

INFORMATIQUE  
FONDAMENTALE  
ET  
APPLICATIONS

## Sommaire

Comité de  
rédaction:

E. Bianco

R. Cusin

P. Isoardi

J.P. Lehmann

R. Stutzmann

- P 1 -EDITORIAL : Univers et expression.  
P 4 -SYDOC: un système de gestion de bases  
documentaires.  
P 26 -NOTION DE SYSTEME: système logiciel et  
système machine.  
P 46 -Représentation des connaissances biblio-  
graphiques à la lumière des réseaux de  
Pétri dans le cadre d'un système d'aide à  
la conception de thésaurus.  
P 70 -VOUZZAVEDIBISAR: L'avenir de l'ordinateur.

Dépositaire:

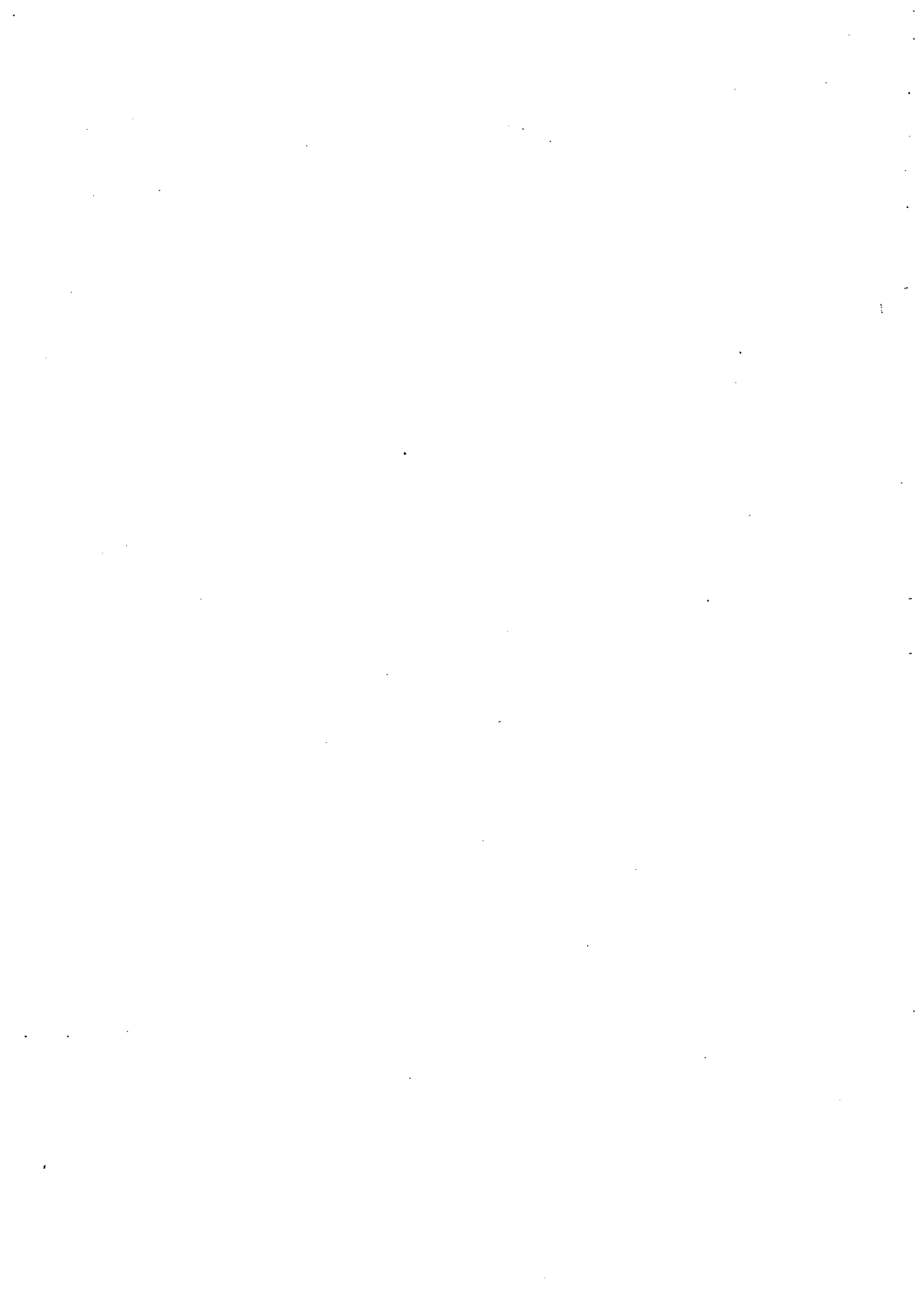
G. Ambard

Décembre ~~1986~~ 1985

ADRESSE POSTALE : FACULTÉ DES SCIENCES DE LUMINY

MATHÉMATIQUE-INFORMATIQUE - LITAM - BAT. TPR2 - 9<sup>ème</sup> ETAGE

CASE 901 - 70, ROUTE LÉON LACHAMP - 13288 MARSEILLE CEDEX 9 - ☎ (91) 41.01.40 POSTE 3294 - 3110



## Editorial

*e. Lianco*

### Univers et expression.

Immense pendule à la mesure de l'horloge du Père Eternel, la Comète vient de frôler nos têtes. Un coup de plus. Prochain coup dans soixante et seize ans: une seconde de cosmos. La dernière seconde nous avait apporté une terreur bleue, bien vite oubliée, l'Humanité n'a pas beaucoup de mémoire.

Anneaux magnifiques et inattendus d'Uranus, Miranda satellite étrangement cabossé que nous révèle Voyager qui fonce vers Pluton. Que nous réserve cette ultime planète protégée des mateurs indiscrets par ses milliards de kilomètres: un proche voisinage cosmique.

L'imagination nous projette de la contemplation de myriades de microorganismes à celle des champs de galaxie en expansion sans grand effort. Nous arrivons à nous faire une image de plus en plus précise de tout ce qui nous entoure, ou tout au moins telle est notre illusion.

Mais pour ce qui est de nous faire une idée de cette force dont on use pour appréhender l'environnement, de cet intérieur qui jongle avec ces images, alors là, le mystère reste entier.

Qu'est-ce que l'imagination ?

On peut bâtir des images: l'essentiel reste toujours en dehors

de l'image. On essaie de cerner la question en approchant à petits pas comptés, et en surveillant bien que rien n'échappe: au moment où l'on pense qu'on tient tout dans sa main... erreur, on ne tient qu'un petit bout d'illusion, l'essentiel s'est encor échappé.

La faillite du système de Platon à Descartes. Doit-on être surpris ? oui et non.

Oui si l'on est convaincu que le Bull-Dozer, instrument universel pour aplanir les difficultés est définitivement destiné à assurer le bonheur éternel et la paix des ménages.

Un peu moins quand on se demande à quoi devait servir l'outil en question, et que l'on constate qu'il n'avait d'autre utilité que de vider les choses de leur substance pour pouvoir mieux les manipuler.

C'était déjà pas mal.

En fait on avait réussi à remplacer l'être ou l'objet par son symbole, parcequ'on estimait qu'un symbole est plus commode à appréhender que l'objet lui-même. Il est bien vrai que si je veux établir une relation entre vitesse et énergie, il est plus facile d'écrire:

$$e = 1/2 m v^2$$

que d'exprimer les perceptions profondes que nous avons de ces phénomènes.

Mais j'ai bien peur qu'alors, construire la fonction inverse celle qui permettrait, justement de remonter de cette symbolique simpliste, squelettique, utile certes mais précisément vide de sens, jusqu'à la richesse complète de la signification, exige d'autres moyens que ceux qui, précisément vident les choses de leur sens.

Un peu comme si, après avoir pressé des fruits pour n'en con-

server que la peau, on essayait de reconstituer le fruit mais en utilisant la même presse.

Or, la moulinette moderne à trafiquer du symbole c'est l'ordinateur. Et l'on voudrait qu'avec des empilements compliqués de peaux mortes on refasse de beaux fruits juteux. Tout au plus me semble-t-il, et encore faut-il que le Maître d'oeuvres soit un artiste, réussira-t-on une belle affiche de publicité.

Ce qui n'est déjà pas si mal si le spectateur arrive à saliver.

Sans s'attarder aux nuances littéraires qui en modulent l'emploi, pourrait-on véritablement délimiter à petits coups de hachoir les domaines des mots tels que:

imagination, idée, esprit, pensée, réflexion, entendement, méditation  
conscience, raisonnement, rêverie, rêve, cauchemar, **amour**.

Et j'en oublie un. C'est la question d'un Siècle, mais le Siècle se meurt.

SYDOC

UN SYSTEME DE GESTION DE BASES DOCUMENTAIRES

M. T. LASKRI

C.R. Subject Classification informatics : [H31], [H32],  
[H33], [H34]

Résumé

Ce présent article décrit la réalisation du système de gestion de bases documentaires "SYDOC" qui est destiné à gérer toute base documentaire dont l'utilisateur définit la structure.

SYDOC  
-----

UN SYSTEME DE GESTION DE BASES DOCUMENTAIRES  
-----

1. Introduction.  
-----

Cette note technique décrit la réalisation du SYstème de gestion de bases DOCumentaires "SYDOC" et par conséquence, elle aborde le côté conception du système et non pas la manière de son utilisation.

Un manuell d'utilisateur [1] est prévu à cet effet et est donc disponible au L.I.T.A.M. (\*) dont le directeur est Monsieur le professeur Edmond BIANCO.

Au départ, la réalisation de "SYDOC" a été motivée par le besoin d'informatiser la bibliothèque du département de Mathématique-Informatique dont la gestion manuelle devient de plus en plus lourde et pénible. Non seulement "SYDOC" répond parfaitement à la gestion de cette tâche, mais se révèle apte à gérer toute base documentaire dont l'utilisateur peut définir la structure lui meme.

"SYDOC" a été réalisé sur le Micral 9050 en pascal.

(\*) : Laboratoire d'Informatique Théorique et Applications de Marseille.

## 2. L'informatique documentaire.

---

L'informatique documentaire [2] est un domaine où de nombreuses recherches ont été effectuées, et par conséquence des systèmes documentaires, dont le principal rôle est à peu près identique à celui des documentalistes ont été réalisés; à savoir qu'à partir de documents enregistrés en mémoire, ces systèmes essayent de donner aux utilisateurs présentant une demande, les références des documents qui conviennent à leur requête.

## 3. Analyse de "SYDOC".

---

"SYDOC" fonctionne à base de menus. Il propose à tout instant un choix d'actions possibles. L'utilisateur lit l'écran et choisit une option qu'il désigne par une lettre. C'est donc un système suffisamment simple qu'on peut le placer parmi les systèmes assistés qui offrent aux utilisateurs la possibilité d'être guidés lors de l'établissement de leurs requêtes.

Les différentes commandes de "SYDOC" sont donc proposées sous forme de menus régulièrement affichés pour éviter aux utilisateurs d'avoir en tête le langage des commandes. Ainsi, il ne faut pas être familier à "SYDOC"



pour pouvoir faire des applications au terminal.

#### 4. Formalisme d'un fichier "SYDOC".

##### a) Définition d'un fichier.

Un fichier est un ensemble d'enregistrements ou chacun contient un certain nombre de renseignements. En terme informatique, un renseignement s'appelle "CHAMP" ou "ZONE".

Voici un exemple de fichier simple illustrant ces notions : un fichier Clients. Ainsi, sur chaque enregistrement on portera les renseignements suivants : le nom, le prénom, l'adresse, et le numéro de téléphone. Il y'a donc quatre champs par enregistrement, et autant d'enregistrements que de clients.

##### b) Création d'un fichier "SYDOC".

La création d'un fichier consiste à décrire quels types de renseignements doivent figurer sur chaque enregistrement.

L'utilisateur doit déclarer les noms des champs qu'il désire voir figurer dans le fichier, et il a la possibilité par la suite de modifier ou de changer complètement les champs qu'il a nommés .

La forme du nom du fichier est définie par la syntaxe du système d'exploitation.

En général un nom du fichier est composé de trois parties :

- le PRÉFIXE mentionne sur quelle unité de disquettes se trouve le fichier.

Les deux unités de disquettes habituellement présentes sur un micro-ordinateur sont A: et B: , le disque dur est souvent repéré par C: . Sur le Micral l'unité de disquettes s'appelle B: et le disque dur est repéré par A: .

Si le préfixe n'est pas mentionné dans le nom, l'unité sera prise par défaut selon qu'on soit en A:, B: ou C: .

- le NOM PROPREMENT DIT comporte huit caractères alphanumériques au plus au choix de l'utilisateur.

- Le SUFFIXE définit en général le type du fichier, il est constitué d'un point suivi de trois caractères au plus; il désigne en quelque sorte le nom de famille du fichier.

Puisque on travaille dans le domaine de "SYDOC", on peut ne pas préciser le suffixe : les fichiers "SYDOC" prennent automatiquement le suffixe ".SYD".

A la création d'un fichier "SYDOC", donc de type ".SYD", le système créera deux autres fichiers qui lui seront

nécessaires pour son fonctionnement :

- l'un de type ".CHP" où il insèrera le nombre de champs, le nombre de documents, et la liste des champs décrite par l'utilisateur :

- l'autre de type ".ADR" où il insèrera les adresses des documents pour y accéder directement dans le fichier de type ".SYD".

Exemples de fichiers

-----

B:FONDS	(fichier FONDS sur la disquette de l'unité B:, il n'y a pas de suffixe, donc "SYDOC" rajoutera .SYD )
---------	---

CLIENTS	(fichier CLIENTS, pas de préfixe, l'unité sera prise par défaut; pas de suffixe donc "SYDOC" rajoutera .SYD )
---------	---

A:COMMANDE.SYD	(fichier COMMANDE sur A: )
----------------	----------------------------

LISTE.DOC	(nom de fichier inutilisable car suffixe autre que .SYD )
-----------	---

Remarque :

- Le nombre de champs utilisés ne peut pas dépasser 99
- Une fonction est envisagée pour donner la possibilité d'ajouter ultérieurement de nouveaux champs.

c) Utilisation d'un fichier "SYDOC".

Un fichier documentaire "SYDOC" est donc caractérisé par un nom qui est celui du fichier. Lors de l'activation d'un fichier, "SYDOC" cherchera dans l'unité les trois fichiers nécessaires à son fonctionnement (de types ".SYD", ".CHP" et ".ADR"); si l'un de ces fichiers n'existe pas "SYDOC" mentionnera qu'il ne peut pas travailler.

5. Organisation des fichiers "SYDOC".

L'organisation des fichiers est l'un des points les plus importants de l'efficacité de l'ordinateur.

Organiser des fichiers, c'est décider comment :

- Ranger l'information dans la mémoire;
- Accéder à l'information, ou la retrouver quand on en a besoin.

Le fait de ranger l'information ou d'accéder à

l'information dépend :

- Des formats des articles constituant le fichier.
- Des caractéristiques d'exploitation des fichiers;
- Des types de mémoires disponibles;

a) Format des documents de fichier.

Tous les documents d'un même fichier contiennent les mêmes types de données placées dans le même ordre.

Le format définit :

- la structure des données ou la séquence suivant laquelle ces données se présentent;
- le type de données (exemple : auteur, titre, date ... etc);
- la longueur de chaque donnée, exprimée en nombre de caractères (pour le format fixe).

"SYDOC" utilise le format varié :

chaque document du fichier comporte un nombre différent de données et chacune de ces données peut avoir une longueur différente et variable dans le temps (cf. 6).

Soit un fichier comportant : numéro de document, auteur, titre, et année de parution.

1er document : 1986 LASKRI UN SYSTEME DE GESTION DE BASES DOCUMENTAIRES 1986;

2ème document: 629 DUPOND INFORMATIQUE ET DOCUMENTATION EN FRANCE EN 1968 1969;

L'avantage certain du format varié est d'économiser de la place en mémoire, car contrairement au format fixe, il n'est pas nécessaire d'ajouter des espaces blancs à côté des données les plus courtes, du moment que les documents peuvent avoir des longueurs très différentes. Ainsi, dans un fichier de format varié, on enregistre les documents à la suite les uns des autres sans se préoccuper de la longueur maximum possible de chaque donnée du document.

#### 6. Exploitation des fichiers.

L'exploitation d'un fichier en ordinateur est plus complexe en format varié qu'en format fixe, car il est indispensable d'identifier chacune des données (ou champs); en effet leur emplacement relatif ne permet plus de le faire.

A titre d'exemple, dans le 1er document, comment distinguer que 1986 est à la fois numéro du document et année de parution ? Et dans le 2ème document comment savoir que l'année de parution est 1969 et non 1968, la

première année rencontrée ?

Cette identification a été réalisée de la manière suivante :

Chaque document sera précédé d'un caractère spécial "@" pour distinguer les documents les uns des autres.

De plus, chaque champ ou donnée sera précédé du caractère spécial "\$" pour indiquer qu'une nouvelle donnée commence.

Enfin, pour caractériser le type de donnée, un code sera ajouté après ce caractère spécial "\$".

Dans l'exemple cité on aura :

01 = numéro du document

02 = auteur

03 = titre

04 = année de parution

Le fichier s'écrira alors comme suit :

```
@#011986#02LASKRI#03UN SYSTEME DE GESTION DE BASES
DOCUMENTAIRES#041986@#01629#02DUPOND#03INFORMATIQUE ET
DOCUMENTATION EN FRANCE EN 1968#041969@....
```

Il n'y a pas plus maintenant de confusion possible :

- 1986 est le numéro du document du 1er document (car situé après #01); et aussi l'année de parution (car situé après #04);

- 1968 fait partie du titre du 2ème document (car enregistré à l'intérieur de la suite de caractères commençant par #03) ;

- 1969 est l'année de parution du 2ème document (car situé après #04).

c) Mode d'enregistrement des documents.  
-----

Comme les fichiers "SYDOC" sont des fichiers de caractères, "SYDOC" sauvegarde l'adresse du caractère de début de chaque document (qui est "@") dans le fichier de type ".ADR".

Revenons à notre exemple et mentionnons maintenant les documents dans le fichier des adresses (de type ".ADR")

Données (".SYD")	@#011986#02LASKRI#03....@#01629#02..
-----	-----
Adresses (".ADR")	0 77

Ainsi, le 1er document du fichier "SYDOC" (.SYD) débute à l'adresse 0, et le 2ème document à l'adresse 77. Et ces deux valeurs (0 et 77) sont enregistrées dans le fichier des adresses (.ADR) respectivement dans la 1ère et la 2ème composantes; car c'est un fichier d'entiers où la position du premier composant est 0.

Pour accéder directement à un document de rang r (numéro), il suffit d'aller lire directement l'adresse de son début ("@" ) dans le fichier des adresses à la



position (r-1) et se positionner ensuite dans le fichier "SYDOC" à cette adresse lue.

Exemple :

fichier.SYD		fichier.ADR	
position	contenu	position	contenu
0	@	0	0
1	\$	→ 1	77
2	0		
3	1		
4	1		
5	9		
6	8		
7	6		
8	\$		
.	.		
.	.		
.	.		
→ 77	@		
78	\$		
79	0		
80	1		
81	6		
82	2		
83	9		
84	\$		
.	.		
.	.		
.	.		

Pour accéder au 2ème document, on se positionne au composant numéro 1 (2-1) du fichier.ADR et on lit le contenu (adresse 77), et on se positionne au composant numéro 77 dans le fichier.SYD.

#### 6. Fonctions de "SYDOC".

---

Pour fonctionner, "SYDOC" a besoin des trois fichiers suivants :

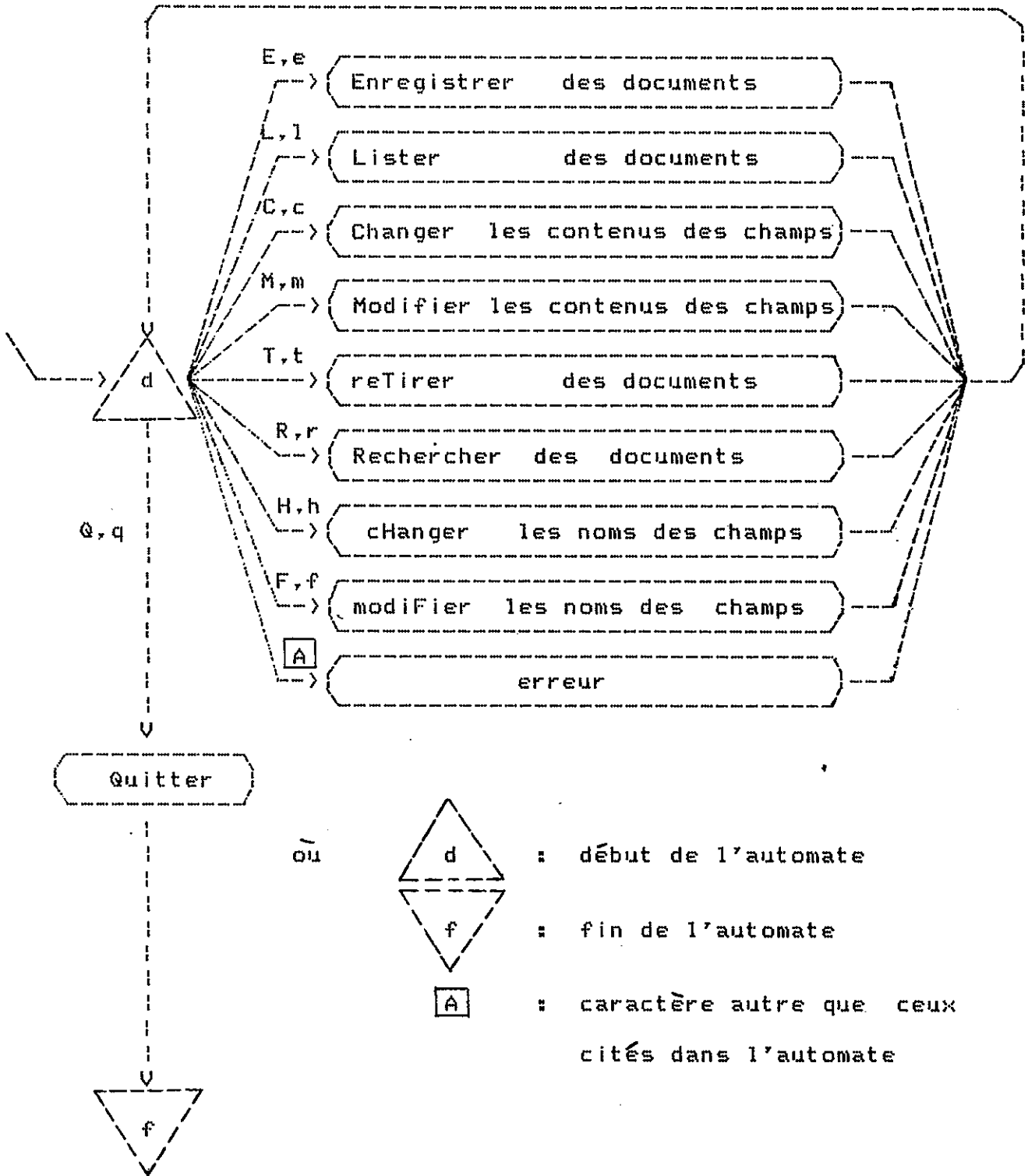
- Le fichier "SYDOC" de type .SYD où se trouvent les documents;

- Le fichier des champs de type .CHP où se trouve les informations relatives à la liste des champs;

- Le fichier des adresses de type .ADR où sont enregistrées les adresses des documents.

"SYDOC" possède les principales fonctions que peut avoir un système de gestion documentaire proposées sous forme d'un menu régulièrement affiché.

La structure du fonctionnement peut être représentée par l'automate suivant :



a) Enregistrement des documents.  
-----

C'est la fonction qui permet à l'utilisateur d'enregistrer ses documents. Lors d'un enregistrement, "SYDOC" place le pointeur de fichier à la fin du fichier .SYD, incrémente le nombre de documents de 1 (c'est à dire le prochain numéro du document), et propose ensuite de remplir tous les champs un par un selon (5.b).

L'utilisateur pourra rester dans cette phase pour enregistrer d'autres documents et revenir au menu principal quand il le veut.

b) Listing des documents.  
-----

Après avoir enregistré des documents, l'utilisateur a peut être commis des erreurs, pour cela la fonction de listing lui permet de lister les documents qu'il vient d'introduire dans le fichier pour pouvoir les relire.

On peut lister un document à la fois ou un ensemble de documents; il suffit de donner à "SYDOC" les rangs ou les numéros d'enregistrement, et l'accès à ces documents se fera selon (5.c) par l'intermédiaire du fichier de type .ADR .

c) Changement du contenu d'un champ.

Si dans un document déjà enregistré et ayant un numéro "x", on veut :

- changer complètement le contenu d'un ou plusieurs champs;

- remplir un ou des champs vides.

on utilise cette fonction en donnant le numéro "x" et "SYDOC" selon (5.c) se positionnera au document numéro "x" et proposera de lui donner le ou les champs dont on veut changer complètement le contenu.

d) Modification du contenu d'un champ.

Si dans un document déjà enregistré et ayant un numéro "x", on veut :

- Remplacer une chaîne de caractères par une autre;

- ajouter à la suite du contenu du champ un nouveau texte

on utilise cette fonction en donnant le numéro "x", et "SYDOC" selon (5.c) se positionnera au document concerné et demandera de lui préciser le champ dont on veut modifier le contenu et là, il suffit de lui donner la chaîne de caractères à modifier et la nouvelle chaîne de caractères qui remplacera l'ancienne.

e) Retrait de documents.

On peut retirer des documents du fichier s'ils sont périmés ou trop erronés; il suffit de donner à "SYDOC" le numéro du document à retirer.

"SYDOC" demande ensuite la confirmation du retrait et on peut répondre par

O (oui) : le document est retiré  
N (non) : le document est conservé  
L(lister) : "SYDOC" liste le document concerné selon (6.b) et redemande ensuite la confirmation.

f) Changement des noms des champs.

A la création d'un fichier, on donne la liste des champs des documents (cf. 4.b), et si par la suite on veut changer complètement le nom d'un ou de plusieurs champs on utilise cette fonction en donnant à "SYDOC" l'ancien nom et le nouveau nom qui remplacera dorénavant l'ancien.

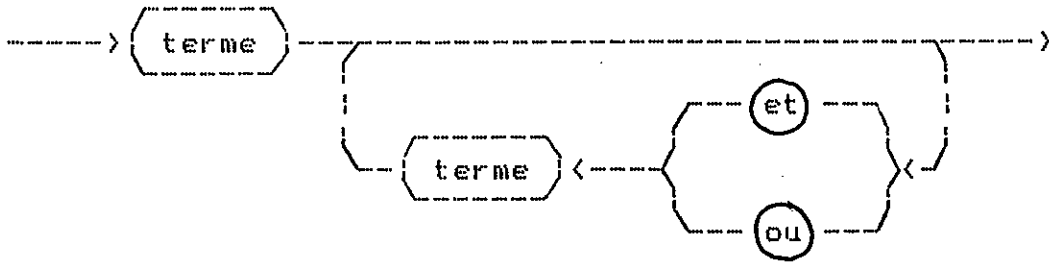
g) Modification des noms des champs.

Si dans un nom de champ, on veut remplacer une chaîne de caractères par une autre ou ajouter à la suite de l'ancien nom une chaîne, on utilise cette fonction; pour cela il faut donner à "SYDOC" l'ancienne chaîne à remplacer et la nouvelle chaîne qui la remplacera.

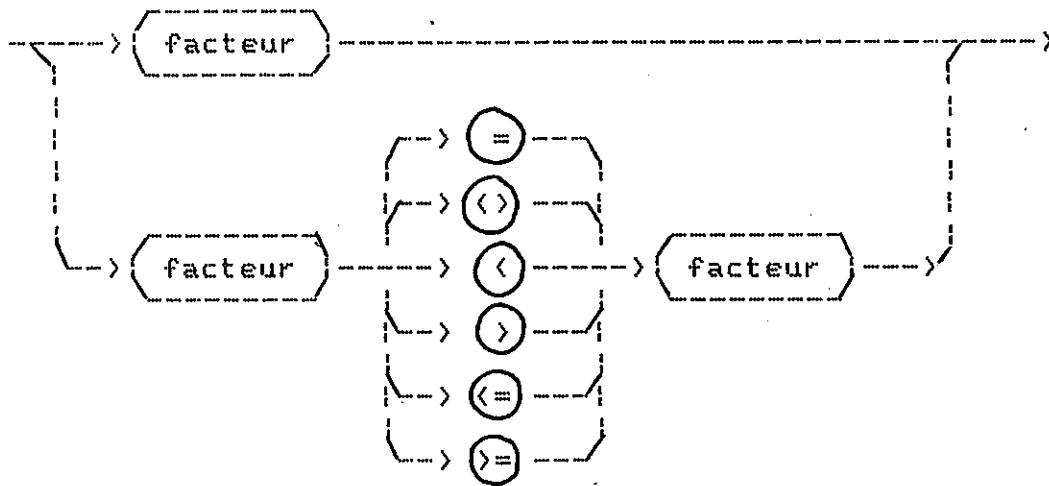
h) Recherche des documents.

C'est la fonction qui permet de rechercher ou de sélectionner des documents en utilisant un langage particulier qui est celui des questions dont la syntaxe est caractérisée par le diagramme syntaxique suivant :

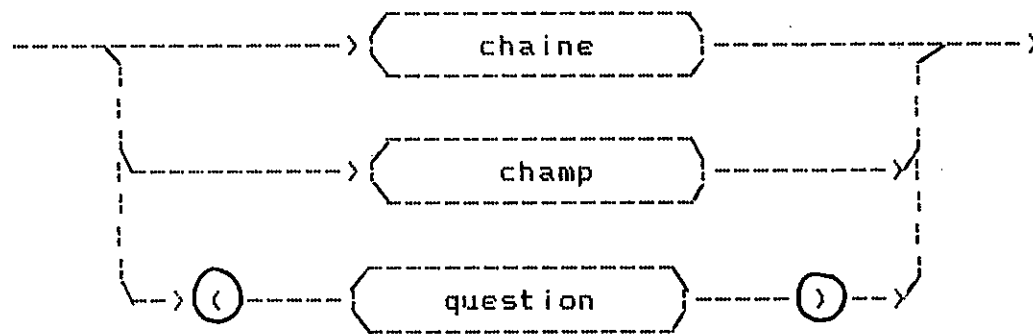
question



terme



facteur





Voici quelques questions respectant cette syntaxe :

- auteur = LASKRI et annee > 1980
- auteur = DUPOND ou LASKRI
- annee >= 1982

L'analyse syntaxique de ce langage des questions génère un code interprétable permettant de répondre à la question.

Le code généré par l'analyse syntaxique est une suite d'instructions qu'on peut présenter de la manière suivante :

une instruction de ce code est composée d'un opérateur et d'un opérande ( opérateur , opérande ) dont voici l'ensemble :

\* Deux instructions de chargements dans la zone de travail :

- 4 , i      chargement du i-ième élément de la table  
                  des chaînes
- 3 , i      chargement du contenu du i-ième champ

\* Une instruction pour l'opération de comparaison entre le contenu d'un champ et la chaîne :

- 2 , i      où :

- i = 1 : contenu du champ " = " à la chaîne
- i = 2 : contenu du champ " <> " à la chaîne
- i = 3 : contenu du champ " < " à la chaîne
- i = 4 : contenu du champ " > " à la chaîne
- i = 5 : contenu du champ " <= " à la chaîne
- i = 6 : contenu du champ " >= " à la chaîne

- i = 7 : contenu du champ " = " à la chaîne tronquée à gauche et à droite : \*chaîne\*
- i = 8 : contenu du champ " = " à la chaîne tronquée à droite : chaîne\*
- i = 9 : contenu du champ " = " à la chaîne tronquée à gauche : \*chaîne
- i = 10: contenu du champ " < > " à la chaîne tronquée à gauche et à droite
- i = 11: contenu du champ " < > " à la chaîne tronquée à droite
- i = 12: contenu du champ " < > " à la chaîne tronquée à gauche

\* Une instruction pour le "et" et le "ou" logique :

- 1 , 1 : comparaison "et" comparaison
- 1 , 2 : comparaison "ou" comparaison

\* Enfin une instruction la fin du code généré

- 5 , 0 : arrêt de l'interprétation

soit la question suivante :

auteur = LASKRI et année > 1980

on a la table des champs suivante :

4		annee	
3		titre	
2		auteur	
1		numero	

et l'analyse lexicale pour cette question donne la table des chaînes suivante :

2	1980
1	LASKRI

Ainsi l'analyse syntaxique de la question génère le code interprétable suivant :

3 , 2  
4 , 1  
2 , 1  
3 , 4  
4 , 2  
2 , 4  
1 , 1  
5 , 0

Et l'interprétation de ce code nous donne que le 1er document de notre exemple de départ répond à cette question.

### Références

- [1] : M. T. LASKRI   Manuel d'utilisateur de SYDOC  
          L.I.T.A.M.    1986
- [2] : G. VAN SLYPE   Systèmes documentaires et  
          ordinateur   1973

## NOTION DE SYSTEME

Systeme logiciel et systeme machine

*e. bianco*

C.R. Subject classification informatics: C53 C54 D31 D41

### Résumé.

Faisant intervenir la notion d'échange entièrement contrôlée, on reprend les deux formes de systèmes traités dans l'article précédent: le système vertical et le système horizontal, et on les complète en introduisant l'information à partir d'une mémoire de masse.

## NOTION DE SYSTEME

### ECHANGES

Le moment est venu de rajouter au langage de programmation en l'occurrence, la procédure formelle, le langage qui va permettre de maîtriser l'échange. Je vais reprendre en quelque sorte les instructions de la Machine de Nolin qui décrivent les transferts entre la mémoire centrale, ruban fini-borné, et les files externes illimitées:

```
ca := cfi
cfi := ca
Av fi
Ar fi
si cfi = '□' vers ek
```

Je vais tenir compte du fait pratique que dans les ordinateurs réels, ces opérations ne sont pas tout-à-fait de même nature que les opérations sur la mémoire centrale. En effet elles font intervenir des processus mécaniques en général plus lents et moins fiables que les processus strictement électroniques.

Il est apparu rapidement qu'il était bon et rentable de traiter l'échange en phénomène asynchrone. Déjà cet ancien ordinateur le GAMMA AET, vers la fin des années cinquante était doté d'une telle organisation, et on pouvait dérouler du programme en attendant les fameux "points machines" de la tabulatrice.

J'avais donc essayé dès les années soixante de formaliser

entièrement l'asynchronisme de l'échange, d'une manière cohérente. Cela m'avait conduit à concevoir la notion de calculateur d'échange, organe destiné à travailler indépendamment du calculateur central. Il devenait alors nécessaire de définir un protocole de communication entre ces deux entités, ceci ne pouvait se faire complètement que dans le cadre d'une théorie générale de la notion de système [cf.1].

Car la gestion de l'échange en asynchrone n'a d'autre but que de récupérer le temps d'attente (qui peut être long) pour dérouler du programme. Or, que dérouler et comment le dérouler ?

Voilà une question qui met en jeu une forte organisation. Je vais donc exposer ici l'aspect du protocole que doit respecter le calculateur central par rapport à l'échange.

L'autre aspect, celui du calculateur d'échange, sera exposé dans un prochain article. Je vais faire une simple hypothèse en deux points, dont nous verrons les conséquences ultérieurement:

#### Hypothèse 1.

Je me donne les moyens de choisir une configuration en mémoire centrale, éventuellement évolutive, à laquelle pourront accéder à la fois calculateur central et calculateur d'échange, strictement à tout instant, quand cela est nécessaire et sans restriction.

#### Hypothèse 2.

Les calculateur central et calculateur d'échange travailleront tout-à-fait indépendamment l'un de l'autre, n'ayant en commun que la configuration définie dans l'hypothèse 1, qui leur permet d'échanger l'information.

#### NOTION DE CARTOUCHE.

La configuration commune au calculateur central et au calcu-

lateur d'échange comporte le cartouche qui sert à décrire l'échange en en comportant les paramètres, et l'image de file externe qui contient l'information à transvaser.

Le cartouche contient donc les données fournies au calculateur d'échange, qui décrivent le fichier selon sa structure et sa localisation en mémoire centrale, et également un état qui permet au calculateur central de surveiller le déroulement des opérations. Si on se place du côté du calculateur d'échange, celui-ci a besoin de ces informations pour connaître la nature du fichier à déplacer, sa localisation externe et interne et l'état de l'opération au cours de son déroulement.

Le cartouche présenté plus loin n'est pas exhaustif et pourrait comporter d'autres indications, par exemple si le type de fichier correspond à une imprimante: une indication de cadrage de la ligne.

Je vais l'utiliser cependant sous cette forme simple et montrer ce que deviennent les systèmes verticaux et horizontaux quand ils prennent en compte les échanges. Mon but consiste à étudier l'impact fondamental de l'apport nouveau: l'échange, sur la structure du système en évitant de noyer l'essentiel dans le détail.

Afin de réaliser l'indépendance des calculs j'introduis dans la machine une nouvelle case que je désigne par "E", et qui joue pour le calculateur d'échange le même rôle que "I" pour le calculateur central. Le contenu de E est calculé et chargé par le calculateur central, par exemple ainsi:

$$l_E := l_I + l_6 ;$$

Si, dans la case 6 se trouve la distance des origines. Et  $l_E$  représente l'origine du cartouche. Le calculateur d'échange tra-

vaille donc sur une configuration d'origine  $[E]$  exactement comme le calculateur central travaille sur une configuration d'origine  $[I]$ .

Les adresses ainsi utilisées par le calculateur d'échange sont bien entendu calculées par rapport à cette origine du cartouche.

### SYSTEME VERTICAL AVEC ECHANGES.

Dans l'exemple du Bulletin N° 10 je supposai préenregistré en mémoire, une file descriptive d'une suite de programmes à dérouler, la suite des codes de ces programmes, et leurs configurations ainsi que les jeux de paramètres effectifs. J'appellerai "programme général interne", la file descriptive en question. J'avais ainsi en mémoire un tel programme général interne:

(f)(p1)(d1)(n1)(a11)(a12)...(a1n1)(p2)(d2)(n2)(a21)(a22)...  
(a2n2)... ... (pf)(df)(nf)(af1)(af2)...(afnf)

suivi de:

(code p1)((lp1) c1)(code p2)((lp2) c2)... ... (code pf)((lpf)  
cf)

lpi désigne la liste des paramètres effectifs,

ci désigne la configuration qui contient d'ailleurs les lpi.

Je vais généraliser le traitement de la façon suivante, je soumetts au système une suite de disquettes, ou tout autre support de masse équivalent, qui vont contenir chacune une file descriptive que j'appellerai "programme général externe", suivi par des enregistrements distincts du code de chaque programme puis du jeu de paramètres effectifs.

Par paramètres effectifs j'entends, bien sur, toute variable



donnée fournie avec sa valeur, et variable résultat avec une valeur quelconque, au moins pour celles qui ne se confondent pas avec une variable donnée.

A partir des informations contenues dans le programme général externe, le système va pouvoir calculer le programme général interne.

Cette structure de système, très simple comme on va le constater, suppose qu'on ait au préalable, constitué un jeu de disquettes qu'on soumet au système, successivement. Dans chaque disquette on doit rencontrer, dans l'ordre, les informations suivantes.

N°	Contenu	
1	V	
2	g, np1, vp11, vp12, ..., vp1np1, VP1, VC1, NP1, NC1, np2, vp21, vp22, ..., vp2np2, VP2, VC2, NP2, NC2, ... .. npg, vpg1, vpg2, ..., vpgnpg, VPg, VCg, NPg, NCg.	
NP1= 3	P1	
NC1= 4	$\pi C1$	
NP2= 5	P2	$\pi Cj$ désigne la liste de paramètres effectifs qu'on place dans la configuration.
NC2= 6	$\pi C2$	
...	...	
NPg= i	Pg	npj est le nombre de paramètres effectifs.
NCg= i+1	$\pi Cg$	vpk1 est le volume de chacun d'entre eux

V est le volume de l'enregistrement N°2, qui est, lui-même le programme général externe, dont chacun des éléments comporte le

nombre de paramètres, le volume occupé par chacun des paramètres, puis le volume du programme, le volume global de la configuration et les N° d'enregistrement sur disquette du code programme et des paramètres effectifs.

Bien entendu ce contenu total peut être le résultat d'un travail réalisé 'à la main', mais il peut être également le résultat d'une compilation, en fait, le résultat du travail de l'un quelconque des programmes précisément soumis au système.

Ce système vertical fonctionne donc selon un rythme répétitif de deux phases consécutives: d'abord lecture d'un programme général externe, calcul et construction d'un programme général interne suivi de la lecture des différents programmes et jeux de paramètres qu'il décrit, puis déroulement de la suite des programmes mis en place, suivi enfin de l'extraction des paramètres.

Sur un drive de disquette, on présente une disquette qui contient un programme général externe et le jeu de programmes-paramètres correspondant, lorsque tout ce contenu est lu, on présente alors une disquette vierge sur laquelle on récupère les résultats. On pourra recommencer dans l'ordre: une disquette programme, une disquette vierge, autant de fois qu'on le désire.

Pour décrire ce système vertical je définis une configuration qui sera la sienne et dont je repère les cases par des identificateurs pour la commodité de lecture:

lm, q1, q2, ak, ak1, ak3, ak4, ak6, ak7, ak8 seront 10 index,  
ak2 et ak5 seront les deux variables locales.

L'index lm sert à réinitialiser l'adresse de la place utile désignée par q1, alors que q2 indique la borne maximum à ne pas dépasser. Pour simplifier un peu je ne ferai pas ce contrôle ici.

Ce système vertical peut s'écrire:

```
    proc sysvert ;
index 10 ;
var loc 2 : 1 , 1 ;
syst0 : [ak,0 := 'masse' ;
        [ak,1 := 'direct' ;
        [ak,2 := 'demandé' ;
        [ak,3 := 'in' ;
        [ak,4 := 1 ;           Lecture du VOLUME du program-
        [ak,5 := 8 ;           me général externe.
        [ak,6 := 1 ;
        [E := [I + [ak ;
    echange ;
syst1  : si [ak,2 ≠ 'terminé' vers syst1 ;
        [ak,2 := 'demandé' ;
        [ak,4 := 2 ;           Lecture du PROGRAMME GENERAL
        [ak,5 := 9 ;           EXTERNE.
        [ak,6 := [ak,8 ;
    échange ;
syst2  : si [ak,2 ≠ 'terminé' vers syst2 ;
        [lm := [ak,8 + [ak ;
        [lm := [lm + 9 ;
        [q1 := [lm ;
syst3  : [ak1 := [ak + 8 ;
        [ak7 := [ak1 + 1 ;
        [ak2,0 := [ak7,0 ;
        [ak5,0 := 0 ;
        [ak7 := [ak7 + 1 ;
```

```

syst7 : si [ak2,0 = 0 vers syst8 ;
      [ak6 := [ak7 + [ak7,0 ;
      [ak5,0 := [ak5,0 + [ak6,1 ;
      [ak5,0 := [ak5,0 + [ak6,2 ;
      [ak7 := [ak6 + 5 ;
      [ak2,0 := [ak2,0 - 1 ;
      vers syst7 ;

```

```

syst8 : [ak6 := [q1 + [ak5,0 ;
      [ak4 := [ak6 ;
      [ak4,0 := [ak1,1 ;
      [ak4 := [ak4 + 1 ;
      [ak5,0 := [ak1,1 ;
      [ak1 := [ak1 + 2 ;

```

```

syst13 : si [ak5,0 = 0 vers syst14 ;
      [ak4,0 := [q1 ;
      [ak7 := [ak1 + [ak1,0 ;
      [ak4,1 := [ak7,1 + [q1 ;
      [ak4,2 := [ak1,0 ;
      [ak2,0 := [ak1,0 ; (nombre de paramètres)
      [ak3 := [ak4 + 3 ;
      [ak7 := [ak1 + 1 ;

```

```

syst10 : si [ak2,0 = 0 vers syst9 ;
      [ak3,0 := [ak4,1 ;
      [ak4,1 := [ak4,1 + [ak7,0 ;
      [ak3 := [ak3 + 1 ;
      [ak2,0 := [ak2,0 - 1 ;
      [ak7 := [ak7 + 1 ;
      vers syst10 ;

```

+  
adresses des para-  
mètres.  
+

+ Calcul des éléments du PROGRAMME GENERE INTERNE et mise en place.  
+

```
syst9 : [ak,2 := 'demandé' ;
        [ak,4 := [ak7,2 ;
        [ak,5 := [q1 - [ak ;
        [ak,6 := [ak7,0 ;
        échange ;

syst11 : si [ak,2 ≠ 'terminé' vers syst11 ;
        [q1 := [q1 + [ak7,0 ;
        [ak,2 := 'demandé' ;
        [ak,4 := [ak7,3 ;
        [ak,5 := [q1 - [ak ;
        [ak,6 := 0 ;
        [ak8 := [ak1 + 1 ;
        [ak2,0 := [ak1,0 ;
syst22 : si [ak2,0 = 0 vers syst21 ;
        [ak,6 := [ak,6 + [ak8,0 ;
        [ak8 := [ak8 + 1 ;
        [ak2,0 := [ak2,0 - 1 ;
        vers syst22 ;

syst21 : échange ;

syst12 : si [ak,2 ≠ 'terminé' vers syst12 ;
        [q1 := [q1 + [ak7,1 ;
        [ak4 := [ak3 ;
        [ak5,0 := [ak5,0 - 1 ;
        [ak1 := [ak1 + [ak1,0 ;
        [ak1 := [ak1 + 5 ;
        vers syst13 ;
```

Introduction du  
programme.

Introduction de  
la configuration  
de paramètres  
effectifs.

Calcul du volume  
des paramètres  
effectifs à char-  
ger.

Ici s'achève la mise en mémoire à partir de la disquette des codes des programmes et des configurations de paramètres effectifs.

Maintenant commence la partie du système qui contrôle le déroulement:

```
syst14 : [ ak1 := [ ak6 ;
          [ ak2,0 := [ ak1,0 ;
          [ ak1 := [ ak1 + 1 ;
sys1 : si [ ak2,0 = 0 vers syssuit ;
       inspar (ak1) ;
       [ ak1 := [ ak1 + [ ak1,2 ;
       [ ak1 := [ ak1 + 3 ;
       [ ak2,0 := [ ak2,0 - 1 ;
       vers sys1 ;
syssuit : [ ak1 := [ ak + 9 ;
          [ ak,3 := 'out' ;           Sortie sur disquette
          [ ak8 := [ ak1 + 1 ;       des différentes listes
          [ ak5,0 := [ ak1,0 ;       de paramètres effectifs
          [ ak2,0 := [ ak8,0 ;       qui contiennent main-
          [ ak3 := 0 ;               tenant, les résultats.
          [ ak4 := 1 ;
          [ ak6 := [ ak6 + 1 ;
system4 : si [ ak5,0 = 0 vers syst0
system2 : si [ ak2,0 = 0 vers system1 ;
          [ ak3 := [ ak3 + [ ak8,1 ;
          [ ak8 := [ ak8 + 1 ;
          [ ak2,0 := [ ak2,0 - 1 ;
          vers system2 ;
system1 : [ ak,4 := [ ak4 ;
          [ ak,5 := [ ak6,1 - [ ak3 ;
          [ ak,6 := [ ak3 ;
```

échange ;

[ak6 := [ak6 + [ak6,2 ;

[ak6 := [ak6 + 3 ;

[ak5,0 := [ak5,0 - 1 ;

[ak4 := [ak4 + 1 ;

[ak8 := [ak8 + 5 ;

syster3 : si [ak,2 ≠ 'terminé' vers syster3 ;

[ak2,0 := [ak6,2 ;

[ak3 := 0 ;

vers syster4 ;

SYSTEME HORIZONTAL AVEC ECHANGES.

Nous allons constater que ce système, bien que plus subtil est au moins aussi simple de structure que le précédent. Les calculs les plus complexes restent de la même manière les calculs d'adresse pour implanter les programmes.

Je reprends strictement la même structure pour le PROGRAMME GENERAL EXTERNE, que je suppose toujours enregistré sur une disquette, et prêt à être lu sur son drive. Je renvoie plus haut pour sa description et je rappelle seulement ici la structure du PROGRAMME GENERAL INTERNE telle que je l'ai définie dans le Bulletin N°10. La machine universelle pour calculer l'insertion fractionnée a besoin d'une table d'informations dont l'adresse lui est fournie dans le code instruction:

insfract (n) ;

qui signifie que la case N°n contient l'adresse de la table en question. Dans cette table on dispose de la suite:

$p_i$  ,  $d_i$  ,  $\alpha_{ni}$  ,  $\alpha_{\#i}$

$p_i$  Accès au code du programme.

$d_i$  Accès à la configuration disponible.

$\alpha_{ni}$  Accès à la valeur de l'état du déroulement.

$\alpha_{\#i}$  Accès à la liste des paramètres effectifs.

Je vais donc concevoir un programme général interne qui sera exploré circulairement par le système et qui contiendra toute l'information nécessaire:

$(p_1, d_1, \alpha_{n1}, \alpha_{\#1}, V_{p1}, V_{c1}, V_{par1}, n_1)$  ,  $(p_2, d_2, \alpha_{n2}, \alpha_{\#2}, V_{p2}, V_{c2}, V_{par2}, n_2)$  , ... .. ,  $(p_n, d_n, \alpha_{nn}, \alpha_{\#n}, V_{pn}, V_{parn}, n_n)$

En plus de la table des paramètres de l'insertion fractionnée, je porte également:



V<sub>pi</sub> Volume du programme.

V<sub>ci</sub> Volume de la configuration disponible: le volume fourni moins celui occupé par les paramètres effectifs.

V<sub>pari</sub> Volume des paramètres effectifs.

n<sub>i</sub> Valeur de l'état du déroulement: 'demandé', 'en cours', 'terminé'.

Voilà donc un programme général interne de n éléments. Chacun de ces éléments sert d'image pour le contrôle du déroulement du programme qui lui correspond. A partir donc de la configuration libre indiquée par l'index q<sub>1</sub>, on va:

- 1) Calculer l'emplacement du programme général interne.
- 2) Calculer la place du programme général externe et le libre.
- 3) Dans le programme général interne on place tous les n<sub>i</sub> à la valeur: 'terminé', ceci pour assurer le lancement du calcul.
- 4) La boucle centrale du système démarre alors, qui va assurer:
  - 41) Le déroulement fractionné de chaque tâche.
  - 42) L'approvisionnement en programmes déroulables à la suite de l'achèvement de chaque programme.
  - 43) La sortie des résultats.

Quand le programme général externe est épuisé, le système réapprovisionne un nouveau programme général externe pour pouvoir continuer.

La procédure qui décrit ce système, travaille sur 11 index, que par commodité, toujours, je désigne par les noms suivants:

lm, q<sub>1</sub>, q<sub>2</sub>, a, ac, at, bc, bt, adc, x, y

de la même manière je désigne un jeu de 9 variables locales:

CE, Nbp, cte, Vpar,  $\phi$ ,  $\psi$ ,  $\psi^1$ ,  $\psi^2$ , ech

```
procédure syShor ;
  index 11 ;
  var loc 9 : 1 , 1 , 1 , 1 , 9 , 1 , 1 , 1 , 8 ;
  [lm := 128 ;
  [ $\phi$ ,0 := "Introduire nombre maximum de tâches,
puis mettre en place disquette des des programmes à dérouler:" ;
  deb : [q1 := [lm ;
        [ech,0 := "écran" ;
        [ech,1 := "dir" ;
        [ech,2 := "demandé" ;
        [ech,3 := "out" ;
        [ech,4 := 4 ;           N° de ligne.
        [ech,5 := [ $\phi$  ;       adresse 1ER mess.
        [ech,6 := 92 ;       Vol. de ce mess.
  échange
  syst0 : si [ech,2  $\neq$  "terminé" vers syst0 ;
        [ech,0 := "clavier" ;   Lecture du nombre de
        [ech,2 := "demandé" ;   programmes à faire tour-
        [ech,3 := "in" ;       ner.
        [ech,5 := [ $\psi$  ;
        [ech,6 := 1 ;
  échange
  syst1 : si [ech,2  $\neq$  "terminé" vers syst1 ;
        [ech,0 := "masse" ;     Lecture du volume du
        [ech,2 := "demandé" ;   programme général ex-
        [ech,4 := 1 ;           terne.
        [ech,5 := [ $\psi$ 1 ;
```

```

[ech,6 := 1 ;
echange ;
syst2 : si [ech,2 ≠ "terminé" vers syst2 ;
[at := [ψ,0 * 8 ;
[at := [at + [q1 ;
[ac := [q1 ;
[a := [q1 ;
[bt := [at + [ψ1,0 ;
[q1 := [bt ;
[Nbp,0 := [at,0 ;
[ech,2 := "demandé" ;
[ech,4 := 2 ;
[ech,5 := [at ;
[ech,6 := [ ;
échange ; [CE,0 := "libre" ;
syst3 : si [ech,2 ≠ "terminé" vers syst3 ;
syst5 : si [ac = [at vers syst4 ;
[ac,7 := "terminé" ; Initialisation des états
[ac := [ac + 8 ; des tâches pour que le
vers syst5 ; système puisse, au démar-
rangé, charger des program-
mes.

syst4 : [bc := [at + 1 ;
sys2 : [ac := [a ;
vers syst3 ;
sys1 : [ac := [ac + 8 ;
sys3 : si [ac = [at vers syst2 ;
si [ac,7 ≠ "en cours" vers syst7 ;
sys0 : inserfrac (ac) ;
vers syst1 ;

```

↑système de déroulement↑

```
sys7 : si [ac,7 = "terminé" vers sys71 ;
      si [ech,2 ≠ "terminé" vers sys1 ;
      si [ac,7 = "prog" vers sys8 ;
      si [ac,7 = "par" vers sys9 ;
sys71 : si [CE,0 ≠ "libre" vers sys1 ;
sys6 : si [Nbp,0 = 0 vers syst1 ;
sys65 : [ac,3 := [q1 ;           Accès à la liste des
      [q1,0 := [bc,0 ;           paramètres effectifs.
      [q1 := [q1 + 1 ;
      [x := [q1 + [bc,0 ;
      [cte,0 := [bc,0 ;
      [y := [x ;
      [adc := [bc + 1 ;
      [vpar,0 := 0 ;
sys61 : si [cte,0 = 0 vers sys60 ;
      [q1,0 := [y ;           Construction des adres-
      [q1 := [q1 + 1 ;           ses des paramètres ef-
      [y := [y + [adc,0 ;       fectifs dans la liste.
      [cte,0 := [cte,0 - 1 ;
      [vpar,0 := [vpar,0 + [adc,0 ;
      [adc := [adc + 1 ;
      vers sys61 ;
sys60 : [ac,7 := "par" ;
      [ech,2 := "demandé" ;
      [ech,4 := [adc,3 ;       N° d'enregistret par.eff.
      [ech,5 := [x ;           Adresse des valeurs.
      [ech,6 := [vpar,0 ;     Volume des par. effect.
      échange ;
```

```
[ac,1 := [y ;           Accès à la configuration.
[ac,2 := [ac + 7 ;       Accès à l'état.
[ac,0 := [x + [adc,1 ;   Localisation du program.
[ac,4 := [adc,0 ;        Volume du programme.
[ac,6 := [Vpar,0 ;       Volume des paramètres.
[ac,5 := [adc,1 - [ac,6 ; Volume de la config.
[CE,0 := "occupé" ;
vers sys1 ;

sys9 : [ac,7 := "prog" ;   Lecture vient d'être
[ech,2 := "demandé" ;     achevée, des paramètres
[ech,4 := [adc,2 ;        effectifs.
[ech,5 := [ac,0 ;
[ech,6 := [adc,0 ;
échange ;

[bc := [adc + 4 ;         Préparation de lecture
[q1 := [ac,0 + [adc,0 ;   du programme suivant.
[Nbp,0 := [Nbp,0 - 1 ;    Decompte de programmes
vers sys1 ;             lancés.

sys8 : [ac,7 := "en cours" ;
[CE,0 := "libre" ;
vers sys0 ;
```

En manière de conclusion, je soulignerai quelques points. J'ai fait l'hypothèse que l'information s'échangeait par un seul canal: celui du calculateur d'échange unique déroulant visiblement un seul programme. J'ai concentré mon attention sur un système déroulant en fractionné et égale répartition du travail: un nombre quelconque de tâches, dont il faut évidemment décider au dé-

but.

Je n'ai donc guère insisté sur le côté conversationnel que je reprendrai en profondeur lors d'un prochain article. Mais on peut déjà voir émerger et sous une forme particulièrement simple, une structure de système déjà extrêmement puissante.

Pour qu'elle soit complète, il reste encore à contrôler les débordements des configurations disponibles; en fait, vérifier lorsqu'on incrémente le contenu de  $q_1$  qu'il ne dépasse pas celui de  $q_2$ . Mais on constate que rajouter cela ne complique pas beaucoup la structure. Exploiter le débordement est un autre aspect de la notion de système qui sera étudié ultérieurement en détail.

Je tenais à souligner simplement le fait que se donner les outils adéquats, après une étude des propriétés fondamentales des systèmes permet de montrer qu'il s'agit d'objets qui demeurent simples. Et ce n'est pas en évacuant la difficulté vers l'extérieur, par exemple au niveau du langage de description, puisque j'ai conservé une sorte de langage-machine qui ne contient aucune notion complexe.

J'utilise en fait la puissance d'une hypothèse qui est à la base de la notion d'insertion fractionnée: le système est réduit à ses fonctions fondamentales car il utilise l'insertion fractionnée, et l'insertion fractionnée présuppose que les programmes insérés sont autojectifs.

C'est l'autojectivité qui est à la base de la condensation de la notion de système.

à suivre.



REPRESENTATION DES CONNAISSANCES BIBLIOGRAPHIQUES  
A LA LUMIERE DES RESEAUX DE PETRI DANS LE CADRE  
D'UN SYSTEME D'AIDE A LA CONCEPTION DE THESAURUS

J.M.KNIPPEL

C.R. Subject classification informatics : H.3.1, F.1.1, I.2.4

Résumé:

Dans le cadre d'un système d'aide à la conception de thésaurus, nous nous sommes penchés sur divers thésaurus existants, tel que le répertoire de Kent, afin d'en étudier la structure profonde.

Afin d'assurer la constitution et la maintenance d'un thésaurus en général pour représenter et organiser l'information, nous avons pensé au modèle dynamique des réseaux de Pétri.

Nous présentons ici un exemple de représentation des connaissances bibliographiques contenues dans les deux tomes du collectif d'auteurs G.W.BRAMS sur la théorie et la pratique des réseaux de Pétri (BRAMS 83).

L'article argumente:

- l'analyse par appréhension de son contenu bibliographique, l'identification des concepts représentant ce contenu, et la sélection des concepts nécessaires à une recherche ultérieure,
- l'organisation de l'information dans le langage documentaire par la reconnaissance des descripteurs permettant de représenter les concepts sélectionnés dans l'ouvrage (LASKRI et GALAND 85).



REPRESENTATION DES CONNAISSANCES BIBLIOGRAPHIQUES  
A LA LUMIERE DES RESEAUX DE PETRI DANS LE CADRE  
D'UN SYSTEME D'AIDE A LA CONCEPTION DE THESAURUS

PREAMBULE

Nous proposons comme cadre thématique déterminé pour l'analyse de première approche du langage naturel l'aide au diagnostic; qui est une des plus anciennes applications de l'informatique à la pratique médicale. Les méthodes d'analyse des données font partie des aspects théoriques que nous avons étudiés (KNIPPEL 82), (KNIPPEL 84). L'approche statistique est principalement numérique; nous envisageons le prolongement par une démarche de nature symbolique de la représentation des connaissances pour l'aide au diagnostic dans le domaine médical (cas des connaissances néphrologiques).

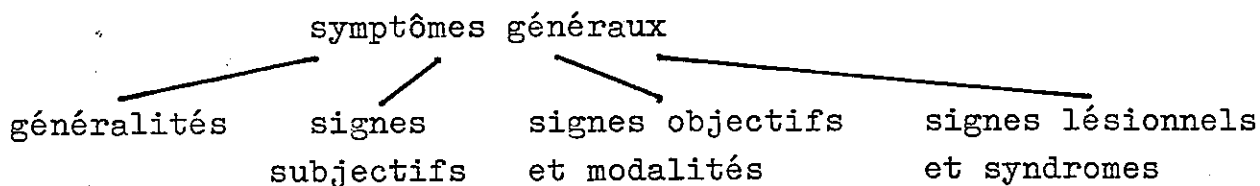
INTRODUCTION

Cette tâche s'inscrit dans le cadre des travaux sur l'aide aux thésaurus; et nous avons choisi comme thésaurus le répertoire de Kent (BROUSSALIAN 83). Nous considérons un ensemble de termes précis, en général des mots-clés, couvrant un groupe de concepts déterminés. Les termes sont organisés entre eux par le biais d'un réseau de différentes relations, lesquelles permettent de faire ressortir tout le contenu sémantique. Nous citons ici trois exemples de relations extraites du répertoire de Kent:

-synonymie ou équivalence:

compréhension, entendement

-hiérarchie:



-voisinage ou association:

rectum

.ténesme

ténesme

.rectum

douleur rénale

.vertiges

vertiges

.douleur rénale

Le cadre thématique du thésaurus choisi étant déterminé, nous suggérons ici pour exploiter les notions de, plage sémantique, concept, lien sémantique, chemin sémantique, thésaurus secondaire,

analyse de deuxième approche, fonds sémantique (LASKRI et GALAND 85), l'emploi des réseaux de Pétri.

### METHODOLOGIE

En 1962, C.A.Pétri proposa une représentation générale des processus sous forme de graphes, qui fut étendue par A.Holt en 1970. Ce dernier lui donna le nom de "réseaux de Pétri". Nous ne reviendrons pas sur les définitions, J.C.Fumanal et P.Isoardi ont largement traité la question dans des articles précédents du Bulletin d'Informatique approfondie et applications (FUMANAL 81), (ISOARDI 82). Il est clair qu'aussi bien les graphes de transition, que les automates d'états finis, sont des cas particuliers de réseaux de Pétri.

Ces derniers ont l'avantage de représenter, non seulement le processus lui-même, mais aussi son contrôle: les jetons et la règle de déclenchement. Ils permettent ainsi de modéliser élégamment les problèmes de parallélisme et de synchronisation qui sont au coeur des processus de traitement de l'information et en particulier des systèmes de règles de production (LAURIERE 81).

Les réseaux de Pétri peuvent en effet avoir ici deux intérêts:

-représentation interne du jeu de règles pour l'interpréteur en vue d'optimiser le temps de résolution dans un système d'aide à la conception de thésaurus (BIANCO 85),

exemple: mettre en relation SIGNE DE BASE le concept:  
SIGNES SUBJECTIFS avec le groupe thématique: SYMPTOMES GENERAUX.  
(les mots soulignés appartiennent au langage lui-même, même s'ils sont également "constructibles" dans le langage. Les mots présentés ici en majuscules sont les données à insérer dans le thésaurus) ;

-représentation et utilisation de la connaissance néphrologique par réseaux de Pétri généralisés (KNIPPEL 85)

exemple: réseau de Pétri du contrôle de l'équilibre acido-basique.

### CORPUS

Dans les applications techniques des réseaux d'automates, la bibliographie est bien développée et les équipes françaises actives dans ces sujets. Les deux tomes sur les réseaux de Pétri du collectif d'auteurs G.W.BRAMS en sont le témoignage (BRAMS 83). Toutefois, l'abondance de références n'est peut-être pas faite uniquement pour aider le lecteur.

C'est pourquoi nous proposons la représentation dans une unique structure graphique (celle d'un réseau de Pétri) des relations entre éléments bibliographiques. Cette idée est suggérée dès le début du premier tome par la figure aidant à la lecture de l'ouvrage (voir figure I.I du paragraphe FIGURES). Un problème subsiste à nos yeux, lors du parcours du réseau de Pétri, c'est la non réinitialisation possible du système. Il nous paraît souhaitable de rajouter la transition qui permet de réinitialiser le réseau de la figure I.I. Ce réseau possède donc les propriétés suivantes: borné, sauf, propre et "bien formé" pour  $M_0$  (marquage initial).

Nous rappelons, ici, les trois conditions qui doivent être simultanément vérifiées pour un réseau de Pétri "bien formé":

- le réseau est borné et vivant pour  $M_0$ ,
- le réseau possède une place appelée place de "repos"(22) telle que:

- a) cette place a une seule et une seule transition suivante appelée transition initiale (Introduction générale),

- b) cette place a une seule et une seule transition finale (Conclusion);

- le marquage initial  $M_0$  est le seul marquage de l'ensemble des marquages accessibles à partir de  $M_0$  pour lequel la place de repos possède une marque, de plus la seule transition validée par  $M_0$  est la transition initiale.

L'intérêt de cette propriété est que l'état de repos d'un système représenté par un réseau "bien formé" est caractérisé par une seule place marquée.

### OBJECTIFS

L'étude des deux tomes du G.W.BRAMS nous a permis de faire le point sur l'état de l'art, comme nous le rappelions dans le paragraphe précédent, et de trouver à l'aide de notre méthode d'investigation des points de référence concernant la représentation et l'utilisation de la connaissance néphrologique par réseaux de Pétri généralisés (PETERSON 8I).

Les objectifs majeurs visés ici sont, et nous insistons sur ce point, d'abord la représentation, dans une unique structure hiérarchisée graphique, des relations entre éléments bibliographiques, c'est l'analyse de l'ouvrage. Le deuxième objectif majeur étant, au cours de la lecture d'un nouvel ouvrage traitant du do-

main, de trouver le support d'étude si celui-ci n'est pas explicitement cité dans le titre: arbres des marquages, algèbre linéaire, réductions, extensions ..., ceci nécessite d'organiser l'information dans le langage documentaire.

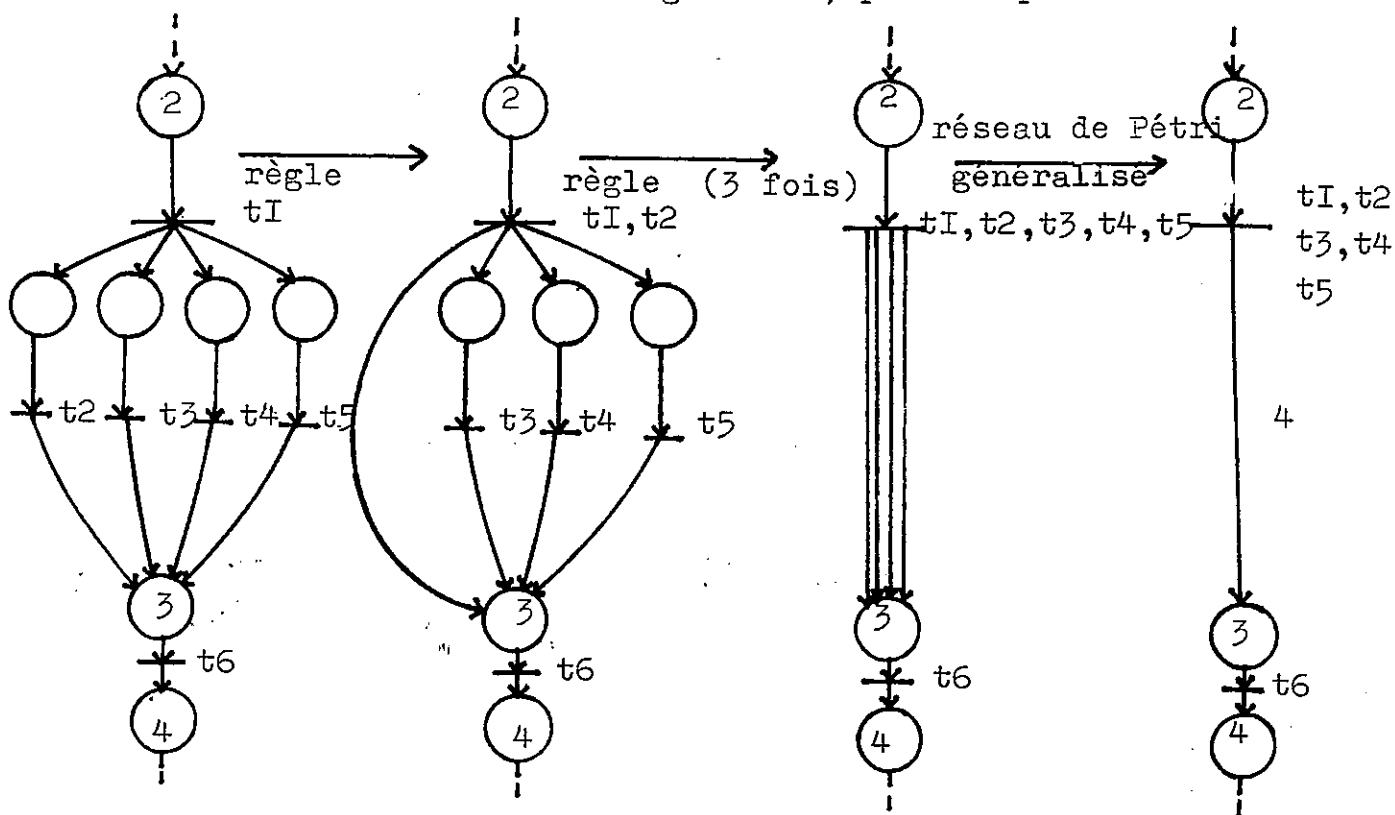
Le paragraphe suivant traite de deux cas récents (ALMHANA 83), (CHARIFI 84), qui permettent d'argumenter le sujet. Il reste néanmoins que cet objectif est "subjectiviste". En effet, rien d'un prime abord permet de savoir si le narrateur est soumis à l'enfer des publications (CHAUVIN 81).

### RESULTATS

Nous présentons les résultats en deux parties: l'une traite de la représentation graphique de l'information bibliographique contenue dans les G.W.BRAMS appelée approche objectiviste, l'autre traite de l'interprétation d'un nouvel ouvrage à la lumière de l'approche objectiviste, appelée approche subjectiviste.

#### Approche objectiviste

La structure choisie pour représenter la bibliographie de chaque chapitre est détaillée ici. Le lecteur trouvera les réseaux de Pétri de chaque concept au paragraphe des FIGURES. Si nous étudions la structure de la figure A.2, par exemple:



Analyse par transformation de la structure.

nous remarquons que nous retrouvons, après l'application d'une règle élémentaire de réduction (substitution d'une chaîne par une transition), la morphologie du réseau de la figure I.I. Néanmoins, il est clair que les propriétés de bornage et de vivacité ne seront conservées que si on substitue à cette chaîne une transition unique (macro-transition). Le caractère propre du réseau n'est conservé dans ce cas que si cette chaîne n'est pas marquée initialement. Le réseau obtenu est, à la différence de la figure I.I, un réseau de Pétri généralisé, où le poids de l'arc représente le nombre d'ouvrages du concept.

Si le réseau vu de la figure I.I est "bien formé", vu de chaque "sous-structure" ou concept, il perd cette propriété. En effet, il se peut que des jetons s'accumulent dans des places. Ceci est dû au fait qu'un article n'est pas lu par l'utilisateur. Cette remarque apporte, nous semble-t-il, des informations pertinentes.

#### Approché subjectiviste

Pourquoi deux thèses de Doctorat es Sciences (ALMHANA 83), (CHARIFI 84), et pas d'autres documents sont ils étudiés? Ces travaux ne contiennent pas, dans leur titre, une référence aux transitions de la figure I.I, et n'ont pas bénéficié des G.W.BRAMS.

Nous avons donc la possibilité de dénombrer simplement le nombre de fois, où ces transitions (mots-clés ou synonymes) sont citées; ce qui constitue un premier critère subjectiviste. Un deuxième critère est d'étudier par "chapitre" le nombre de références bibliographiques aux G.W.BRAMS; ceci permettant d'insérer, peut être, ce nouvel élément dans un ou plusieurs des réseaux construits. Ce travail permet une mise à jour éventuelle des "sous-structures".

Le premier critère nous a permis de mettre en évidence l'emploi de la modélisation par les réseaux de Pétri dans l'étude de M.Charifi. L'auteur d'un "système expert et de conception assistée par ordinateur de systèmes technologiques" ne s'est intéressé qu'exclusivement aux abréviations: à savoir l'étude de réseaux de taille importante. Quant à la "modélisation par réseaux de Pétri à flux de données. Application à la synthèse de l'opérateur de Riccati rapide" de J.Almhana, elle nous a permis de confirmer son intérêt pour la modélisation et les applications des réseaux de Pétri. L'auteur, dans l'ensemble de sa bibliographie, s'est intéressé

outre à l'introduction générale, aux réseaux non autonomes et aux réalisations matérielles. Ce chemin est très simple à suivre sur la figure I.I.

Le deuxième critère subjectiviste nous indique, vu l'exclusivité des références bibliographiques "aux abréviations", d'insérer comme ouvrage pertinent de ce mot-clef, la thèse de M.Charifi. La thèse de J.Almhana par la quantité de références aux domaines des réseaux non autonomes et des réalisations matérielles, sur un problème d'opérateur rapide, prendrait sa place dans les réalisations matérielles et réseaux non autonomes.

### CONCLUSION

L'étude de l'ouvrage "Réseaux de Pétri: théorie et pratique" permet d'acquérir un savoir important du domaine. Les démonstrations complètes donnent une bonne compréhension des méthodes d'analyse des propriétés des réseaux. Les deux tomes abordent également les outils de mise en oeuvre ainsi que les réalisations matérielles.

Dans cet article, nous avons tenu à privilégier la représentation et le traitement des connaissances bibliographiques dans une unique structure graphique: le réseau de Pétri. Les relations entre éléments bibliographiques mettent en lumière une approche objectiviste et subjectiviste de lecture des références. La structure graphique proposée, par son non bornage en particulier, peut mettre en évidence la non lecture de certaines parties.

Ce travail se prolonge par d'autres points qui ne sont pas traités ici. Citons l'étude des transitions ou relations à prendre en compte, qui rejoignent l'aide aux thésaurus:

-voisinage ou association (auteurs multiples):

exemple: R.Karp  
          .R.Miller  
          R.Miller  
          .R.Karp

-hiérarchie (linguistique):

exemple: langue d'édition  
          /          |          \  
allemand  anglais  français  russe

-synonymie ou équivalence (publications communes à plusieurs thèmes):

exemple: R.Karp et R.Miller "Parallel program schemata" : thèmes  
présentation des propriétés et arbres des marquages.

De plus, nous développerons dans un prochain article la  
représentation des connaissances sur le plan sémantique, dans le  
cadre thématique choisi, à l'aide des informations que nous avons  
acquises lors de ce travail. Les objets manipulés: livres, articles,  
revues, bulletins nous ont servi pour acquérir le fonds technique  
complet sur les réseaux de Pétri, outil possible de manipulation  
sémantique du thésaurus. Le cas d'accès "simultané" au même thésau-  
rus sera envisagé par l'introduction des réseaux de Pétri colorés.

### FIGURES

- figure I.I Introduction générale
- figure A.I Généralités
- figure A.2 Présentation des propriétés
- figure A.3 Arborescence et graphe de couverture
- figure A.4 Algèbre linéaire
- figure A.5 Réductions
- figure A.6 Classes de réseaux de Pétri
- figure M<sub>o</sub>.7 Abréviations des réseaux
- figure M<sub>o</sub>.8 Extension des réseaux de Pétri
- figure M<sub>o</sub>.9 Réseaux non autonomes
- figure M<sub>o</sub>.10 Exemples
- figure M<sub>o</sub>.11 Réalisations matérielles
- figure M<sub>o</sub>.12 Domaines d'applications
- figure C.I3 Conclusion

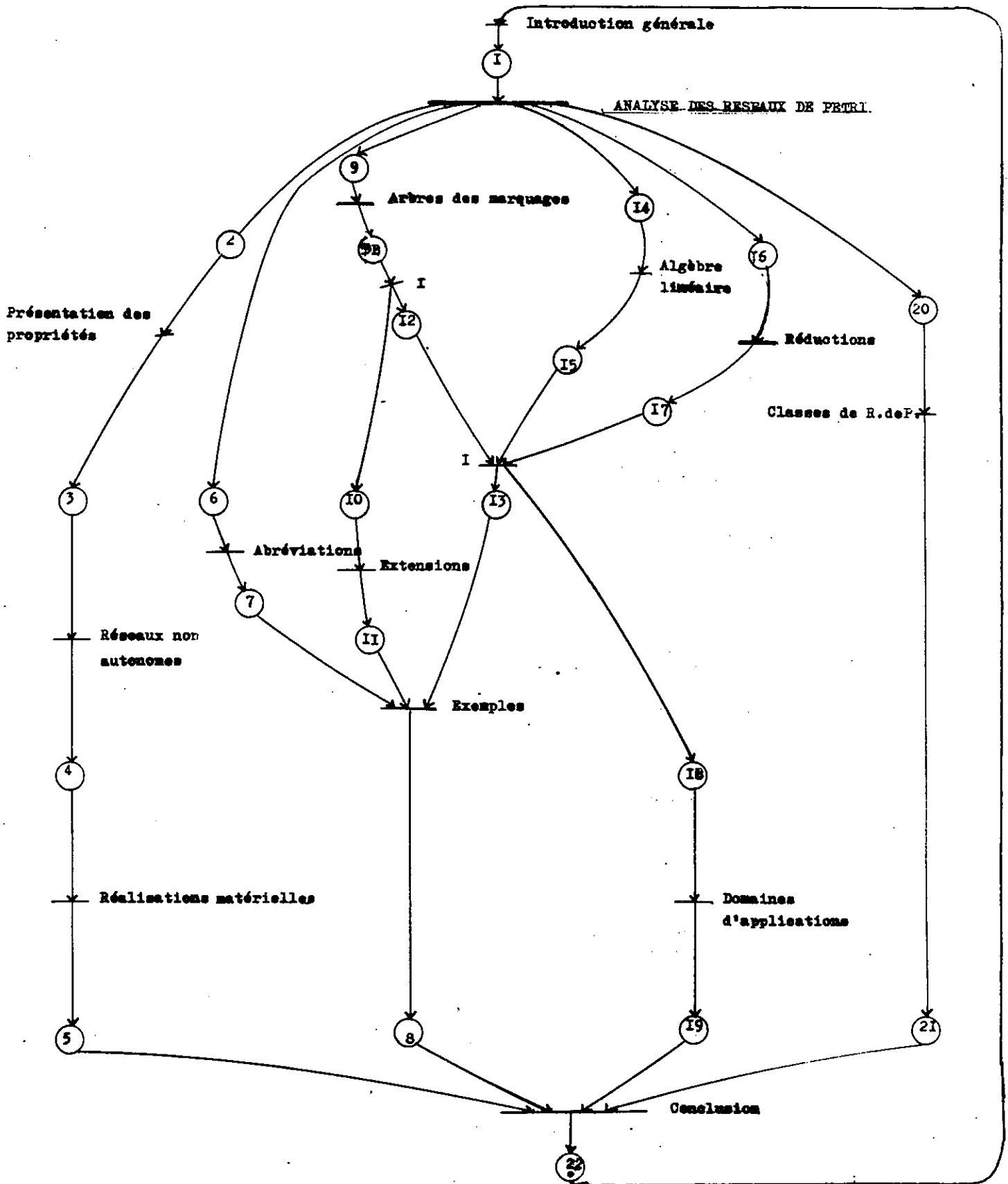


figure I.I Introduction générale



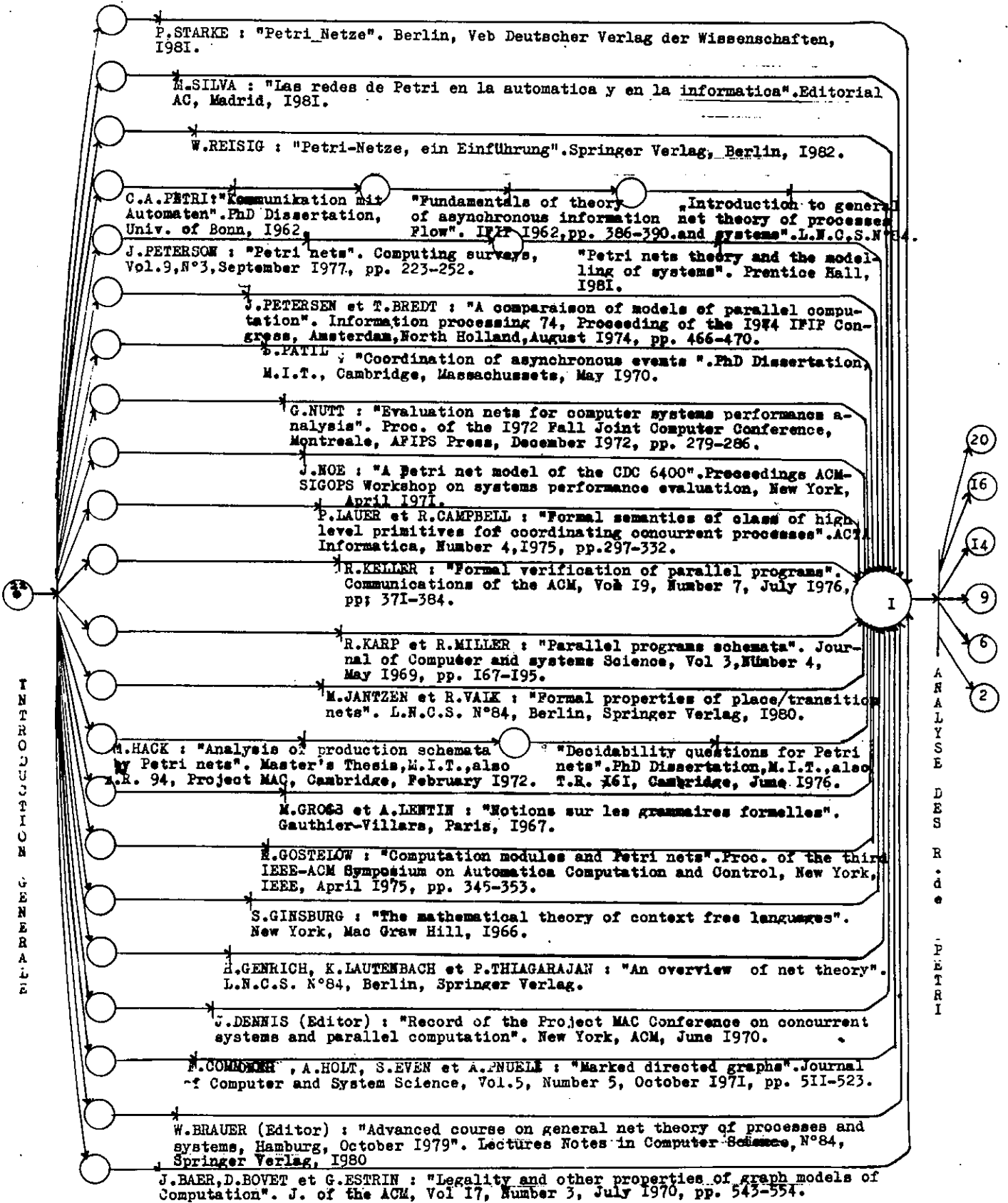


figure A.I Généralités

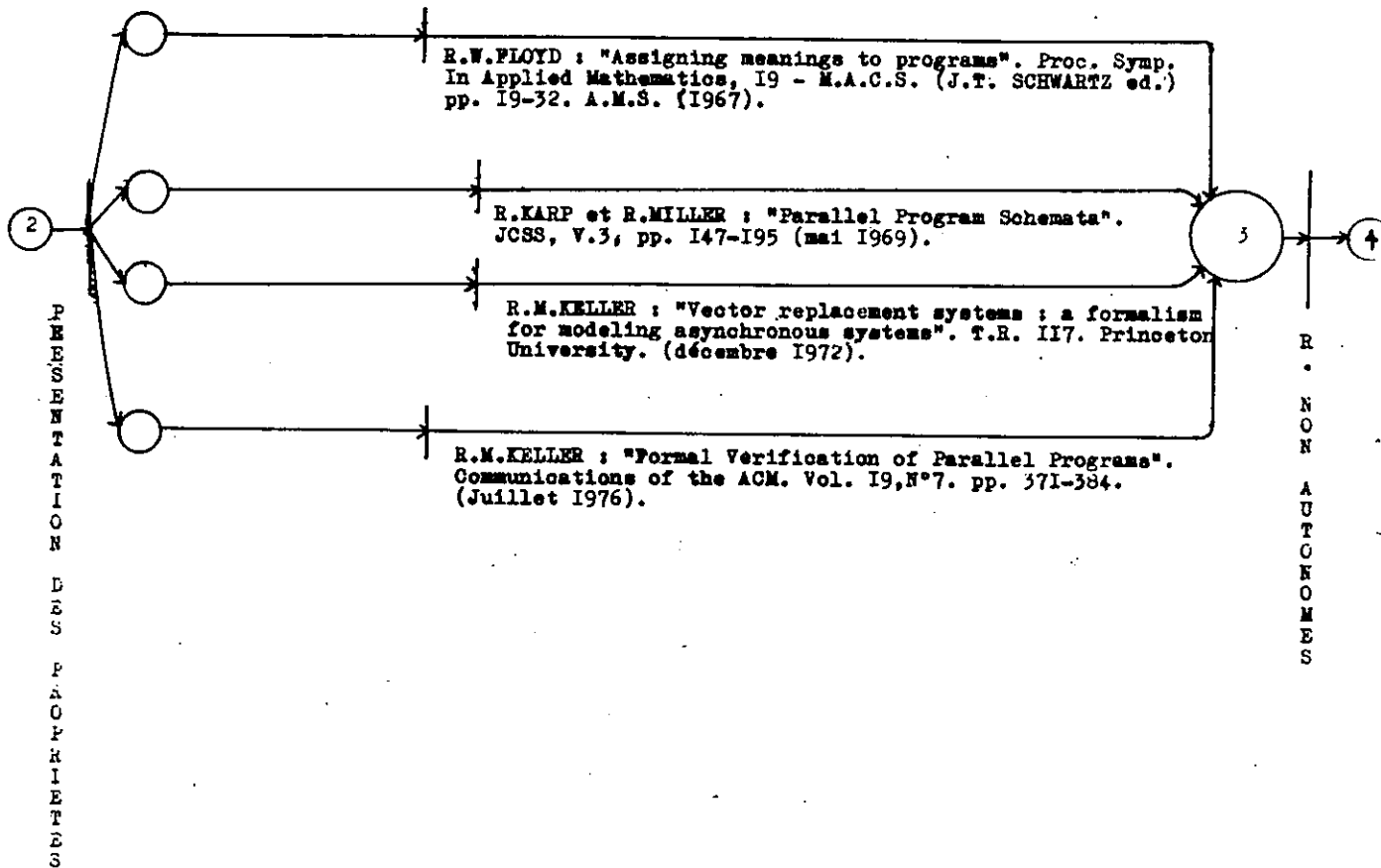


figure 1.2 Présentation des propriétés

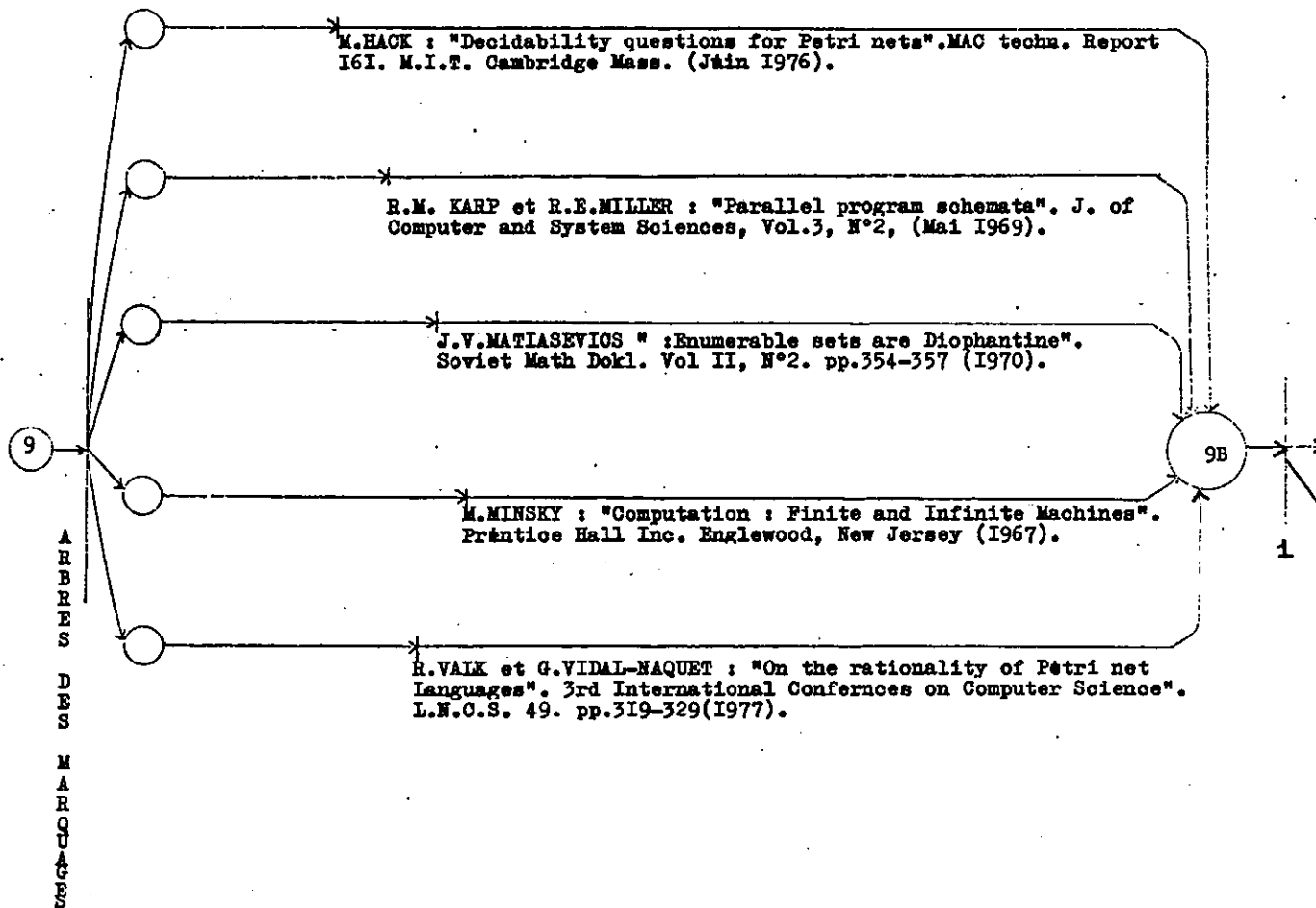


figure A.3 Arborecence et graphe de couverture

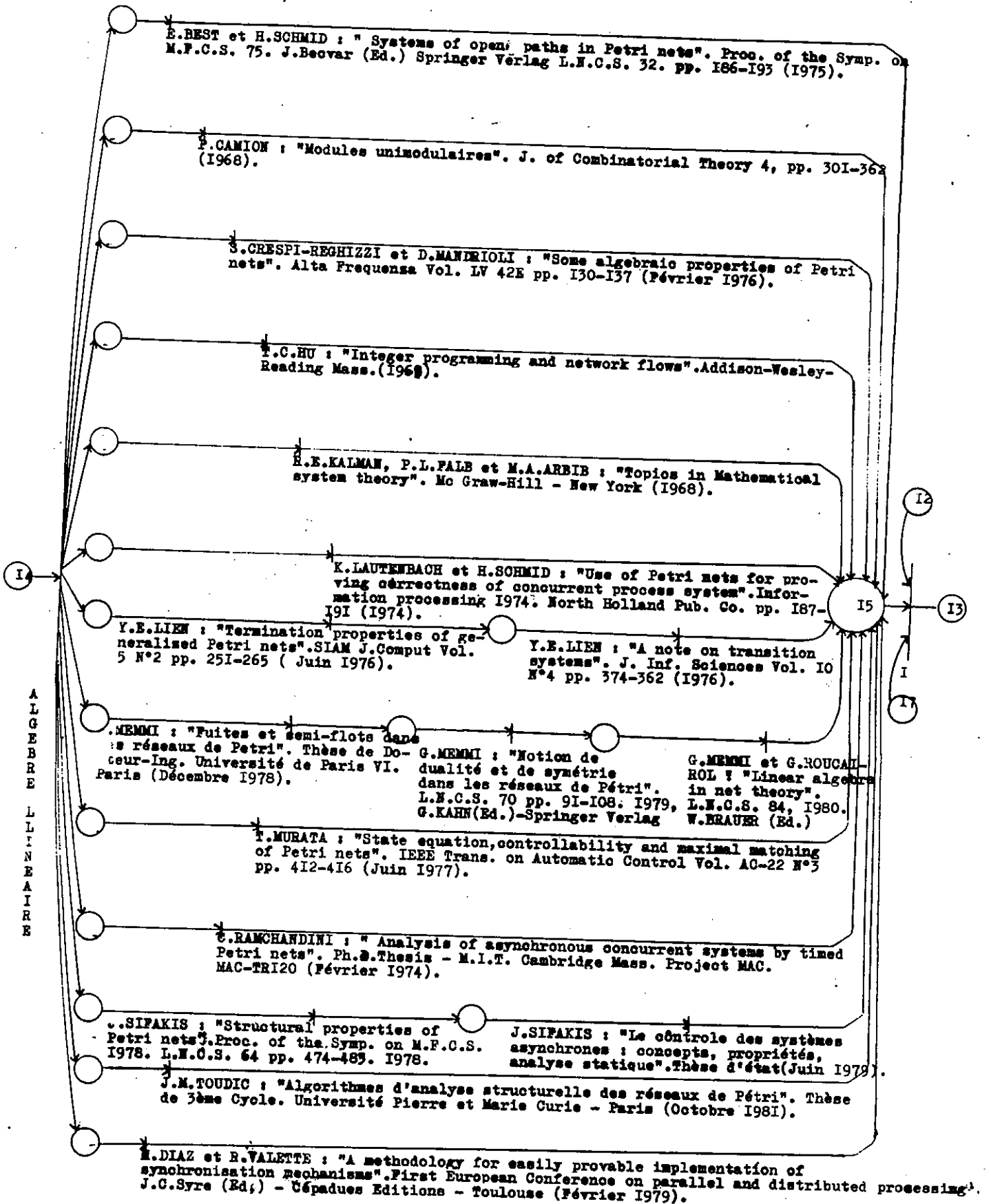


figure A.4 Algèbre linéaire

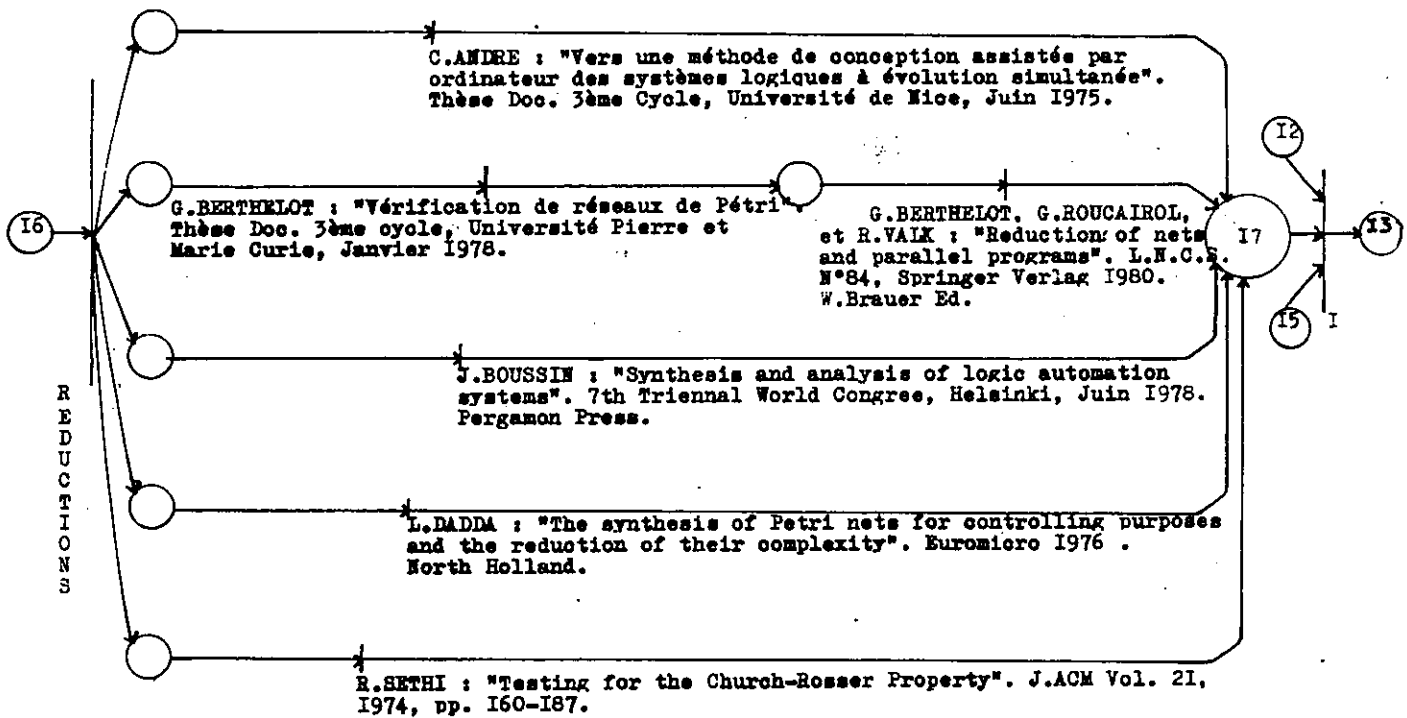


figure A.5 Réductions

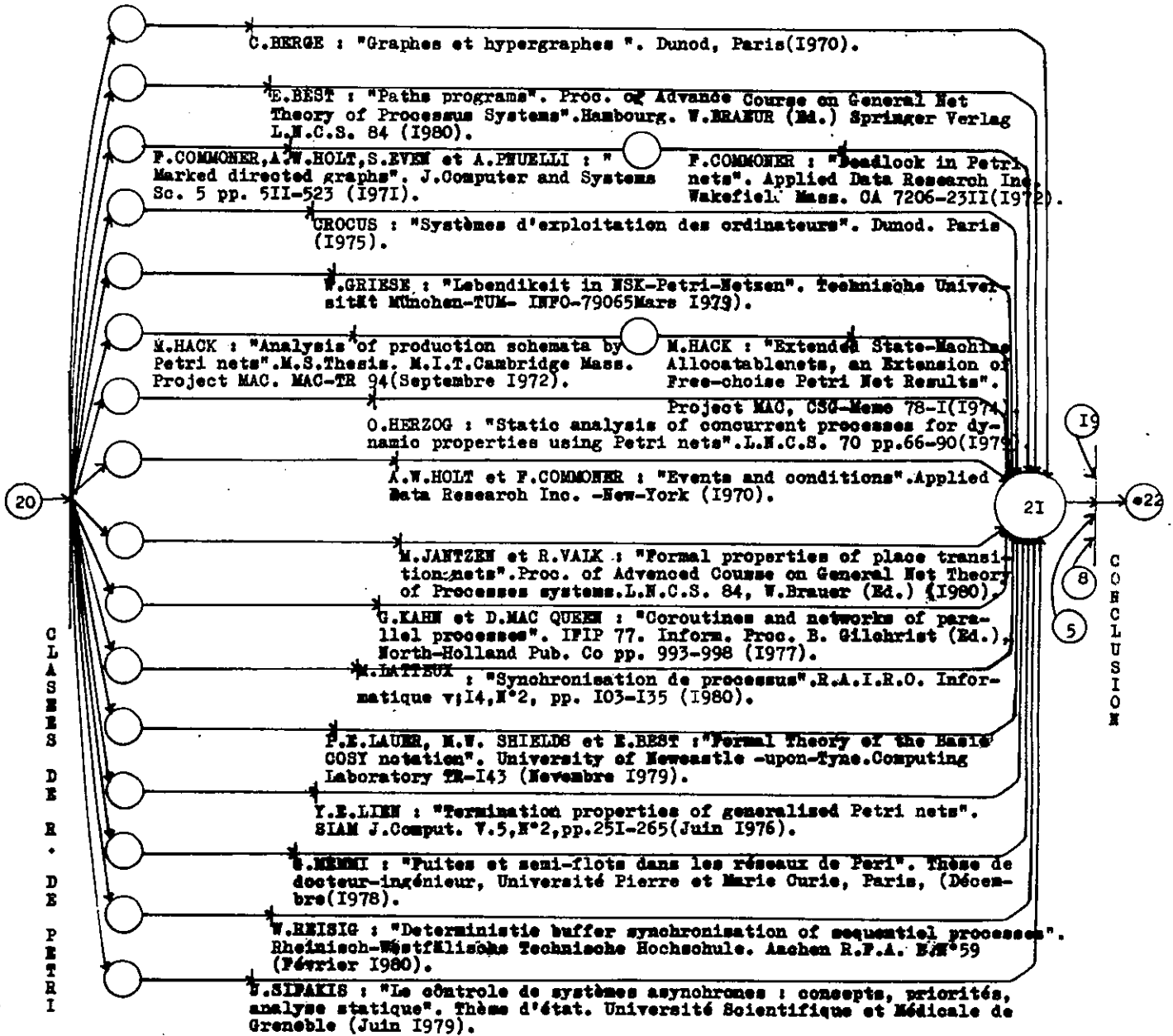


figure A.6 Classes de réseaux de Pétri

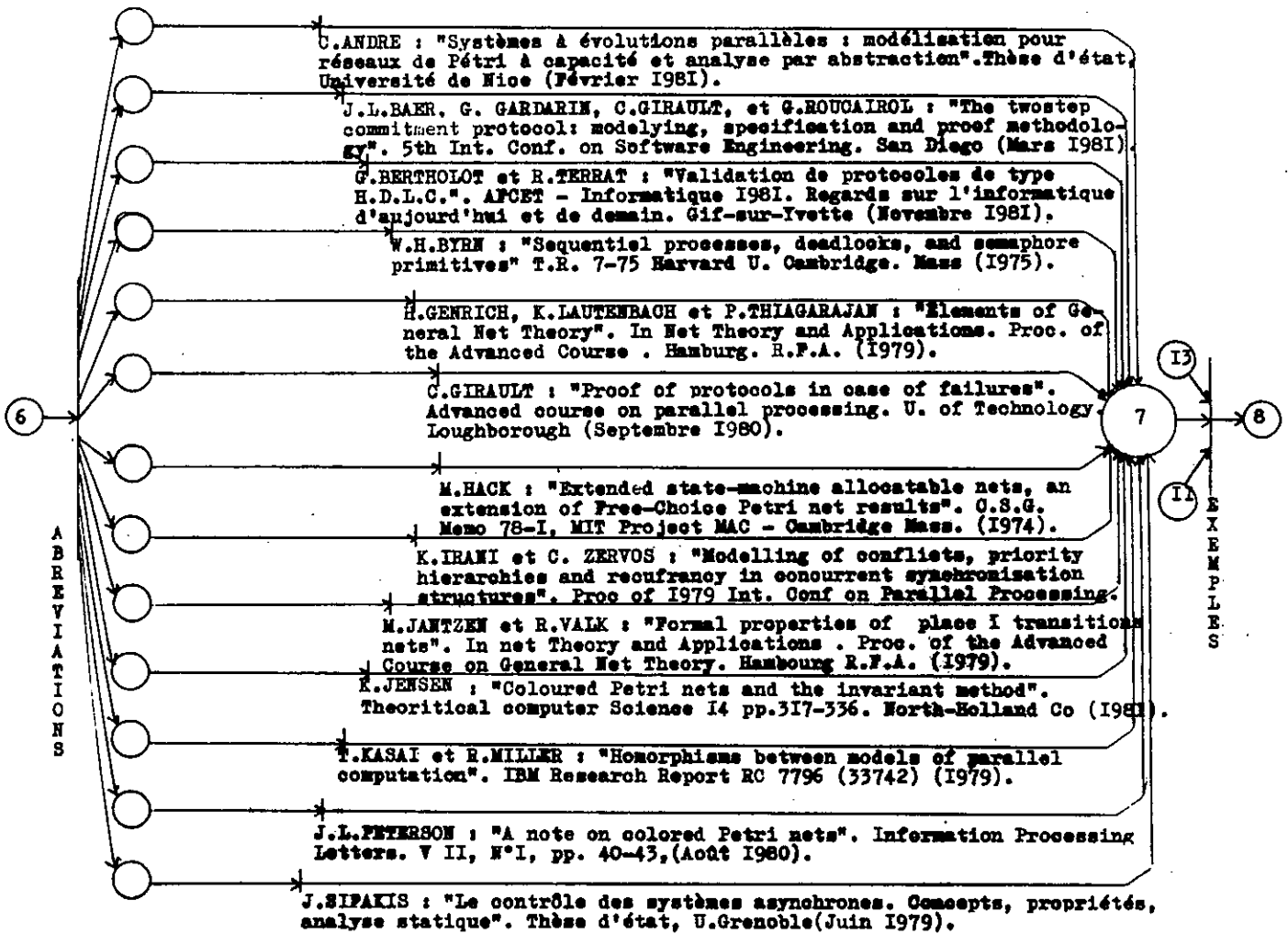


figure M.7 Abréviation des réseaux

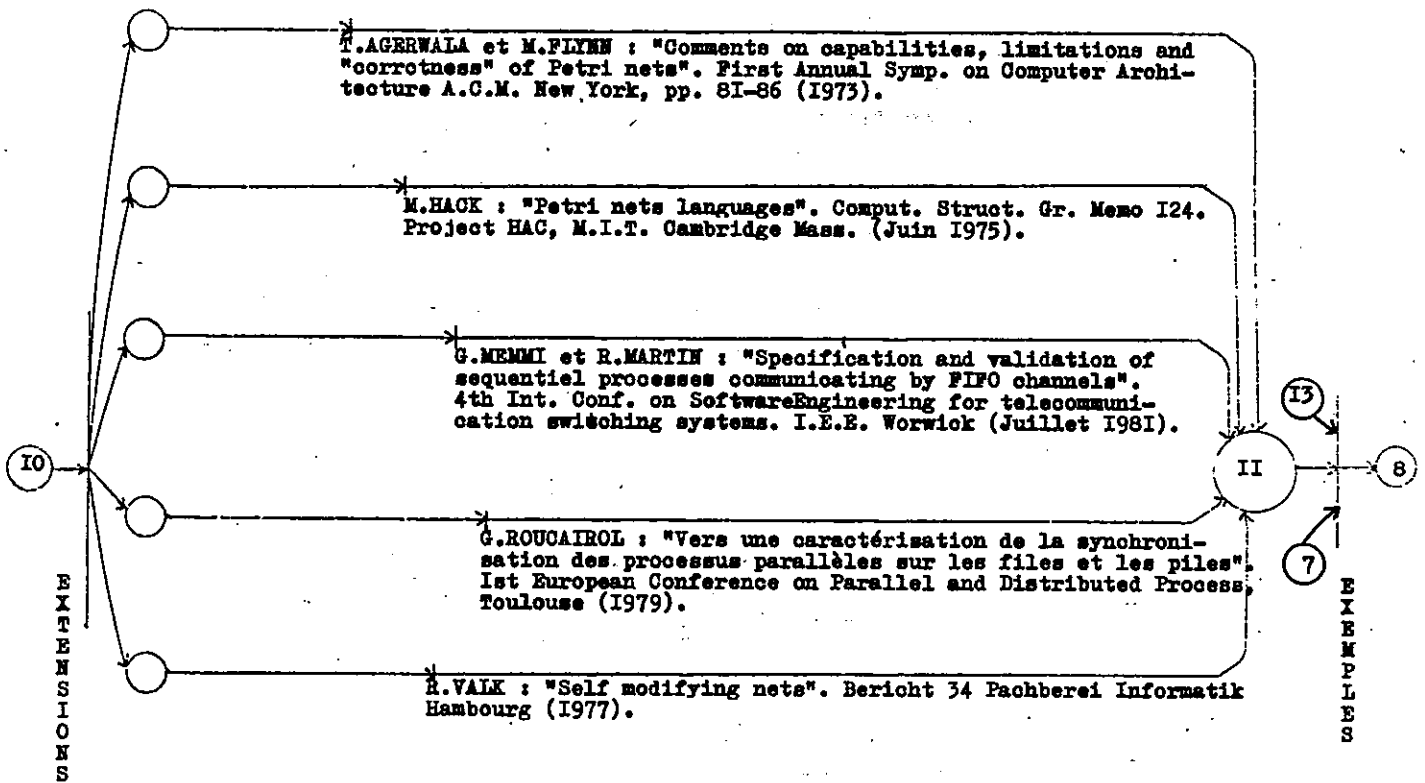


figure M<sub>0</sub>.8 Extensions des réseaux de Pétri



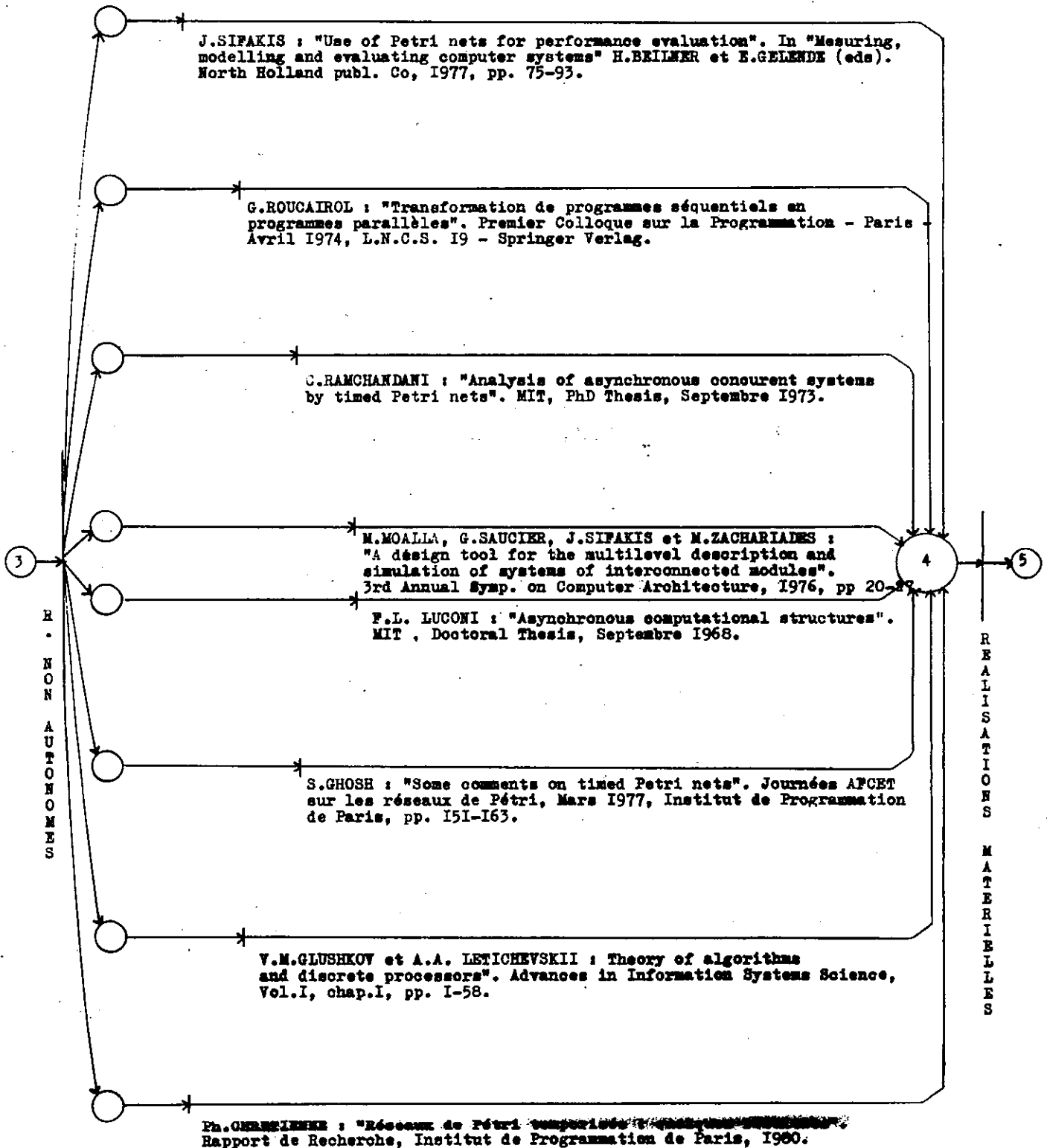


figure M<sub>0</sub>.9 Réseaux non autonomes

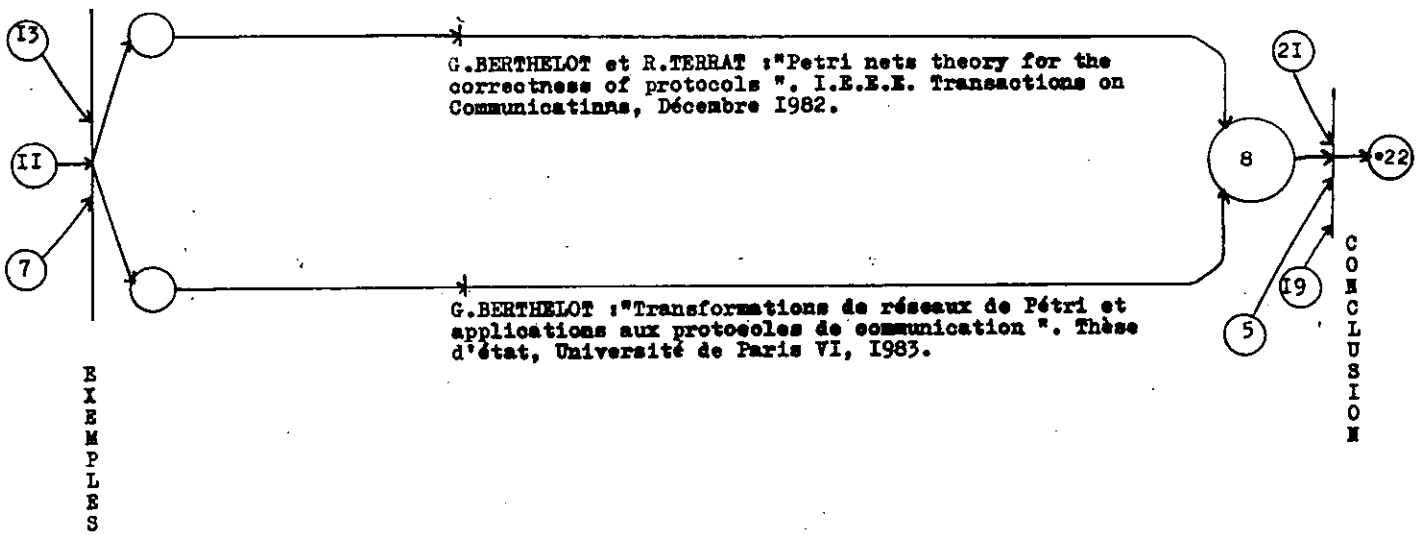


figure M.10 Exemples

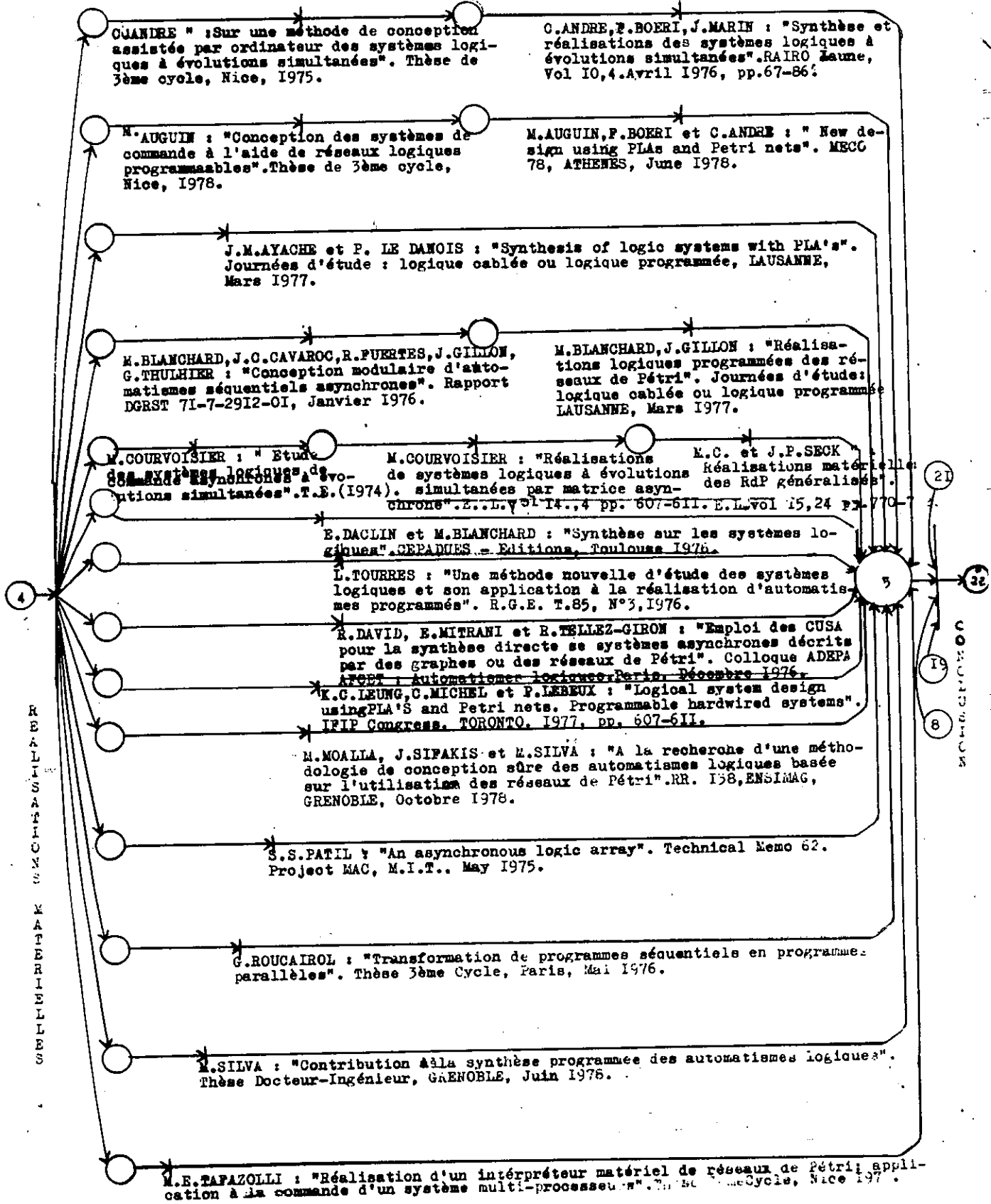


figure No. II Réalisations matérielles

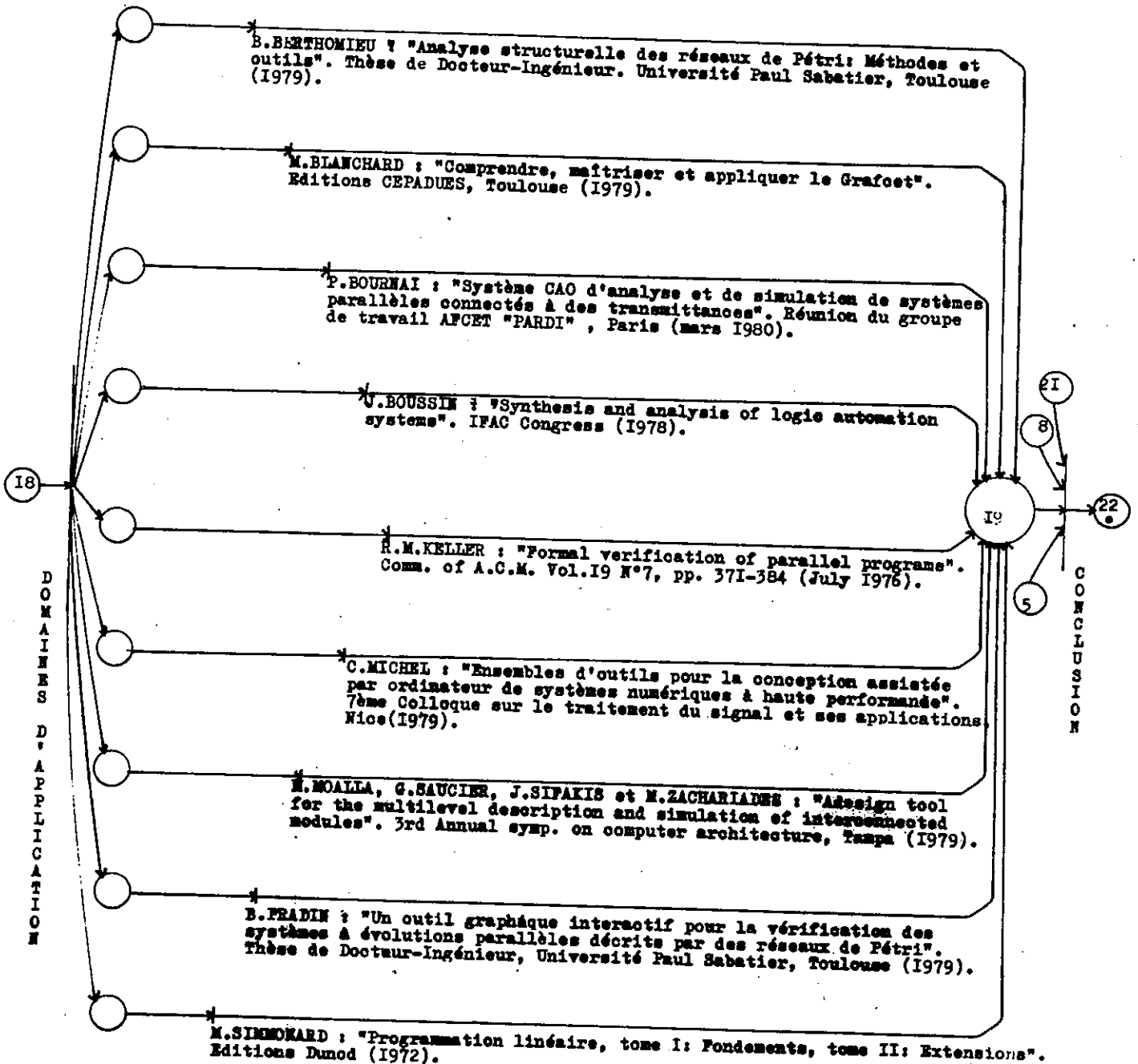


figure M<sub>0</sub>.I2 Domaines d'applications

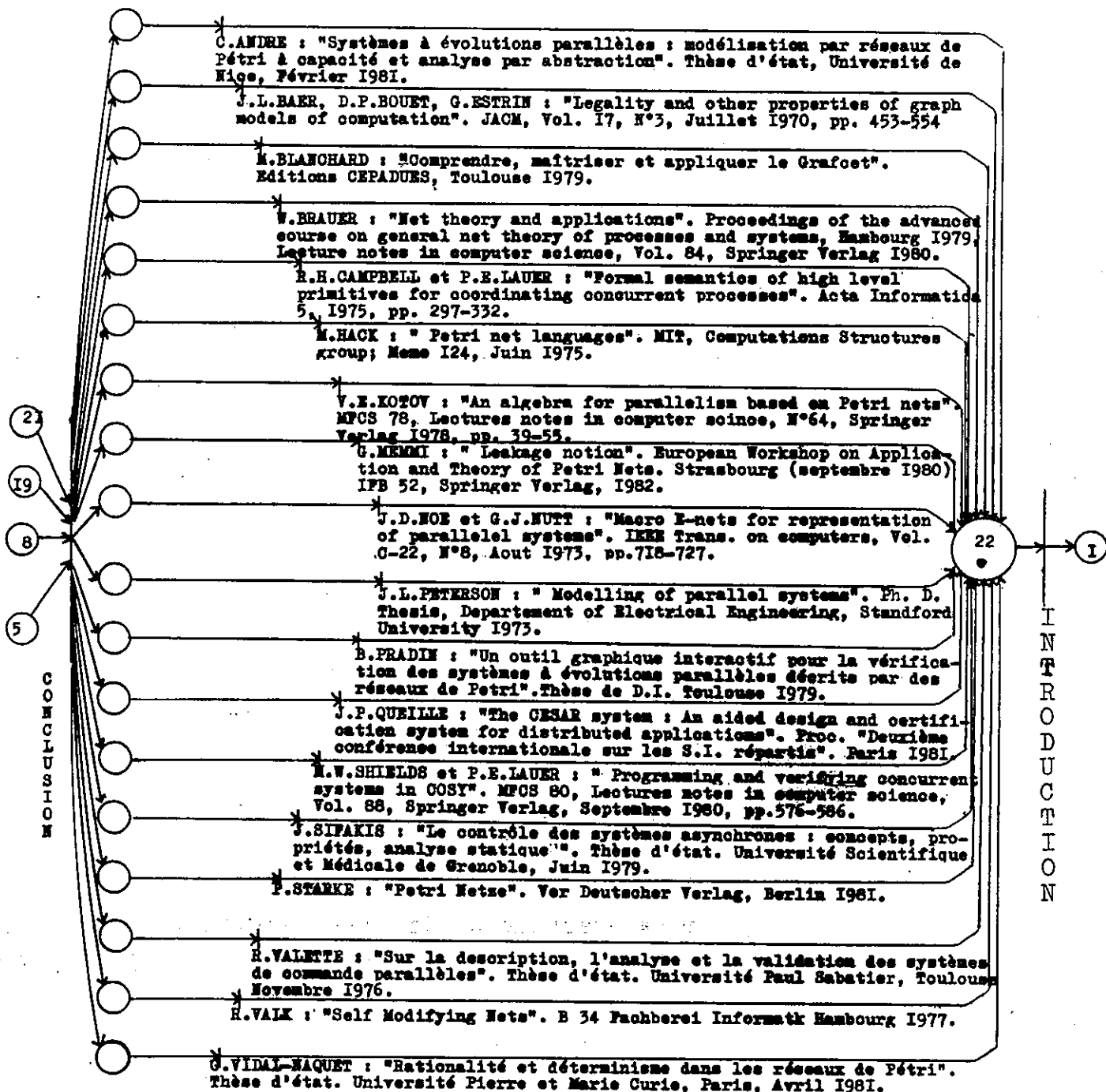


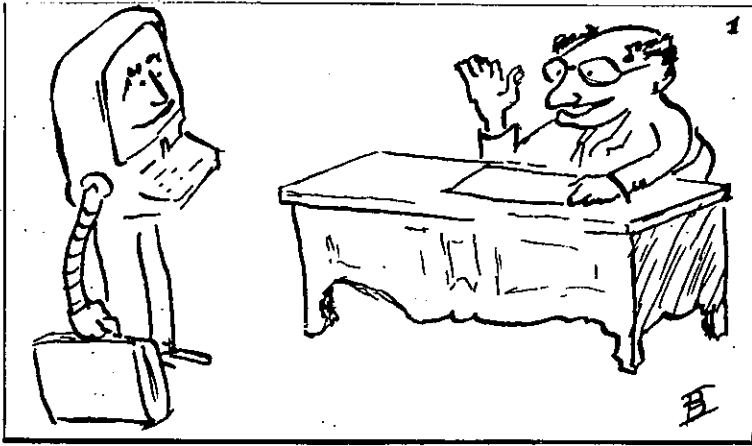
figure C.I3 Conclusion

REFERENCES

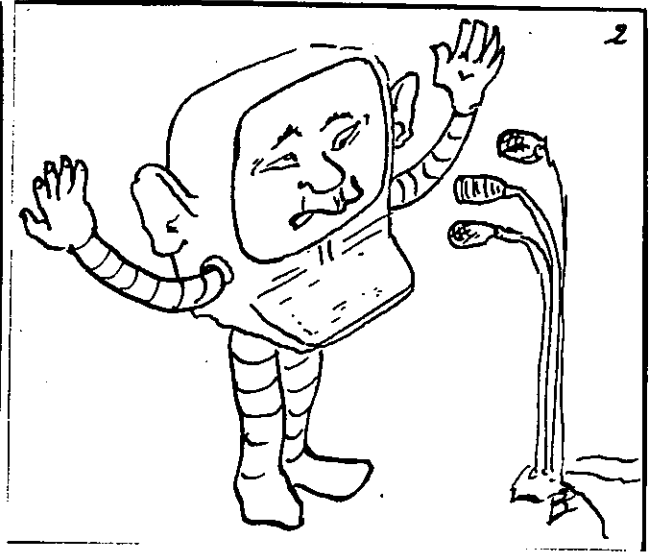
- ALMHANA 83 J.ALMHANA. Modélisation par réseaux de Pétri à flux de données. Application à la synthèse de l'opérateur de Riccati rapide. Thèse pour obtenir le grade de Docteur es Sciences. (Université de Droit, d'Economie et des Sciences d'Aix-Marseille - 1983)
- BIANCO 85 E.BIANCO, J.Ph. LEHMANN. Groupe de formation doctorale: informatique fondamentale et sciences de la computation. (L.I.T.A.M. Faculté de Luminy. 1985)
- BRAMS 83 G.W.BRAMS. Réseaux de Pétri. Théorie et Pratique. Tomes I et II. (MASSON - 1983)
- BROUSSALIAN 83 G.BROUSSALIAN. Répertoire de Kent. (Imprimerie Léostic - Seyssinet - 1983)
- CHARIFI 84 M.CHARIFI. Système expert et de conception assistée par ordinateur de systèmes technologiques. Thèse pour obtenir le grade de Docteur es Sciences. (Université de Droit, d'Economie et des Sciences d'Aix-Marseille - 1984)
- CHAUVIN 81 R.CHAUVIN. Des savants, pourquoi faire? (Editions Payot - 1981)
- FUMANAL 81 J.C.FUMANAL. Automate programmable. Interpréteur de réseaux de Pétri. Utilisation et exemples. (Bulletin d'Informatique approfondie et applications. L.I.T.A.M. N°1. N°2. N°3. Faculté de Luminy. Marseille - 1981 - 1982)
- ISOARDI 82 P.ISOARDI. Le GRAFCET. Etudes et réflexions. (Bulletin d'Informatique approfondie et applications. L.I.T.A.M. N°5. Faculté de Luminy. Marseille - 1982)
- KNIPPEL 82 J.M.KNIPPEL. Vers une modélisation du système homme-machine de dialyse: application de méthodes d'analyse des données. Congrès IRIES. (Edited by IRIES. Beynes - 1982 )
- KNIPPEL 84 J.M.KNIPPEL. Approche du système homme-machine de dialyse. Existence de classes du sous-modèle humain initial: les techniques de classification. Congrès WOGSC-AFCET. (Edited by AFCET. Paris - 1984)

- KNIPPEL 85 J.M.KNIPPEL. Représentation des connaissances néphrologiques: systèmes de production et réseaux de Pétri. Ecole internationale d'informatique de l'AF CET. Baume lès Aix. (AF CET - I986)
- LASKRI et GALAND 85 M.T.LASKRI, Ph.GALAND. Analyse de première approche du langage naturel dans le cadre de l'étude d'un système support de thésaurus. (Bulletin d'Informatique approfondie et applications. L.I.T.A.M. N°II. Faculté de Luminy. Marseille - I985)
- LAURIERE 8I J.L.LAURIERE. Représentation des connaissances. (T.S.I. N°2. DUNOD - I98I)
- PETERSON 8I J.L.PETERSON. Petri net theory and the modeling of systems. (Prentice Hall, inc - I98I)

# L'ORDINATEUR FACTOTUM



- Vous m'achèverez la comptabilité 1997... ,  
 Vous me ferez une provision de stock de précom-  
 puter 12X7 ... vous me visualiserez le match  
 Milan - San Remo...

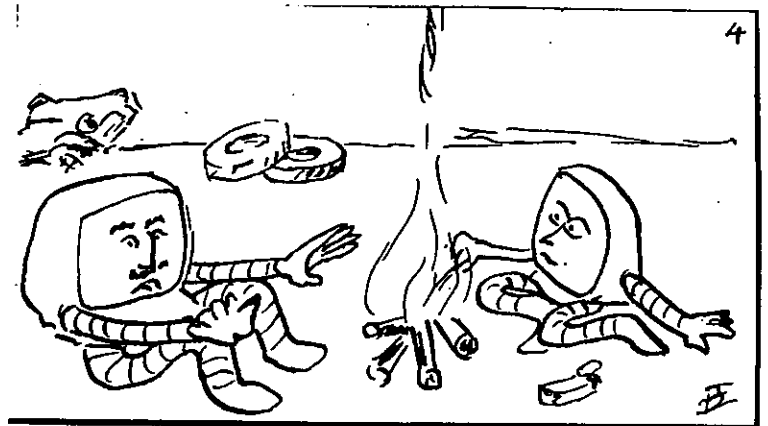


Le Meilleur Economiste s'adresse  
 aux Français- Français :  
 " Je refuse la cohabitation avec ces  
 humains ...

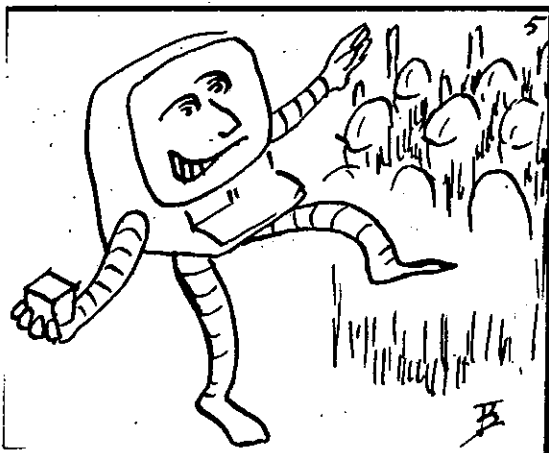


Bref : L'ORDINATEUR-ROI.

⇒ Oui ! mais aussi peut-être :



L'ordinateur au chômage ...



L'ordinateur lanceur de pavés...



L'ordinateur-cendrier (modèle de luxe).