombre > := dimension d'un tableau

type := réel entier booléen

Exemple: etage CONF, (reel A, ent B, bool Z, ent D(4), reel C(10)) ;

On pourra grâce aux étages, préparer à l'avance des configurations de données ayant la même structure, mais des valeurs de paramètres différentes, sur lesquelles un même programme pourra

s'appliquer.

ECRITURE DES PARAMETRES DANS UN ETAGE

Chaque étage pourra être complété séparément par une instruction d'écriture où ne figurera par exemple qu'un sous-ensemble des paramètres déclarés dans la construction de la configuration. En effet, c'est à ce niveau que se fera la distinction entre une donnée fournie au programme et un résultat. Une initialisation n'étant pas nécessaire si l'on est certain que le programme écrira une valeur dans ce paramètre. On laissera le choix à l'utilisateur d'écrire ou non la valeur du paramètre dans le texte de l'instruction, le système posera les questions utiles si cette valeur ne se trouve pas dans l'instruction, attendant la réponse pour faire plusieurs vérifications, par exemple, le type, la dimension...

ecrire < ident >, etage < n1 >, (liste param.) ;

nom de la configuration := < ident >

n1 := < nombre > := numero de l'étage

Pour décrire la liste des paramètres nous allons voir les possibilités du langage par l'écriture de quelques exemples.

Exemple 1: ecrire ESSAI, etage 2, (A=3.45, B=2, D(3)=7) ;

Dans ce cas, le système ne pose aucune question, toutes les valeurs étant données dans l'instruction d'écriture.

Exemple 2: ecrire ESSAI, etage 1, (A, B, D(2)=6) ;

Le système posera les questions suivantes:

A := ??

B := ??

car dans l'écriture de la phrase, seul les paramètres A et B n'étaient pas définis.

Exemple 3: ecrire ESSAI, etage 4 ;

Le système demandera les valeurs de tous les paramètres déclarés dans la construction de l'étage de la configuration.

Exemple 4: ecrire ESSAI, etage 2, (3.67, 45, 6, 1) ;

Le système fera la correspondance entre la valeur et le rang des paramètres. Cela pourrait être utile dans le cas de petites configurations, lorsqu'on est sûr qu'il n'y a pas d'erreur.

Exemple 5: ecrire ESSAI, etage 3, (D(I) : I=1 à 3, B, C=6) ;

Utile pour le remplissage des tableau.

D(1) := ??

D(2) := ??

D(3) := ??

B := ??

On peut imaginer aussi une remise à zéro d'un étage complet ou d'un tableau à l'aide d'une fonction, de la manière suivante:

ecrire ESSAI, etage 4, (ZERO) ;

ecrire ESSAI, etage 3, (A=5, D(ZERO), B) ;

LECTURE D'UN ETAGE

Nous allons maintenant définir par des exemples, les différentes manières de lire les étages d'une configuration, soit pour vérifier l'exactitude des valeurs, soit pour lire les résultats après l'exécution du programme.

lire < ident >, etage < nombre >, (liste de par. ou fct.) ;

nom de configuration := < ident > ;

numéro de l'étage := < nombre > ;

La syntaxe de la phrase sera étudiée par des exemples.

Exemple 6: lire ESSAI, etage 2 ;

A l'affichage on obtient toutes les valeurs de l'étage ainsi que leurs noms. On pourrait ajouter des fonctions pour changer la présentation des résultats, par exemple, avoir les valeurs en hexadécimal, en octal, en ASCII...

```
lire ESSAI, etage 2, (HEXA) ;
```

```
lire ESSAI, etage 4, (ASCII) ;
```

```
lire ESSAI, etage 3, (H,O,D(6), *B) ;
```

* := ASCII

O := OCTAL

H := HEXA

Exemple 7: lire ESSAI, etage 1, (A, B, C) ;

Donne parmi l'ensemble des paramètres, uniquement les valeurs de A, B, C.

Exemple 8: lire ESSAI, etage 3, (D(I) : I = 1 à 10) ;

Lorsqu'on utilise des tableaux, permet de lire tout ou partie du tableau.

ECHANGE DES CONFIGURATIONS

On voit ici que l'utilisation du langage de traitement des données est liée à l'utilisation d'un compilateur d'échanges qui prendra en compte les transferts entre la mémoire et les files externes. Le langage de communication auquel on fera appel pour utiliser le compilateur du langage de traitement de données pourra par la suite, et de manière immédiate ou différé, faire appel au compilateur d'échange. Le compilateur devra prendre en compte la sauvegarde et le chargement des différentes zones de données. Par exemple, une phrase du type:

save ESSAI ;

permet la sauvegarde de toute la zone, où

save ESSAI, etage 1 ;

permet la sauvegarde d'un étage.

charge ESSAI, etage 2 ;

permet au système de placer dans la zone de données l'étage 2 de la configuration ESSAI. L'application de la première procédure sur cette zone de données se fera par la phrase suivante:

applique P1, etage 2 de ESSAI, (A, B), 2000 ;

P1 doit posséder deux paramètres et la zone réservée au déroulement est constituée de 2000 élément minimum. Au système de placer dans les cases réservées aux adresses des paramètres les bonnes indications pour retrouver les valeurs des paramètres.

A ce moment du traitement le langage de communication peut appeler un compilateur de déroulement, qui grâce à des fonctions utilitaires, comme le déroulement avec trace, arrêt sur une valeur, sur une étiquette..., aidera à la mise au point finale du programme en surveillant son déroulement.

GLISSEMENT D'UN HYPERPAVE DANS UN CHAMP

MEMOIRE DE DIMENSION N

Jean-Philippe Lehmann

C.R. Subject classification informatics: B3

Résumé: On généralise le concept de fenêtre à la dimension N, et on propose une solution pour sa matérialisation.

DESCRIPTION

DOMAINE TECHNIQUE

Hardware.

5 INDICATION DE L'ETAT DE LA TECHNIQUE ANTERIEURE

Actuellement, dans les techniques de l'informatique, les structures de mémoires se présentent, en dernière analyse, comme des ensembles d'emplacements, adressables individuellement, tous disjoints les uns des autres. Il peut se faire que dans certaines applications, notamment graphiques, l'utilisateur ne soit pas conscient de ce cloisonnement, mais, au niveau interne, cette contrainte demeure: à l'ensemble des adresses correspond de façon injective, un ensemble d'emplacements qui sont complètement séparés les uns des autres. Dans le brevet 8412946, déposé le 16 août 1984
10 on expose une invention visant à supprimer cette contrainte; la solution exposée consiste en une nouvelle manière de structurer les blocs-mémoires et de les assembler entre eux, permettant alors un adressage par une fenêtre qui peut glisser dans l'espace mémoire; toutefois, dans ce brevet l'organisation de la mémoire
15 était linéaire, unidimensionnelle.
20

EXPOSE

1)Le problème.

Il s'agit de supprimer la contrainte décrite ci-dessus en créant une structure de mémoire qui permet d'associer à deux adresses
25 consécutives, deux emplacements qui en fait, vont se chevaucher au plus près, dans un sens qui est précisé ci-après, le tout dans le cas très général où, la structure mémoire sera n-dimensionnelle, ainsi que la fenêtre qui permet de l'accéder; de plus, on veut pouvoir masquer au choix la zone mémoire spécifiée par
30 l'adresse communiquée, et enfin, on veut qu'il soit possible d'assembler entre eux, les blocs mémoires qu'on aura définis, de manière que l'assemblage total continue d'avoir les propriétés obtenues au niveau des blocs.

2)La solution.

35 Dans le procédé qu'on va exposer, vont intervenir plusieurs éléments qui peuvent être réalisés de diverses façons dans diverses technologies; on donnera donc une description fonctionnelle de chacun de ces éléments, étant entendu, d'une part qu'ils réfèrent à toute technologie matérialisant les fonctions décrites, d'autre
40 part que ces fonctions décrites sont, soit déjà matérialisées dans des dispositifs existants, soit immédiatement matérialisables. On appelle élément minimum, tout dispositif pouvant jouer le rôle

d'une cellule mémoire élémentaire, dans laquelle il est possible de lire ou d'écrire, une quelconque parmi p configurations; par exemple, il peut s'agir d'un octet de bits ($p=2^8$). Pour de tels éléments, chacune des configurations forme un tout indissociable, transmis vers ou reçu depuis, l'extérieur, sans qu'il y ait à décomposer, durant le transfert, l'information en éléments constitutifs plus simples. Chaque élément minimum est relié à l'extérieur par un canal d'entrée et un canal de sortie, grâce auxquels transitent les données que constituent les configurations. Dans ce qui suit, on supposera que ces deux canaux sont confondus, en un seul bidirectionnel. A chaque élément minimum peut s'appliquer une commande dont la fonction est de l'activer ou le désactiver, c'est à dire, soit de permettre une opération de lecture ou d'écriture, soit d'interdire de telles opérations. Une autre commande éventuelle peut s'appliquer, spécifiant le sens du transfert. Soient alors $N_0.N_1.....N_{n-1}$ tels éléments. Soient $k_0.k_1....k_{n-1}$ canaux, pouvant permettre chacun le transit de p configurations distinctes, (pour $0 \leq i \leq n-1$ on aura $1 \leq k_i \leq N_i$). L'ensemble de ces canaux est noté D, et chaque canal sera noté $D^{i_0, i_1, \dots, i_{n-1}}$. Dans cette notation pour $0 \leq j \leq n-1$ on aura $0 \leq i_j \leq k_j - 1$. Un élément minimum et sa commande d'activation seront notés respectivement $M^{i_0, i_1, \dots, i_{n-1}}$ et $CS^{i_0, i_1, \dots, i_{n-1}}$. Dans cette notation, pour $0 \leq j \leq n-1$ on a $0 \leq i_j \leq N_{j-1}$.

25 Une adresse qu'on veut communiquer au système qu'on veut structurer est constituée par la donnée d'un n-uple $(i_0, i_1, \dots, i_{n-1})$; dans cette notation, pour $0 \leq j \leq n-1$ on a $0 \leq i_j \leq N_{j-1}$. Une telle information permettra en effet de localiser dans la mémoire, qu'on peut se représenter comme un hyperparallélépipède, l'élément minimum dont les coordonnées sont celles du n-uple ci-dessus.

30 Pour atteindre le but annoncé, il faudra alors que soit sélectionné dans la mémoire l'hyperpavé représenté par les $k_0.k_1....k_{n-1}$ éléments minimum dont les coordonnées sont les éléments du produit cartésien $\{i_0, i_0+1, \dots, i_0+k_0-1\} \times \{i_1, i_1+1, \dots, i_1+k_1-1\} \times \dots$

35 $\dots \times \{i_{n-1}, i_{n-1}+1, \dots, i_{n-1}+k_{n-1}-1\}$ et ceci de façon que l'information relative à l'élément $M^{i_0+u_0, i_1+u_1, \dots, i_{n-1}+u_{n-1}}$ circule sur le canal $D^{u_0, u_1, \dots, u_{n-1}}$ et ceci pour tout u_j variant de 0 à k_j-1 , bornes comprises ($0 \leq j \leq n-1$). Si un tel résultat est atteint l'opération de glissement continu avec chevauchement sera réalisée.

On note cependant, qu'un problème se posera lorsque, dans le n-uple d'adresses on trouvera des éléments i_j qui vérifient $i_j + k_j - 1 > N_j - 1$; il y a dans ce cas, débordement, et se pose la question de l'organisation de la structure pour permettre la jonction d'un bloc à un autre. Ce point est examiné plus loin.

D'autre part, en même temps que l'adresse, est communiquée au système une condition de masquage relativement à l'hyperpavé sélectionné. Celle-ci est déterminée par la donnée de n conditions du type $(H_0^i, H_1^i, \dots, H_{k_i-1}^i)$ i variant de 0 à n-1 et où chaque H_j^i vaut 0 ou 1. L'élément minimum sélectionné au niveau de l'adressage ne devra être actif que si, étant noté comme on l'a vu plus haut, $M^{i_0+u_0, \dots, i_{n-1}+u_{n-1}}$, alors $H_{u_0}^0, H_{u_1}^1, \dots, H_{u_{n-1}}^{n-1}$ sont tous à 1.

On peut maintenant indiquer comment procéder:

a) en premier lieu, pour permettre structurellement l'opération de glissement, on met en communication le canal attaché à un élément minimum, avec chacun des canaux de l'ensemble D, et on fait ceci pour chaque élément minimum; cette mise en communication signifie fonctionnellement qu'on dispose du moyen d'associer de façon bijective à chacune des p configurations circulant sur le canal d'un élément minimum, une des p configurations pouvant circuler sur chaque canal de D. Dans le cas le plus simple cette bijection sera l'identité. On définit ainsi, pour chaque élément minimum $k_0 \cdot k_1 \dots k_{n-1}$ liaisons; ces liaisons sont chacune sous le contrôle d'une commande notée $C_{u_0, u_1, \dots, u_{n-1}}^{i_0, i_1, \dots, i_{n-1}}$ éventuellement associée à une commande du type R/\bar{W} , le tout ayant pour fonction d'activer ou non la liaison en autorisant le transfert (dans un sens ou l'autre) ou en l'interdisant.

b) La structure n-dimensionnelle de la mémoire qu'on est en train de concevoir, va permettre de définir les conditions d'activation de chaque commande $C_{u_0, u_1, \dots, u_{n-1}}^{i_0, i_1, \dots, i_{n-1}}$ à partir de la réalisation conjointe de conditions nécessaires au niveau de chaque coordonnée de l'espace produit; si on examine pour une coordonnée i_j du n-uple d'adresse les conditions nécessaires à satisfaire, on obtient le tableau I, dans lequel on ne fait pas figurer les indices relatifs à la coordonnée particulière choisie, pour ne pas alourdir l'écriture; les indices + réfèrent au prolongement de l'espace mémoire, le long de la coordonnée considérée. De façon générale c'est ainsi qu'on utilisera en indice le signe +; de même le signe -, en indice, réfèrera aux commandes ou adresses du bloc antérieur à celui qu'on considère.

Tableau I

	H_0	H_1	H_{k-2}	H_{k-1}
0	CS^0	CS^1	CS^{k-2}	CS^{k-1}
1	CS^1	CS^2	CS^{k-1}	CS^k
5
.....
k-1	CS^{k-1}	CS^k	CS^{2k-3}	CS^{2k-2}
k	CS^k	CS^{k+1}	CS^{2k-2}	CS^{2k-1}
10
.....
N-k	CS^{N-k}	CS^{N-k+1}	CS^{N-2}	CS^{N-1}
N-k+1	CS^{N-k+1}	CS^{N-k+2}	CS^{N-1}	CS^{0+}
15
.....
N-1	CS^{N-1}	CS^{0+}	CS^{k-3+}	CS^{k-2+}

Pour distinguer les signes +, lorsqu'ils sont utilisés comme des indices, de leur utilisation courante, il suffit de noter que dans le premier cas, ils ne sont suivis par aucun autre symbole; on fera de même lorsqu'on utilisera le signe -, dans le sens qui a été indiqué plus haut. Le tableau II, en même temps qu'il indique comment les commandes CS d'un bloc suivant seront affectées, permet de déduire comment les coordonnées adresses du bloc précédent influenceront les commandes CS du bloc qu'on est en train de structurer. Au total l'examen du tableau TI permet d'exprimer la condition nécessaire d'activation d'une commande du type CS ' , , i_j , , en fonction de la coordonnée adresse et de la condition de masquage spécifiée pour cette coordonnée. La condition nécessaire et suffisante d'activation de la commande du type CS^{i₀, i₁, ..., i_{n-1}} étant obtenue par la conjonction de la réalisation des conditions du type CI, relatives à chaque i_j, j variant de 0 à n-1, bornes comprises: (Adr désigne la valeur de l'adresse)

35 (CI)	$1 \leq v \leq k_j$ $k_j - 1 \leq i_j \leq N_j - 1$	$Adr = i_j - k_j + v$ ET $H_{k_j - v}^j = 1$
	$0 \leq v \leq i_j$ $0 \leq i_j \leq k_j - 2$	$Adr = v$ ET $H_{i_j - v}^j = 1$
	$0 \leq i_j \leq k_j - 2$ $1 \leq v \leq k_j - i_j - 1$	$Adr = (N_j - k_j + i_j + v)^-$ ET $H_{k_j - v}^j = 1$

c) On procède de la même façon en ce qui concerne les commandes $C_{i_0, i_1, \dots, i_{n-1}}$ et $C_{u_0, u_1, \dots, u_{n-1}}$. On examine là encore au niveau d'une coordonnée, les conditions nécessaires à satisfaire; la tableau TII, ci-après, présente pour chaque valeur de l'adresse (relativement à la coordonnée considérée), en regard, la liste des commandes du type précisé ci-dessus, qu'il faut activer. Les indices relatifs aux autres coordonnées ne figurent pas.

Tableau TII

	0	C_0^0	C_1^1	C_{k-2}^{k-2}	C_{k-1}^{k-1}
10	1	C_0^1	C_1^2	C_{k-2}^{k-1}	C_{k-1}^k
	2	C_0^2	C_1^3	C_{k-2}^k	C_{k-1}^{k+1}

	N-k	C_0^{N-k}	C_1^{N-k+1}	C_{k-2}^{N-2}	C_{k-1}^{N-1}
15	N-k+1	C_0^{N-k+1}	C_1^{N-k+2}	C_{k-2}^{N-1}	C_{k-1}^{0+}
	N-k+2	C_0^{N-k+2}	C_1^{N-k+3}	C_{k-2}^{0+}	C_{k-1}^{1+}

	N-1	C_0^{N-1}	C_1^{0+}	C_{k-2}^{N-3+}	C_{k-1}^{N-2+}

Il est alors possible, à partir de TII, d'exprimer la condition nécessaire d'activation d'une commande du type $C_{i_0, i_1, \dots, i_j, \dots, u_j, \dots, u_{n-1}}$ en fonction de la coordonnée adresse. La condition nécessaire et suffisante d'activation de la commande du type $C_{i_0, i_1, \dots, i_{n-1}}$ et $C_{u_0, u_1, \dots, u_{n-1}}$ étant obtenue par la conjonction de la réalisation des conditions du type CII, relatives à chaque couple (i_j, u_j) j variant de 0 à n-1, bornes comprises: (Adr désigne la valeur de l'adresse, et le signe - est utilisé en indice comme précédemment)

(CII)	$0 \leq u_j \leq k_j - 1 \leq i_j \leq N_j - 1$	Adr = $i_j - u_j$
	$0 \leq i_j < u_j \leq k_j - 1$	Adr = $(N_j - i_j + u_j)^-$

d) Il reste à définir les conditions à satisfaire pour permettre l'assemblage de blocs structurés selon les conditions précédentes, entre eux. On se place dans l'espace d'une coordonnée, et on examine ce qui se passe, lorsque trois blocs se suivent B^- , B et B^+ ; on a vu que les adresses finales de B^- doivent influencer sur les commandes d'activation dans B, et de même les adresses finales de

B sur les commandes de B^+ ; il faut donc se donner les moyens de communiquer les informations constituées par les k_i-1 adresses de B^- au bloc B, c'est à dire prévoir dans B, un canal d'entrée qui puisse accueillir de telles informations; de la même façon, pour
5 communiquer les informations constituées par les k_i-1 dernières adresses de B à B^+ , on peut prévoir un canal de sortie du bloc B à cet effet. Ceci est exprimé par les conditions CIII, ci-après, qui devront être satisfaites, pour tout i , i variant de 0 à $n-1$.

10

(CIII)

Tout bloc possède au niveau de la coordonnée i , un canal d'entrée pouvant accepter k_i-1 informations distinctes, associées aux dernières adresses du bloc précédent, éventuellement un canal de sortie, pour ses dernières adresses à transmettre au bloc suivant.

15

20

25

30

35

40

e) Au total et en résumé, le procédé général permettant de structurer l'espace mémoire conformément au but poursuivi, peut se décrire ainsi: - au niveau des entrées, un ensemble de canaux pouvant recevoir les coordonnées d'adresses internes au bloc, un ensemble de canaux pouvant recevoir les coordonnées d'adresses selon CIII du bloc précédent, un ensemble de canaux pouvant recevoir la condition de masquage, une éventuelle entrée R/\bar{W} , enfin l'ensemble des canaux réservés au transit des données.

- au niveau des sorties, outre les canaux données, un ensemble de canaux*, communiquant les adresses finales d'une ou plusieurs coordonnées au bloc suivant selon CIII. (*:éventuel)

- au niveau interne, mise en communication de chaque canal donnée d'un élément minimum, avec tous les canaux de D; activation des commandes $CS^{i_0, i_1, \dots, i_{n-1}}$ en fonction de l'adresse et du masque, selon les conditions définies en CI; activation des commandes $C^{i_0, i_1, \dots, i_{n-1}}$ en fonction des adresses communi-

quées au système, selon les conditions définies en CII.

(L'éventualité notée en CIII tient à ce que ces informations peuvent parvenir à un bloc sans provenir du précédent.)

3) Avantages de la methode

Les programmeurs qui travaillent au niveau machine sont confrontés en permanence aux difficultés entraînées par l'absence de souplesse que détermine la structure cloisonnée des mémoires actuelles; quand ces mêmes difficultés leur sont, en partie, épargnées, c'est qu'elles ont été réglées au niveau de la programmation interne au processeur, par exemple par le biais de la microprogrammation; il n'en reste pas moins qu'au niveau de la structure de la mémoire,

le clâisonnement demeure, en dernière analyse, une caractéristique essentielle; certes diverses techniques, à la fois hard et soft, sont mises en oeuvre, pour permettre, par le jeu d'une seule instruction machine, l'accès direct à une donnée éventuellement complexe; mais la question essentielle, consiste à savoir combien de tops d'horloge interne, au niveau le plus fin, sont nécessaires pour la réalisation de telles instructions, et cela, même si les opérations déclenchées, sont invisibles au niveau du programmeur-machine, car elles n'en ont pas moins une existence; il ne serait pas judicieux non plus, de comparer les avantages de l'invention décrite à d'autres procédés mettant en oeuvre des techniques de parallelisme plus élaborées ou de pipe-line: en effet, dans la description qu'on a faite, ces techniques n'ont pas été mises en oeuvre, alors que la méthode peut parfaitement les supporter, et de ce fait, être elle-même, considérablement améliorée; dans ces conditions une comparaison pertinente ne semble pouvoir être menée qu'en se posant le genre de questions suivant: une adresse est délivrée à la mémoire, quelle est l'emplacement résultant immédiatement spécifié, sans actions annexes supplémentaires, sans autres tops d'horloge, que ce soit au niveau du processeur central, ou au niveau de dispositifs secondaires voués à sous-traiter l'information constituée par l'adresse? Pour illustrer ces propos, on peut par exemple, considérer un processeur tel que le NS32032: on constate alors que le temps d'accès à un mot de 16 bits ou à un mot de 32 bits est fonction de son adresse, ce qui signifie, qu'en fait, lorsque l'information accédée chevauche deux emplacements-mémoire, deux accès sont nécessaires au lieu d'un seul, comme notre procédé le permet. La structure qu'on propose est en fait assimilable à un espace auquel il est possible d'atteindre par le moyen d'une fenêtre, qui peut être instantanément située, n'importe où, dans cet espace; il ne s'agit plus d'adresser, ni des objets, ni même des cases ou des ensembles de cases, mais de se localiser dans un espace sans aucune autre restriction que celle des dimensions de la fenêtre par laquelle l'espace peut être observé. A partie d'une telle faculté, il devient possible d'envisager une gestion de suites d'informations de longueur variable à la fois optimisée et assouplie par rapport à l'état actuel de la technique; non seulement le nombre d'accès à la mémoire est diminué, mais encore la taille nécessaire de celle-ci se trouve réduite. On peut également indiquer que l'exploitation des ressources offertes par une telle structure est déjà possible avec les processeurs exis-

-tants, mais que la voie est également ouverte à la conception de processeurs et de langages-machine tirant le parti maximum de l'invention proposée, en étendant les avantages précédemment définis. Le fait de disposer d'un masque, permet en outre de définir, à chaque accès mémoire, les voies-données qui seront actives à cette phase; en conséquence les autres laissées libres, peuvent être affectées éventuellement à un autre usage, ainsi que les registres internes du processeur associés aux voies masquées.

La généralisation du procédé à un espace à n dimensions, offre, entre autres des domaines d'application aux calculs en parallèle, aux calculs graphiques, aux techniques de reconnaissance de formes: ainsi le repérage d'une configuration de bits donnée, de dimensions inférieures à celle de la fenêtre, constitue un problème aisément résolu, avec la structure de mémoire proposée, avec un gain de temps considérable. Les avantages décrits ne constituent pas une liste exhaustive.

DESCRIPTION DES DESSINS

Les dessins 1,2,3,4,5 permettent de se représenter l'assemblage d'un bloc, relativement au mode de réalisation n°6, et dans le cas où $p = 1$, $n = 2$, $N_0 = 7$, $N_1 = 5$, $k_0 = 4$, $k_1 = 3$. On est donc dans le cas de la dimension 2; deux décodeurs sont utilisés D^K et D^L , qui fournissent 7 sorties pour D^L , de L_0 à L_6 , et 5 sorties pour D^K , de K_0 à K_4 . Les notations D^L et D^K font référence à un dispositif ligne et un dispositif colonne. La fenêtre est, en ligne de taille 4, et en colonne de taille 3; il y a donc trois entrées en ligne (4-1), E_4^L , E_5^L , E_6^L , et deux entrées en colonne (3-1), E_4^K , E_3^K , ces entrées étant prévues pour le chevauchement de l'information sélectionnée, sur le bloc décrit et son précédent. Pour le chevauchement du bloc et de son suivant, sont prévues les sorties lignes L_4 , L_5 , L_6 , et colonnes K_3 et K_4 ; tout ceci figure dans le dessin 5. Chaque dessin comporte une ou plusieurs zones rectangulaires, dont le câblage n'est pas figuré dans le dit dessin, mais apparaît dans le dessin de numéro inférieur; pour que le tout soit intelligible, il faut donc considérer successivement les dessins dans l'ordre de leur numérotation. Dans le dessin 4, sont câblées les commandes d'activation, lignes et colonnes, relatives aux CS, ainsi que celles relatives aux C. Dans le dessin 3 est représenté l'ensemble des 35 éléments minimum et dans le dessin 2, le détail de l'un d'entre eux, avec sa connection au bus à 12 canaux binaires. On n'a pas figuré dans ces dessins le câblage de la condition de masquage. Le dessin 1 représente le bloc R, dont 12 exemplaires

interviennent dans le dessin 2.

EXPOSES DE MODES DE REALISATION

Les conditions décrites dans l'exposé général et résumées dans le
 paragraphe 2)e), définissent la solution générale du problème po-
 5 sé, en indiquant les liaisons à réaliser, et comment doivent être
 activées les commandes mises en jeu, à partir des données fournies
 au système, adresse et masquage. Ces liaisons et ces fonctions
 font maintenant l'objet de modes de réalisation particuliers, sans
 10 que ceux qu'on expose ci-après soient limitatifs; il s'agit seule-
 ment de matérialiser dans des dispositifs adéquats ces liaisons
 et ces fonctions, ou plus précisément, de montrer comment ces li-
 aisons et fonctions peuvent être exprimées dans un cadre logique
 booléen, à partir duquel, leur matérialisation dans des dispositifs
 15 classiques et connus, devient évidente. Les technologies pouvant
 intervenir dans ces réalisations, peuvent être différentes, MOS,
 CMOS, TTL, etc..., du moment que les matérialisations qu'elles per-
 mettent, correspondent à la description fonctionnelle des organes
 et aux liaisons qu'il faut installer. Par exemple les canaux qui
 interviennent dans les modes de réalisation, peuvent être des fils
 20 électriques, mais on pourrait aussi bien envisager des fibres op-
 tiques; ces canaux sont à deux états physiques, mais ils pourraient
 en avoir davantage; les organes autorisant ou non les flux d'infor-
 mations, peuvent être des buffers 3-états, l'essentiel étant de
 réaliser une fonction de disconnection en un point d'un canal.
 25 Dans ce qui suit on s'exprimera dans le cadre de la logique positi-
 ve.

1) Mode de réalisation n°1.

Le n-uple des adresses est communiqué au système par un ensemble
 de n bus, chacun possédant m_i canaux binaires, ($2^{m_i} \gg N_i$). A cha-
 30 cun de ces bus, est associé un dispositif de décodage classique,
 ayant en sortie 2^{m_i} canaux binaires, l'activation de chacun de ces
 canaux correspondant à chacune des adresses allant de 0 à N_i ; les
 canaux supplémentaires sont inutilisés; ces sorties sont notées
 $S_0^i, S_1^i, \dots, S_{N_i-1}^i$. Pour permettre au système de recevoir l'in-
 35 formation relative aux adresses finales du bloc précédent, on le
 munit des $k_0-1+k_1-1+\dots+k_{n-1}-1$ entrées ainsi notées, $E_{N_i-1}^i, E_{N_i-2}^i,$
 $\dots, E_{N_i-k_i+1}^i$ pour i variant de 0 à n-1; de même on prévoit en
 sortie les suivantes: $S_{N_i-1}^i, S_{N_i-2}^i, \dots, S_{N_i-k_i+1}^i$, i variant
 comme ci-dessus. Chaque élément minimum possède une commande notée

CS^{i₀,i₁,...,i_{n-1}} et éventuellement une commande R/ \bar{W} , permettant d'activer cet élément et de le positionner en émission ou en réception. La condition de masquage est transmise au système par un ensemble de n bus, chacun possédant k_i canaux binaires; chacun de ces bus est noté H₀ⁱ, H₁ⁱ, ..., H_{k_i-1}ⁱ. Il existe un bus bidirectionnel à k₀.k₁.....k_{n-1} canaux binaires, et chaque canal données, attaché à un élément minimum, est simplement relié à tous les canaux de D; chaque liaison ainsi établie est mise sous le contrôle d'une commande C^{i₀,i₁,...,i_{n-1}} éventuellement associée à une commande R/ \bar{W} , pour autoriser le flux d'informations ou non, dans un sens ou l'autre.

Les équations logiques relatives à l'activation des CS^{i₀,i₁,...,i_{n-1}} sont alors immédiatement obtenues, par l'examen des conditions CI données dans la description générale, et qui prennent ici l'aspect suivant:

	$CS^{i_0, i_1, \dots, i_{n-1}} = CS^{i_0} \cdot CS^{i_1} \dots CS^{i_{n-1}}$
CI ¹	$CS^{i_j} = \sum_{v=1}^{k_j} S_{i_j - k_j + v}^j \cdot H_{k_j - v}^j$ $k_j - 1 \leq i_j \leq N_j - 1$ $0 \leq j \leq n - 1$
20	$CS^{i_j} = \sum_{v=1}^{k_j - i_j - 1} E_{N_j - k_j + i_j + v}^j \cdot H_{k_j - v}^j \oplus \sum_{v=0}^{i_j} S_v^j \cdot H_{i_j - v}^j$ $0 \leq i_j < k_j - 1$ $0 \leq j \leq n - 1$

Les signes \sum et \oplus sont associés au OU logique inclusif, le signe \cdot est associé au ET logique. Il est immédiat de concevoir à partir de ces équations logiques, toutes sortes de cablages logiques qui les réalisent, par exemple en substituant aux signes précisés ci-dessus les portes électroniques correspondantes. Les équations logiques relatives aux commandes C^{i₀,i₁,...,i_{n-1}} sont aussi obtenues à partir des conditions CII de la description générale, et prennent l'aspect suivant: (là encore, la conception, à partir de telles équations, de dispositifs de cablages les réalisant, se résout de façon évidente, comme on l'a indiqué ci-dessus)

	$C_{u_0, u_1, \dots, u_{n-1}}^{i_0, i_1, \dots, i_{n-1}} = C_{u_0}^{i_0} \cdot C_{u_1}^{i_1} \cdot \dots \cdot C_{u_{n-1}}^{i_{n-1}}$
	$C_{u_j}^{i_j} = S_{i_j - u_j}^j \quad 0 \leq u_j \leq i_j \leq k_j - 1$
CII ¹	$C_{u_j}^{i_j} = E_{N_j - i_j + u_j}^j \quad 0 \leq i_j < u_j \leq k_j - 1$
	Pour j variant de 0 à n-1, bornes comprises.

5 Pour assembler les blocs entre eux de façon que le chevauchement soit possible de l'un à l'autre, les conditions CIII nous donnent la manière de procéder: il faut pouvoir communiquer, pour chaque coordonnée, les $k_j - 1$ adresses dernières d'un bloc à son suivant, et de même au niveau des sorties. Dans le cas présent, assembler à un

10 hyperparallélépipède un autre, cela signifie que le dernier est obtenu à partir du premier par une translation de longueur k_j , le long de la i_j ème coordonnée; de ce fait, toutes les équations logiques établies pour les autres coordonnées, n'ont pas à être modifiées; pour la coordonnée concernée, il suffira de relier les sorties

15 $S_{N_j - v_j}^j$ aux entrées respectives du bloc suivant $E_{N_j - v_j}^j$ et ceci dans le domaine $1 \leq v_j \leq N_j - k_j + 1$. Il devient ainsi possible d'empiler les hyperparallélépipèdes, sans aucune autre limitation que celle de leur nombre.

2) Mode de réalisation n°2.

20 On se place dans le cas précédent sans que la possibilité du masquage soit offerte. Les notations utilisées sont les mêmes, et la solution consiste à:

- pour chaque élément minimum, relier son canal données à tous les canaux de D, comme il a été exposé dans le mode de réalisation
- 25 n°1.

- réaliser des dispositifs de câblage satisfaisant aux conditions

	$CS_{i_0, i_1, \dots, i_{n-1}} = CS_{i_0} \cdot CS_{i_1} \cdot \dots \cdot CS_{i_{n-1}}$
CI ²	$CS_{i_j}^j = \sum_{v=1}^{k_j} S_{i_j - k_j + v}^j \quad \text{avec } k_j - 1 \leq i_j \leq N_j - 1$ $\text{et } 0 \leq j \leq n-1$
30	$CS_{i_j}^j = \sum_{v=1}^{k_j - i_j - 1} E_{N_j - k_j + i_j + v}^j \oplus \sum_{v=0}^{i_j} S_v^j$ $\text{avec } 0 \leq i_j < k_j - 1 \quad \text{et } 0 \leq j \leq n-1$

- réaliser des dispositifs de câblage satisfaisant aux conditions suivantes: celles décrites dans le tableau CII¹.

-relier les sorties $S_{N_j - v_j}^j$ aux entrées du bloc suivant $E_{N_j - v_j}^j$ dans le domaine $1 \leq v_j \leq N_j - k_j + 1$.

3) Mode de réalisation n°3.

On se place dans le cas du mode de réalisation n°1, lorsque $n=1$.

5 - chaque élément minimum a son canal données relié à tous les canaux de D, qui sont au nombre de k.

- les commandes CS^i de chaque élément minimum sont ainsi câblées:

$$\begin{array}{l}
 \text{CI}^3 \\
 \hline
 CS^i = \sum_{v=1}^k S_{i-k+v} \cdot H_{k-v} \quad \text{avec } k-1 \leq i \leq N-1 \\
 \hline
 CS^i = \sum_{v=1}^{k-i-1} E_{N-k+i+v} \cdot H_{k-v} \oplus \sum_{v=0}^i S_v \cdot H_{i-v} \\
 \hline
 \text{avec } 0 \leq i < k-1
 \end{array}$$

10

- les commandes C_u^i autorisant les liaisons au bus D, sont alors:

$$\begin{array}{l}
 \text{CII}^3 \\
 \hline
 C_u^i = S_{i-u} \quad \text{pour } 0 \leq u \leq i \leq k-1 \\
 \hline
 C_u^i = E_{N-i+u} \quad \text{pour } 0 \leq i < u \leq k-1
 \end{array}$$

15 - les sorties S_{N-v} sont reliées aux entrées du bloc suivant E_{N-v} , v variant de 1 à N-k+1, bornes comprises.

4) Mode de réalisation n°4.

On se place dans le cas précédent, mais sans condition de masquage.

- chaque élément minimum a son canal données relié à tous les canaux de D.

20 - les commandes CS^i de chaque élément minimum sont ainsi câblées:

$$\begin{array}{l}
 \text{CI}^4 \\
 \hline
 CS^i = \sum_{v=1}^k S_{i-k+v} \quad \text{avec } k-1 \leq i \leq N-1 \\
 \hline
 CS^i = \sum_{v=1}^{k-i-1} E_{N-k+i+v} \oplus \sum_{v=0}^i S_v \quad \text{avec } 0 \leq i < k-1
 \end{array}$$

-les conditions relatives aux commandes C_u^i sont celles décrites en CII³.

25 - les sorties S_{N-v} sont reliées aux entrées E_{N-v} du bloc suivant. On note qu'on retrouve ainsi la solution décrite dans le brevet n° 8412946 qui a été déjà signalé dans le début de la description.

5) Mode de réalisation n°5.

On se place dans le cas du mode de réalisation n°1, lorsque $n=2$.

30 La mémoire peut alors être conçue comme un rectangle, ainsi que la fenêtre d'accèsion, comme un rectangle de dimensions inférieures à celles du précédent. Les dessins qui figurent dans ce brevet illustrent cette situation, sans toutefois faire intervenir la condition de masquage. On a introduit quelques différences de

notation, qu'on reprend ici, pour que la compréhension des dessins soit plus aisée: la notation utilisée fait intervenir en indices les lettres K et L, pour référencer les colonnes et les lignes; on a ainsi à la place de $CS^{i_0, i_1} = CS^{i_0} \cdot CS^{i_1}$, l'écriture suivante:

5 $CS_{ij} = CS_i^L \cdot CS_j^K$. De même k_0, k_1, N_0, N_1 deviennent respectivement k^L, k^K, N^L, N^K ; les sorties du décodeur ligne sont notées L_i et celles du décodeur colonne K_j ; les commandes qui étaient notées $C_{u_0, u_1}^{i_0, i_1}$ sont notées C_{uv}^{ij} . La solution peut alors être décrite:

10 - chaque élément minimum a son canal données relié à tous les canaux de D, qui sont au nombre de $k^L \cdot k^K$.

- les commandes CS_{ij} sont ainsi câblées:

$CS_{ij} = CS_i^L \cdot CS_j^K$ pour $0 \leq i \leq N^L - 1$ et $0 \leq j \leq N^K - 1$.	
$CS_i^L = \sum_{v=1}^{k^L} L_{i-k^L+v} \cdot H_{k^L-v}^L$ avec $k^L - 1 \leq i \leq N^L - 1$	
$CS_i^L = \sum_{v=1}^{k^L-i-1} E_{N^L-k^L+i+v}^L \cdot H_{k^L-v}^L \oplus \sum_{v=0}^i L_v \cdot H_{i-v}^L$ avec $0 \leq i < k^L - 1$	CI ⁵
$CS_j^K = \sum_{v=1}^{k^K} K_{j-k^K+v} \cdot H_{k^K-v}^K$ avec $k^K - 1 \leq j \leq N^K - 1$	
$CS_j^K = \sum_{v=1}^{k^K-j-1} E_{N^K-k^K+j+v}^K \cdot H_{k^K-v}^K \oplus \sum_{v=0}^j K_v \cdot H_{j-v}^K$ avec $0 \leq j < k^K - 1$	

15

- les commandes C_{uv}^{ij} sont ainsi câblées:

$C_{uv}^{ij} = Y_u^i \cdot X_v^j$
$Y_u^i = L_{i-u}$ pour $0 \leq u \leq i \leq k^L - 1$
$Y_u^i = E_{N^L-i+u}^L$ pour $0 \leq i < u \leq k^L - 1$
$X_v^j = K_{j-v}$ pour $0 \leq v \leq j \leq k^K - 1$
$X_v^j = E_{N^K-j+v}^K$ pour $0 \leq j < v \leq k^K - 1$

20

CII⁵

25

- les sorties L_i relatives aux $k^L - 1$ dernières adresses-lignes d'un bloc sont reliées aux entrées E_j^L correspondantes, et de même pour les sorties K_j relatives aux $k^K - 1$ dernières adresses colonnes qu'on relie aux entrées E_j^K correspondantes, comme il a été indiqué dans les modes précédents.

6) Mode de réalisation n°6.

On reprend le cas précédent sans que la possibilité du masquage soit offerte; la solution est alors obtenue:

- pour les liaisons internes et externes de la même façon;
- 5 - les équations CI⁶ et CII⁶, sont les mêmes que les équations CI⁵ et CII⁵, dans lesquelles on supprime les termes en H. les dessins illustrent particulièrement cette situation.

7) Mode de réalisation n°7.

On se place dans le cas du mode de réalisation n°1, lorsque n=3.

- 10 On peut alors se représenter la mémoire comme un volume, plus précisément comme un parallélépipède rectangle, le fenêtré d'accession étant également un parallélépipède rectangle et de dimensions inférieures à celles du précédent. Ce mode permet, par exemple d'accéder à des plans à l'intérieur d'un volume, soit en fixant une des
- 15 valeurs k_1 à 1, soit en agissant au niveau du masquage. La solution est ainsi obtenue:(on revient aux notations du début)

- chaque élément minimum a son canal données relié à tous les canaux de D, qui sont au nombre de $k_0.k_1.k_2$.
- les commandes d'activation des éléments minimum sont alors:

20
$$CS^{i_0, i_1, i_2} = CS^{i_0} \cdot CS^{i_1} \cdot CS^{i_2}$$

les équations définies en CI¹ étant reprises et spécifiées en remplaçant n par la valeur 2. CI⁷

- pour les autres commandes on a:

25
$$C_{u_0, u_1, u_2}^{i_0, i_1, i_2} = C_{u_0}^{i_0} \cdot C_{u_1}^{i_1} \cdot C_{u_2}^{i_2}$$

les équations définies en CII¹ étant reprises et spécifiées en remplaçant n par la valeur 2. CII⁷

- les sorties $S_{N_0-v_0}^0, S_{N_1-v_1}^1, S_{N_2-v_2}^2$, sont respectivement reliées aux entrées $E_{N_0-v_0}^0, E_{N_1-v_1}^1, E_{N_2-v_2}^2$ du bloc suivant, lorsqu'on a $1 \leq v_0 \leq N_0 - k_0 + 1, 1 \leq v_1 \leq N_1 - k_1 + 1, 1 \leq v_2 \leq N_2 - k_2 + 1$.

30 8) Mode de réalisation n°8.

On se place dans le cas précédent, sans que la possibilité du masquage soit offerte. La solution consiste à:

- pour les liaisons internes et externes à procéder de la même façon que dans le mode n°7.
- 35 - pour les équations CI⁸ et CII⁸, celles-ci seront obtenues en reprenant les équations CI⁷ et CII⁷, dans lesquelles on supprime les termes en H.

9) Mode de réalisation n°9.

On se place à nouveau dans le cas général du mode de réalisation n°1, mais la question d'assembler des blocs entre eux n'est pas envisagée, et au contraire, on veut que le bloc soit bouclé sur lui-même. La solution est alors identique à celle décrite dans le mode de réalisation n°1, à cette modification près: les sorties $S_{N_j - v_j}^j$ sont reliées aux entrées $E_{N_j - v_j}^j$ du même bloc.

10) Mode de réalisation n°10.

On se place dans le cas du mode de réalisation n°2, avec bouclage du bloc sur lui-même comme ci-dessus dans le mode n°9. La solution est identique à celle du mode n°2, mais on relie les sorties correspondant aux dernières adresses d'un bloc, aux entrées correspondantes du même bloc.

11) Mode de réalisation n°11.

On se place dans le cas du mode n°3, en apportant la même modification que ci-dessus, mutatis mutandis.

12) Mode de réalisation n°12.

On se place dans le cas du mode n°4, en apportant la même modification que ci-dessus, mutatis mutandis.

13) Mode de réalisation n°13.

On se place dans le cas du mode n°5, en apportant la même modification que ci-dessus, mutatis mutandis.

14) Mode de réalisation n°14.

On se place dans le cas du mode n°6, en apportant la même modification que ci-dessus, mutatis mutandis.

15) Mode de réalisation n°15.

On se place dans le cas du mode n°7, en apportant la même modification que ci-dessus, mutatis mutandis.

16) Mode de réalisation n°16.

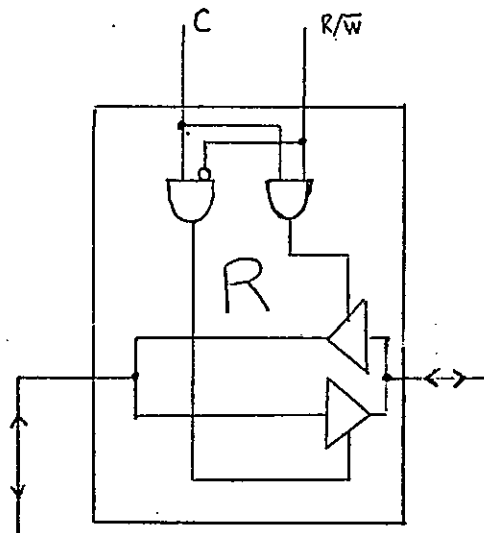
On se place dans le cas du mode n°8, en apportant la même modification que ci-dessus, mutatis mutandis.

REMERCIEMENTS

Nous remercions P.Isoardi pour les discussions fructueuses menées avec lui sur ce sujet, et notamment pour la suggestion d'incorporer aux dispositifs proposés une condition de masquage.

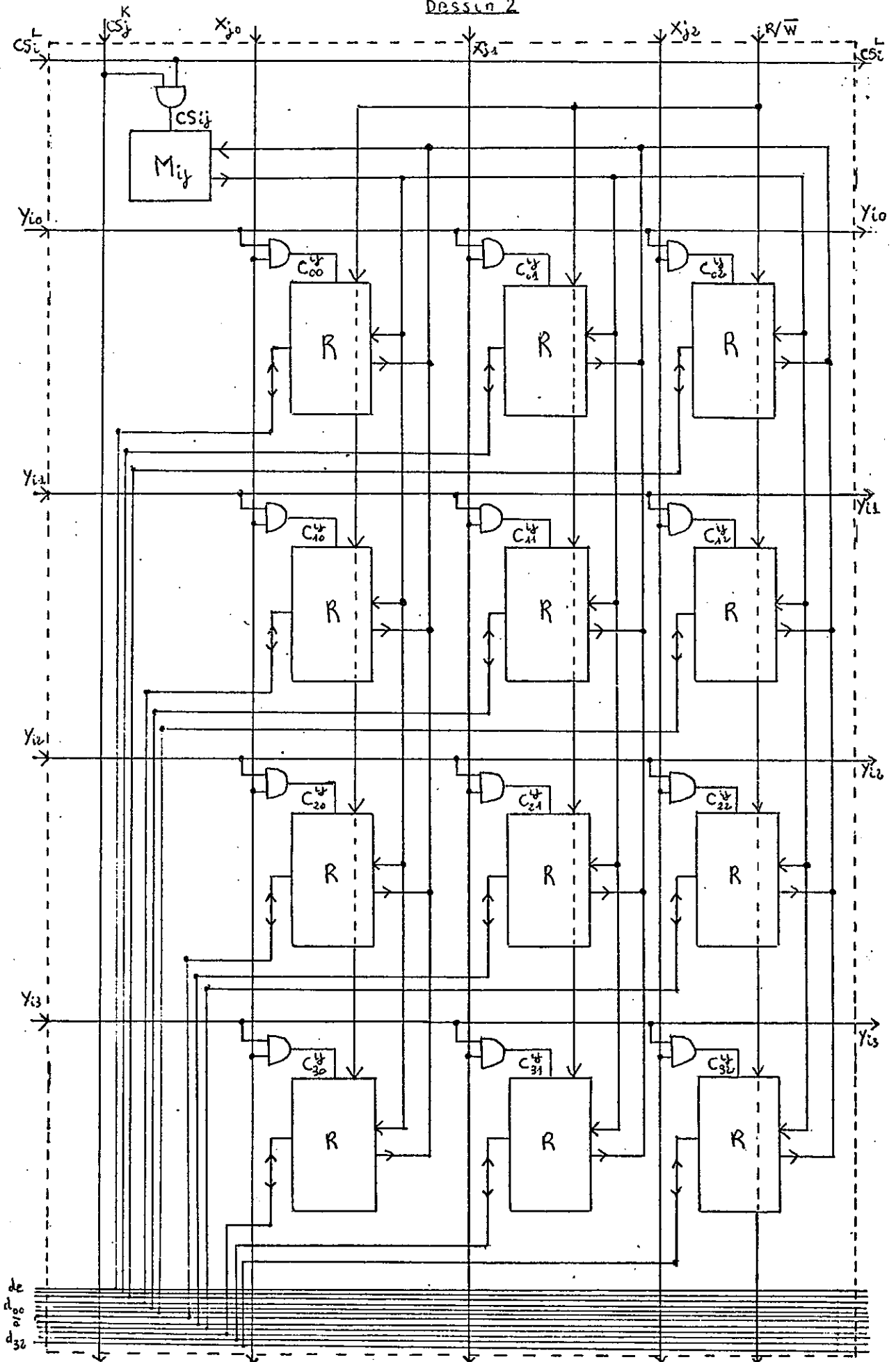
BIBLIOGRAPHIE

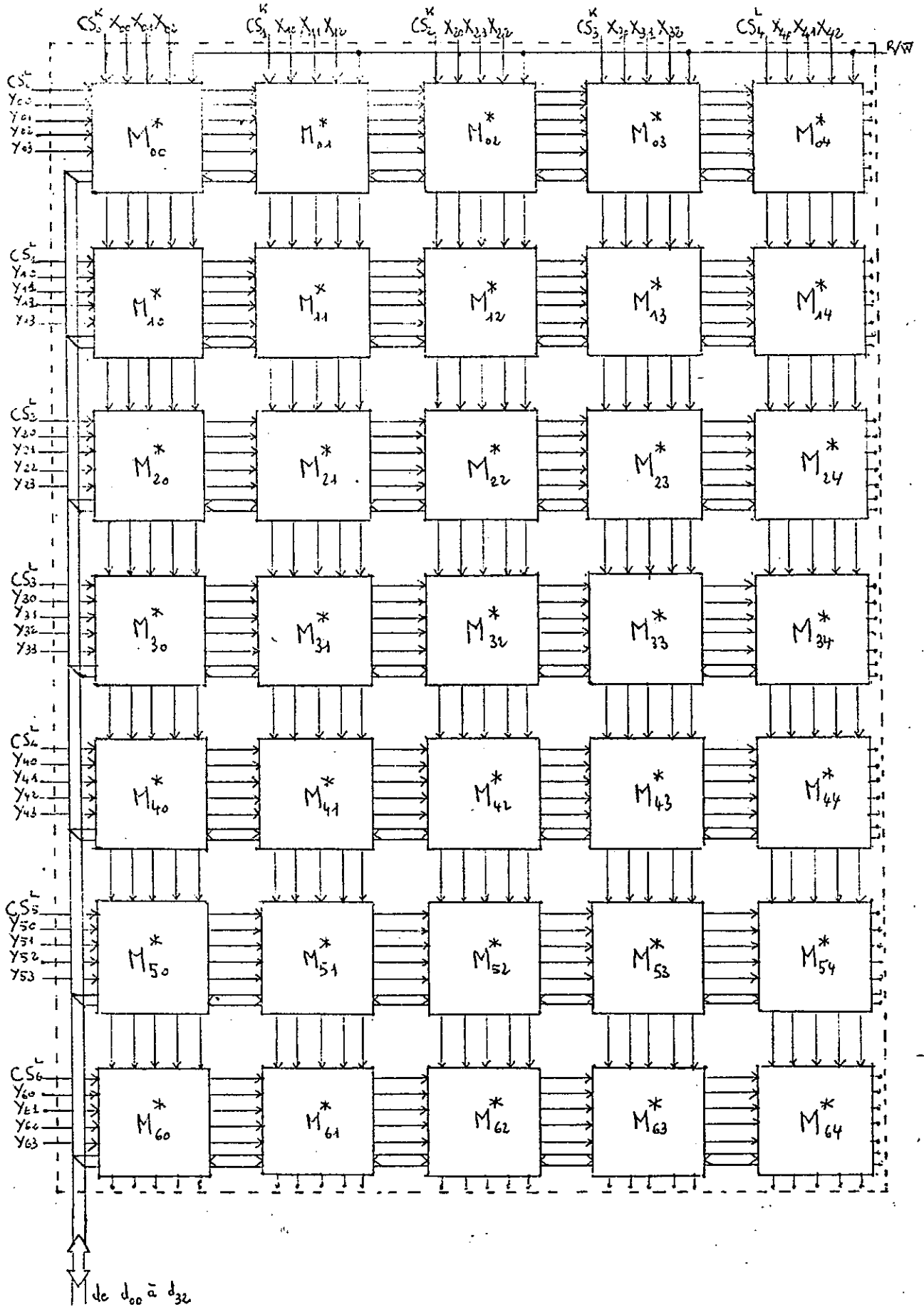
"Nouvelle structure de mémoires adressables par une fenêtre à déplacement continu".IRMA Strasbourg. P.Isoardi et J.Ph.Lehmann. Janvier 1985.



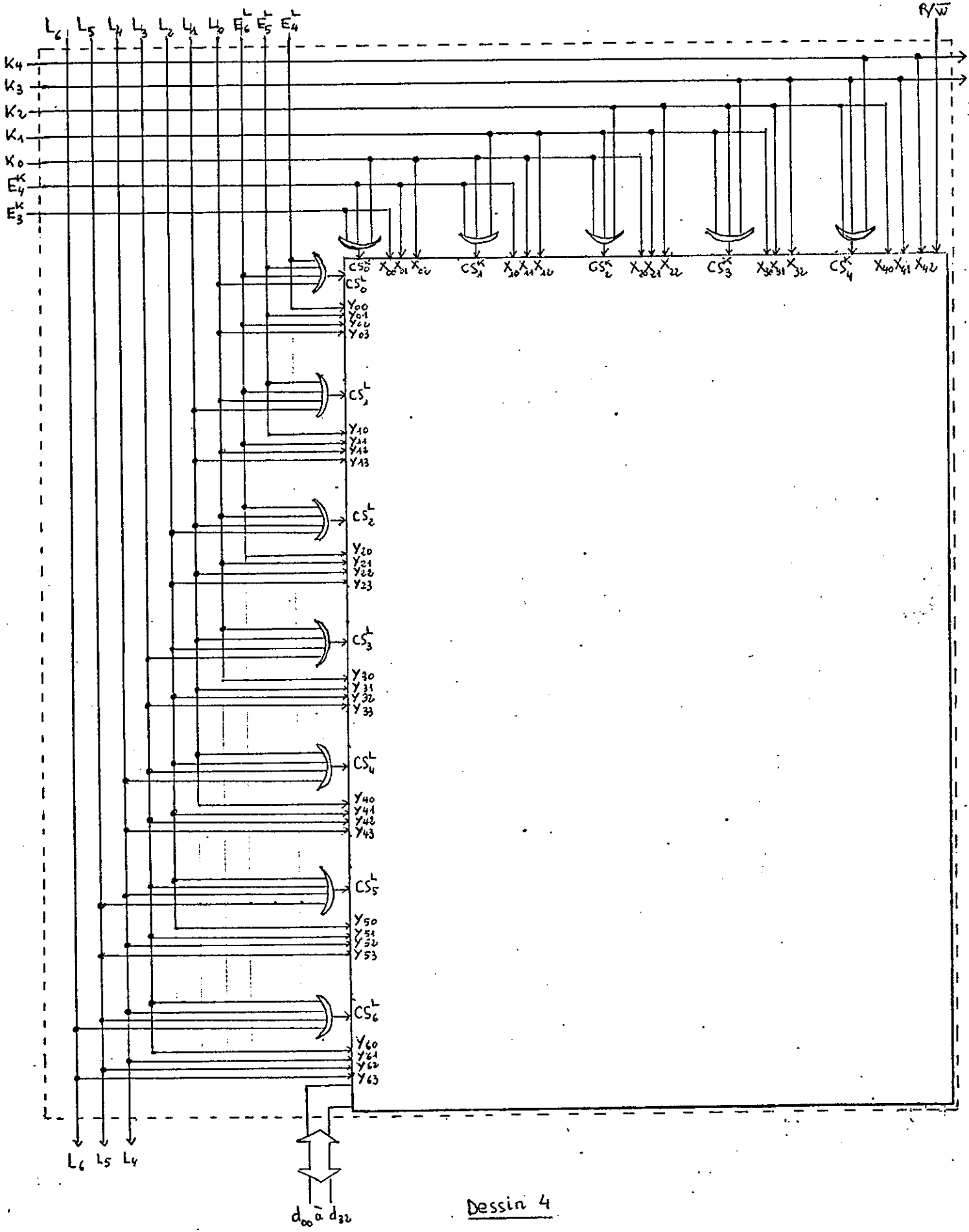
Dessin 1

Dessin 2



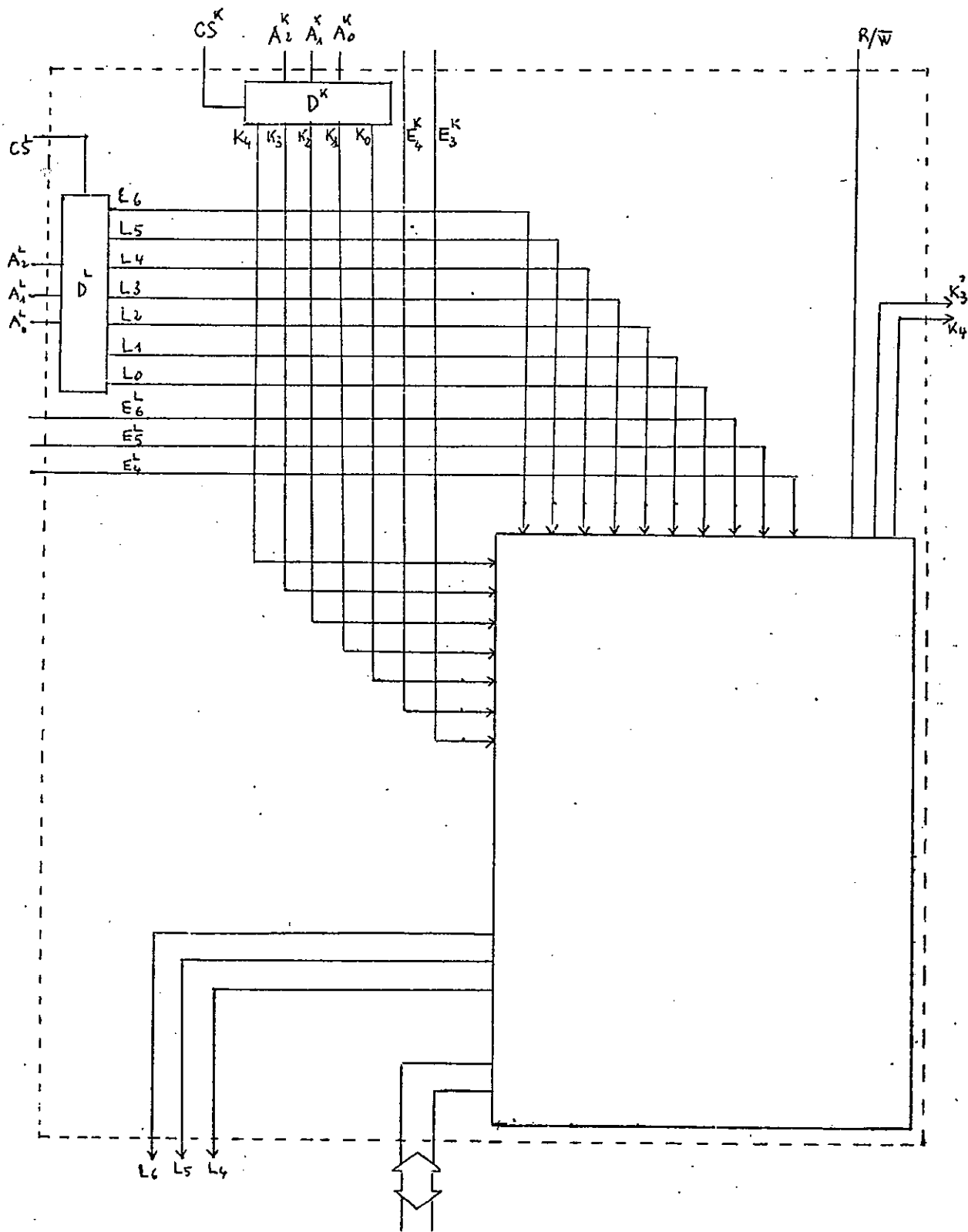


Dessin 3



Dessin 4

Dessin 5



1st INTERNATIONAL CONFERENCE

1^{re} CONFÉRENCE INTERNATIONALE

2, 3, 4 Septembre 1986 - AIX-EN-PROVENCE - FRANCE

**INTERNATIONAL PATRON
COMMITTEE
COMITÉ INTERNATIONAL
DE PATRONAGE**

Chairman / Président :
Professor H.A. SIMON,
Carnegie Mellon University, Pittsburgh

**INTERNATIONAL SCIENTIFIC
COMMITTEE
COMITÉ SCIENTIFIQUE
INTERNATIONAL**

Chairman / Président :
Pr. J.L. LE MOIGNE,
G.R.A.S.C.E. ENRS 935,
Faculté d'Economie Appliquée,
Université d'Aix-Marseille III FRANCE

**NATIONAL PROGRAM and
ORGANISATION COMMITTEE
COMITÉ NATIONAL
de PROGRAMME
et d'ORGANISATION**

Chairman / Président :
Dr. P. BOURGINE - COMAGREF
Chercheur associé au Centre Mondial
de l'Informatique - TOULOUSE

**CONFERENCE EDITOR
(Preprint and proceedings)
ÉDITEUR des ACTES**

Dr. J.L. ROOS
Centre d'Economie Industrielle
Faculté des Sciences Économiques
Citéau Lafarge - Route des Mûres
13290 LES MILLES
Tel. CE (42) 26.35.16
INSEE (42) 24.41.31
Telex : INSECH 440 253 F

**LOCAL SECRETARIAT
SECRETARIAT LOCAL**

Françoise LABAN
G.R.A.S.C.E. - CNRS 935
Faculté d'Economie Appliquée
5 Av. Victor Hugo
13100 AIX-EN-PROVENCE
Tel. (42) 27.95.01

**CONFERENCE SECRETARIAT
SECRETARIAT
de la CONFÉRENCE**

Elisabeth FAYOLA
Catherine FRACHON
A.F.C.E.T.
156, Bd. Perret
F 75017 PARIS
Tel. (1) 765.24.19
Telex : 290 163 EURTEL Code 325

**DE NOUVELLES MÉTHODES D'ÉTUDE EN SCIENCES
ÉCONOMIQUES, SCIENCES DE GESTION, SCIENCES
DE L'ORGANISATION**

PAR LES DÉVELOPPEMENTS DE L'INTELLIGENCE
ARTIFICIELLE, DES SYSTÈMES INTELLIGENTS ET DE LA
MODÉLISATION CONCEPTUELLE

THÈMES

**1.1. NOUVELLES MÉTHODES DE MODÉLISATION DES
SYSTÈMES COMPLEXES.**

- Acquisition et représentation des connaissances socio-économiques : les apports de l'intelligence Artificielle.
- Représentation des connaissances, formalisation du raisonnement et modélisation conceptuelle dans les sciences sociales.
- Conception des systèmes intelligents et application aux études économiques et organisationnelles.
- Conception de systèmes experts appliqués aux études industrielles et financières.
- Modélisation assistée par ordinateur : modèles micro, logiques d'études statistiques et systèmes de planification.
- Meta-modélisation : l'étude des méthodes de modélisation.

**1.2. SYSTÈMES D'INFORMATION INTELLIGENTS POUR LES
ÉTUDES ÉCONOMIQUES**

- Compréhension du langage économique traduction automatique de textes économiques.
- Reconnaissance des formes d'organisation économique
- Manipulation et recherches de documents.
- Application à l'enseignement et à la formation (conception de simulateurs...)

**1.3. LANGAGE ET MACHINES SPÉCIFIQUES POUR LES
ÉTUDES ÉCONOMIQUES.**

- Machine à concepts spécifiques, du type "cinquième génération".
- Base de données de volume important et base de données relationnelles.
- Systèmes interactifs et graphiques spécifiques.
- Techniques d'adressage et de recherches dans les fichiers de données économiques.

**2.1. SYSTÈMES D'AIDE À LA DÉCISION DANS LES CONTEXTES
ORGANISATIONNELS DE GESTION.**

- Système d'information intelligent et système interactif d'aide à la décision (S.I.A.D.).
- Procédure de validation pour les S.I.A.D.
- Problèmes spécifiques d'interconnexion des systèmes de données et des S.I.A.D.
- Systèmes de simulation en planification : jeux stratégiques.

2.2. MODÈLES INTELLIGENTS DE PRÉVISION ET D'ÉTUDES.

- Modèles de planification et d'évaluation
- Modèles macro et micro économiques
- Modèles d'études en production et distribution
- Systèmes Experts en analyse financière, analyse du risque, gestion de portefeuille.

2.3. SYSTÈME INTELLIGENT DE GESTION ADMINISTRATIVE.

- Gestion du personnel, paye...
- Fiscalité, systèmes légaux...
- Comptabilité de gestion et systèmes connexes

2.4. MODÉLISATION DES GRANDS SYSTÈMES.

- Planification régionale et urbaine.
- Études d'impact.
- Gestion des grandes ressources collectives
- Organisation des grands systèmes de santé
- Étude des systèmes de transport et communication.

2.5. SYSTÈMES DE DIAGNOSTICS.

- Systèmes généraux de diagnostic
- Systèmes nouveaux de détection des dangers
- Systèmes de sécurité et systèmes de prévention

1 - MÉTHODES ET PROTOTYPES

2 - DOMAINES D'APPLICATION

OBJECTIF GÉNÉRAL

La conférence sera consacrée aux développements méthodologiques, désormais prometteurs, que connaissent les études et recherches dans les domaines des Sciences et techniques des Systèmes Sociaux (Sciences économiques, Sciences des Organisations, Sciences de Gestion), suscitées par la progression contemporaine de l'Intelligence Artificielle et des domaines connexes : Systèmes Intelligents, systèmes de reconnaissance des formes, Systèmes Experts, formalisation en résolution de problème et en raisonnements automatisés, techniques de simulation et de jeux interactifs, modélisation et conception assistées par ordinateur.

On se propose d'explorer les voies ouvertes par la fertilisation mutuelle suscitée depuis quelques années par les recherches en modélisation intelligente en Sciences Sociales et en Sciences de l'Ingénieur : en mettant notamment en valeur les expérimentations par simulation, les applications spécifiques et les recherches plus concrètes caractéristiques tant des progrès épidémiologiques et méthodologiques des études que des contenus plus proprement pratiques des diverses techniques de modélisation des systèmes complexes.

**PRÉSENTATION et SÉLECTION
des COMMUNICATIONS**

L'objectif de la conférence est de susciter des contributions orientées en valeur non seulement la tenue de la fertilisation mutuelle de l'économique et de l'intelligence Artificielle, mais aussi la pertinence et l'intérêt des recherches dans ces domaines pour assister les études d'intervention dans les systèmes socio-techniques complexes.

Comme ce domaine de recherche est encore relativement nouveau (bien qu'il bénéficie de l'intense activité de recherche déployée depuis cinquante ans en Recherche Opérationnelle, en Sciences de Gestion et en Sciences Économiques), les comités de programme veilleront à privilégier à la fois les qualités d'ouverture thématique et de rigueur méthodologique.

Chaque projet de contribution fera l'objet d'au moins trois rapports de lecture indépendants. Les textes proposés pourront être soit brefs (trois pages) soit longs (6 pages), compte tenu du thème abordé et des mémoires exposés. La procédure de sélection sera conforme aux normes internationales usuelles, à savoir ici les règles établies par l'IFAC, dans son "manuel d'organisation des conférences", 1979 (Copie adressée sur demande).

**PROCÉDURE de PRÉSENTATION
des COMMUNICATIONS**

Les textes complets (trois pages ou six pages) rédigés en français ou en anglais devront parvenir en quatre exemplaires au :

Secrétariat de la conférence internationale
L'ÉCONOMIQUE ET L'INTELLIGENCE ARTIFICIELLE
c/o Professeur Jean-Louis Le MOIGNE
Faculté d'Economie Appliquée G.R.A.S.C.E. (CNRS 935)
5, Avenue Victor Hugo - 13100 AIX-EN-PROVENCE - FRANCE

La page de garde des communications précisera : Titre, nom et adresse des auteurs, résumé (en français et français pour les communications rédigées en français), et thèmes.

- 1^{er} Août 1985 : Réception des bulletins d'intention
- 6 Janvier 1986 : Réception des textes complets en quatre exemplaires
- 15 Mars 1986 : Notification aux auteurs
- 15 Mai 1986 : Réception des textes définitifs pour les actes (publiés par l'AFCEI)

Une sélection des communications sera publiée en langue anglaise par Pergamon Press après la conférence.

Seules peuvent être présentées des communications non publiées. L'IFAC dissuade des ordres de publication.



ASSOCIATION FRANÇAISE POUR LA CYBERNÉTIQUE
ÉCONOMIQUE ET TECHNIQUE
ASSOCIATION RECONNUE D'UTILITÉ PUBLIQUE PAR
DÉCRET DU 24 AOÛT 1976.



