

وزارة التعليم و البحث العلمي

MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

2ex

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT d'Electronique

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

ETUDE ET REALISATION D'UN
MICRO-SYSTEME EXPERT
D'AIDE AU DIAGNOSTIC
DES PANNES DE T.V

Proposé par :

K. KASDI

Etudié par :

A. SAHED

B. SEKHRI

Dirigé par :

K. KASDI

PROMOTION : Juin 1986

وزارة التعليم والبحث العلمي

MINISTÈRE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT d'Electronique

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

ETUDE ET REALISATION D'UN
MICRO-SYSTEME EXPERT
D'AIDE AU DIAGNOSTIC
DES PANNES DE T.V

Proposé par :

Etudié par :

Dirigé par :

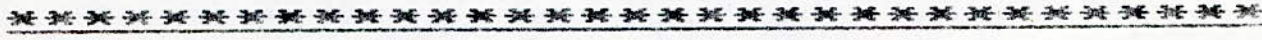
PROMOTION :

D E D I C A C E S

A NOS PARENTS
ET
A NOS AMIS.....



REMERCIEMENTS



*
*
*
* Nous tenons à exprimer nos plus vifs remerciements à
* notre promoteur Mile K.KASDI pour sa gentillesse et pour
* l'aide précieuse qu'elle nous a prodiguée durant tout
* notre travail , sans oublier Monsieur F.BRIKCI qui nous
* a suivis et conseillés durant toute la première partie
* de notre travail .
*
* Toute notre reconnaissance à l'ensemble des
* enseignants ayant contribué à notre formation .
*
* Que tous ceux qui, de près ou de loin , ont participé
* à la réalisation de ce projet voient ici l'expression
* de notre gratitude .
*
* Nous tenons à remercier les membres du jury qui par
* leur présence et leurs jugements , vont honorer ce
* modeste travail .
*
*
*

S O M M A I R E

- <u>INTRODUCTION</u>	5
- <u>CHAPITRE 1 : LES SYSTEMES EXPERTS</u>	6
I.1 BUT.....	7
I.2 DEFINITION FONCTIONNELLE.....	7
I.3 DOMAINES D'APPLICATION.....	7
I.4 ORGANISATION DE PRINCIPE.....	8
I.5 LE MOTEUR D'INFERENCE.....	9
I.6 FORMULATION DES REGLES.....	9
I.7 FONCTIONNEMENT DU MOTEUR D'INFERENCE.....	9
I.8 MODE D'INVOCATION DES REGLES.....	12
I.9 LES MODES DE REPRESENTATIONS.....	13
I.10 LES STRATEGIES DE DEVELOPPEMENT.....	14
I.11 LA MONOTONIE.....	15
- <u>CHAPITRE 2 : LES STRUCTURES DE DONNEES</u>	16
II.1 INTRODUCTION.....	17
II.2 METHODES DE STRUCTURATION.....	17
2.1 LES VARIABLES DYNAMIQUES	17
2.2 LES STRUCTURES DE BASE...	18
II.3 OPERATIONS SUR UNE LISTE.....	19
MONODIRECTIONNELLE.....	

S O M M A I R E (S U I T E)

- CHAPITRE 3 : REALISATION DU MOTEUR D'INFERENCES		
III.1	INTRODUCTION.....	NNN
III.2	CARACTERISTIQUES DU M.I.....	NNN
III.3	POSITION DU PROBLEME.....	NNN
III.4	PRINCIPALES PARTIES.....	NNN
	CONSTITUANTS LE M.I.....	NNN
III.5	LES FICHIERS UTILISES.....	NNN
III.6	LE SUPERVISEUR.....	NNN
III.7	LE MODULE DE DIALOGUE.....	NNN
III.8	LE MODULE D'APPRENTISSAGE.....	NNN
III.9	LE MODULE DE RAISONNEMENT.....	NNN
- CHAPITRE 4 : ETUDE DES PANNES TV		
IV.1	INTRODUCTION.....	5
IV.2	SCHEMA SYNOPTIQUE ET FONCTIONS DES DIFFERENTS BLOCS.....	5
IV.3	ETUDE DES PANNES ENGENDREES... PAR CHAQUE BLOC.....	5
- CHAPITRE 5 : REDACTION ET IMPLEMENTATION D'UNE BASE DE CONNAISSANCES RELATIVE AUX PANNES DE TELEVISION		
V.1	METHODE	7
V.2	EXEMPLE D'EXECUTION	7
- ANNEXES		
	. LES SEQUENCES ESCAPE (ANSI)...	7
	. GLOSSAIRE.....	7
	. CONCLUSION.....	8

INTRODUCTION

L'intelligence a constitué la base des travaux de nombreux chercheurs depuis des siècles (Psychologues, Philosophes Théologiens, Neurophysiciens). Il est très difficile d'être précis sur le mot "intelligence". La réponse traditionnelle donnée par les Psychologues "l'intelligence est ce que mesurent les tests d'intelligence" est totalement dépourvue d'intérêt. Il est plus simple d'étudier les mécanismes permettant un comportement intelligent. C'est ce que proposent d'étudier les nouvelles disciplines telles que : LA CYBERNETIQUE et L'INTELLIGENCE ARTIFICIELLE.

L'intelligence artificielle notée (I.A) a pour sujet l'étude des activités de l'homme pour lesquelles aucune méthode standard n'est a priori connue. Ainsi si l'informatique classique est la science du traitement de l'information, l'I.A, nouvelle branche de cette science, s'intéresse à des cas où ce traitement ne peut être ramené à une méthode simple algorithmique et à des problèmes dont la combinatoire est tellement vaste que la solution optimale, lorsqu'elle existe, ne peut être atteinte à cause des contraintes de temps et d'espace mémoire.

Il n'est pas facile de répertorier tous les domaines concernés par l'I.A, tant il existe d'imbrications et d'interactions. Pour simplifier, on peut distinguer les domaines suivants : traitement de l'image, synthèse vocale, reconnaissance vocale, pensée, raisonnement, résolution automatique de problèmes, compréhension du langage et traduction, etc...

L'application la plus importante de l'I.A est le système expert. Un système expert est un ensemble de logiciels exploitant des connaissances explicites à un domaine particulier, pour offrir un comportement comparable à celui de l'expert humain.

En effet, les connaissances humaines croissent exponentiellement sur un nombre incalculable de sujets: Médecine, Physique, Chimie, Astronomie, Technologie..., et seuls les experts humains très pointus dans leur domaine possèdent et manipulent ces connaissances. De plus la formation de ces experts est très longue par rapport à leur durée de vie.

Le but de notre travail est de réaliser un noyau de système expert d'aide au diagnostic des pannes de télévision. Cette réalisation se compose de deux parties essentielles :

- Développement d'un programme "MOTEUR D'INFERENCE" écrit en langage Pascal.
- Rédaction et implémentation d'une base de connaissances expertes sur les pannes de télévision.

***** CHAPITRE 1 *****

**
** LES.....SYSTEMES **
.....
.....EXPERTS.....

I.1	BUT.....	7
I.2	DEFINITION FONCTIONNELLE.....	7
I.3	DOMAINES D'APPLICATION.....	7
I.4	ORGANISATION DE PRINCIPE.....	8
I.5	LE MOTEUR D'INFERENCE.....	9
I.6	FORMULATION DES REGLES.....	9
I.7	FONCTIONNEMENT DU MOTEUR D'INFERENCE	9
7.1	CYCLE DE BASE D'UN MOTEUR D'INFERENCE.....	9
7.2	PHASE D'EVALUATION.....	10
7.3	PHASE D'EXECUTION.....	10
7.4	REGIME IRREVOCABLE ET PAR TENTATIVES.....	10
7.5	REPRESENTATION D'UN CYCLE DE BASE D'UN MOTEUR.....	11
7.6	MODELES DE COMPATIBILITE.,	11
7.7	EFFETS DES DECLENCHEMENTS DES REGLES.....	12
I.8	MODE D'INVOCATION DES REGLES.....	12
I.9	LES MODES DE REPRESENTATIONS.....	13
I.10	LES STRATEGIES DE DEVELOPPEMENT.....	14
I.11	LA MONOTONIE.....	15

LES SYSTEMES EXPERTS

I-1 BUT

Le but d'un système expert est de modéliser le comportement d'un expert humain accomplissant une tâche intellectuelle dans un domaine précis.

I-2 DEFINITION FONCTIONNELLE

Les systèmes experts sont des logiciels destinés à remplacer ou assister l'homme dans un domaine où est reconnue une expertise humaine :

- insuffisamment structurée pour constituer une méthode de travail précise, sûre, complète, directement transportable sur ordinateur.
- sujette à révisions ou compléments (selon l'expérience accumulée).

I-3 DOMAINES D'APPLICATION

Les systèmes experts apparaissent particulièrement adaptés pour la résolution de certains types de problèmes où :

- l'on dispose d'une grande quantité de connaissances incomplètes ou incertaines de type plutôt heuristique qu'algorithmique sujette à révision.
- la stratégie de résolution est dynamique et intelligente et dépend du problème et de son contexte .

Un système expert doit donc :

- fournir une possibilité de dialogue homme-machine aisé.
- donner (souvent) à l'utilisateur "le chemin qu'il a suivi pour aboutir au résultat".

Grace à ces deux propriétés les systèmes experts se trouvent particulièrement bien adaptés aux domaines suivants :

-L'ENSEIGNEMENT ASSISTE PAR ORDINATEUR_____ c'est un domaine de choix grace aux caractéristiques de dialogue et de comportement intelligent du système expert

-RESOLUTION DES PROBLEMES_____

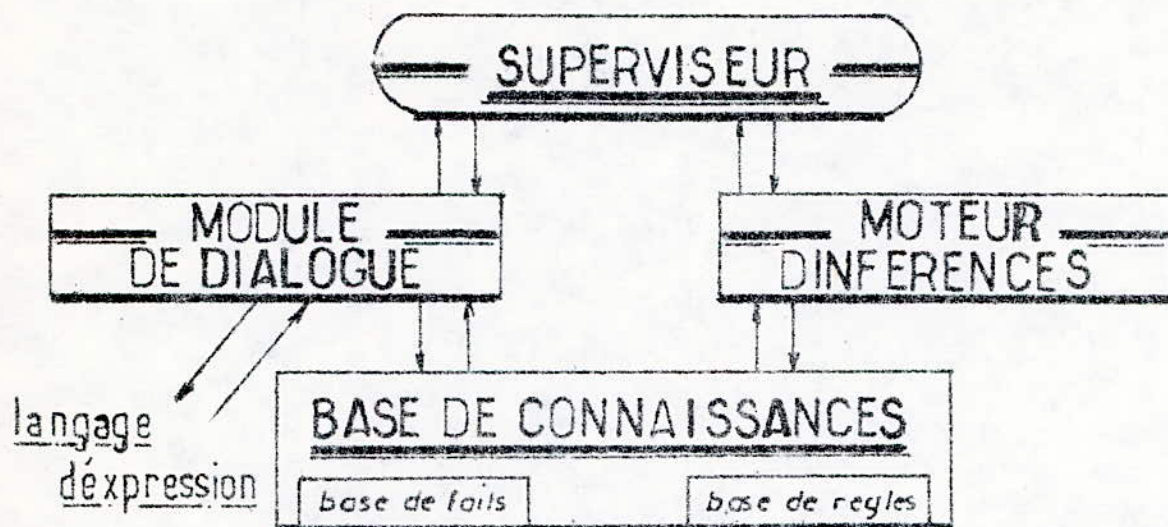
ce domaine comprend :

- . la reconnaissance des formes.
- . la robotique.
- . les jeux.
- . la démonstration automatique de théorèmes.

-LES SYSTEMES D'AIDE AU DIAGNOSTIC_____

- les plus classiques sont les systèmes d'aide au diagnostic :
- . médical.
 - . de pannes de matériel

I-4 ORGANISATION DE PRINCIPES D'UN SYSTEME EXPERT



Un système expert devant satisfaire les caractéristiques de :

- 1 - possibilité de dialogue aisé.
- 2 - capacité de raisonnement
- 3 - connaissance approfondie du domaine d'application

doit comporter :

1- UN MODULE DE DIALOGUE _____

avec un langage d'expression le plus proche possible du langage de l'expert humain.

2- UNE BASE DE CONNAISSANCES _____

pour accueillir la connaissance spécifique d'un domaine d'application. Cette base de connaissances doit être isolée et organisée de manière à permettre une structure évolutive. Cette base est en fait constituée de deux bases distinctes :

a- UNE BASE DE FAITS (BF) :

Cette base représente le " savoir " du système. Il s'agit de situations considérées comme établies, ou comme à établir.

b- UNE BASE DE REGLES (BR) :

Cette base représente le "savoir-faire" du système sur le domaine. Ces règles indiquent quelles conséquences tirer ou quelles actions accomplir lorsque telle situation (fait) est établie ou à établir.

I-5 LE MOTEUR D'INFERENCE

C'est un programme qui met en oeuvre des mécanismes généraux de combinaison de faits et de règles.

Selon différentes stratégies, le moteur d'inférences puise parmi les règles, les interprète, les enchaîne pour déduire de NOUVELLES CONNAISSANCES, jusqu'à satisfaire des conditions d'arrêt dépendant du moteur d'inférences et de la base de connaissances disponible. En général, l'application des règles provoque des modifications de la base des faits, parfois aussi de la base des règles.

I-6 FORMULATION DES REGLES

Chaque règle définit par sa représentation :

- a- les effets de la règle.
- b- les conditions de déclenchement de la règle.

Pour sélectionner une règle particulière, on ne fournit pas le nom de la règle mais un groupe de faits susceptibles d'être compatibles avec les conditions de déclenchement de la règle. Ce mode d'accès est dit associatif. En effet, l'accès à une règle se fait par la donnée d'un fragment d'information (groupe de faits) contenu dans la règle (accès par filtrage ou PATTERN MATCHING).

I-7 FONCTIONNEMENT DU MOTEUR D'INFERENCE

7.1 CYCLE DE BASE D'UN MOTEUR D'INFERENCE.

Le moteur d'inférences enchaîne des cycles de travail comportant deux phases : une phase d'évaluation et une phase d'exécution. La base des règles contient un ensemble de règles rédigées sous formes " SI PREMISSES... ALORS... CONCLUSION ". D'une manière générale, une règle est toujours représentée par :

REGLE = < DECLENCHEUR > + < CORPS >

Au démarrage du moteur d'inférences, la base de faits contient l'énoncé du problème à résoudre, sous la forme de faits établis (expression du problème ou énoncé du problème) et faits à établir (buts du problème).

En PHASE D'EVALUATION : le moteur consulte la base de règles et à l'aide des techniques de filtrage détermine les règles à déclencher selon l'état de la base des faits.

En PHASE D'EXECUTION : le moteur déclenche les règles retenues par la phase d'évaluation. L'arrêt du moteur a lieu :

- soit en phase d'évaluation en cas d'absence de règles déclenchables au vue de la base de faits,
- soit en phase d'exécution lorsque la solution est établie.

7.2 PHASE D'EVALUATION

Elle comprend trois étapes

a- sélection ou restriction___ :

Cette première étape détermine à partir d'un état présent ou passé de la base de connaissances un sous ensemble de faits F1 de la base de faits et un sous ensemble R1 de la base de règles qui méritent d'être comparés dans l'étape de filtrage.

b- filtrage_____ :

Dans cette étape le moteur compare la partie déclencheur de chacune des règles de R1, sélectionnées précédemment, par rapport à l'ensemble des faits F1 . Un sous ensemble R2 de R1 sera alors déduit et comprendra l'ensemble des règles compatibles avec F1 . R2 est appelé " ensemble des conflits".

c- résolution des conflits_____ :

Dans cette étape ,le moteur détermine les règles qui doivent être effectivement déclenchées. Il déduira donc un sous ensemble R3 de R2 .

Le choix des règles à déclencher dépend du moteur d'inférences et rarement de la signification des règles . On peut choisir par exemple les règles les moins complexes ou les premières règles de R2 ou bien encore les règles qui apparaissent les plus prometteuses ou les moins coûteuses .

Si l'ensemble R3 est vide , le moteur d'inférences s'arrête et reconsidère l'ensemble R2, selon le régime de fonctionnement.

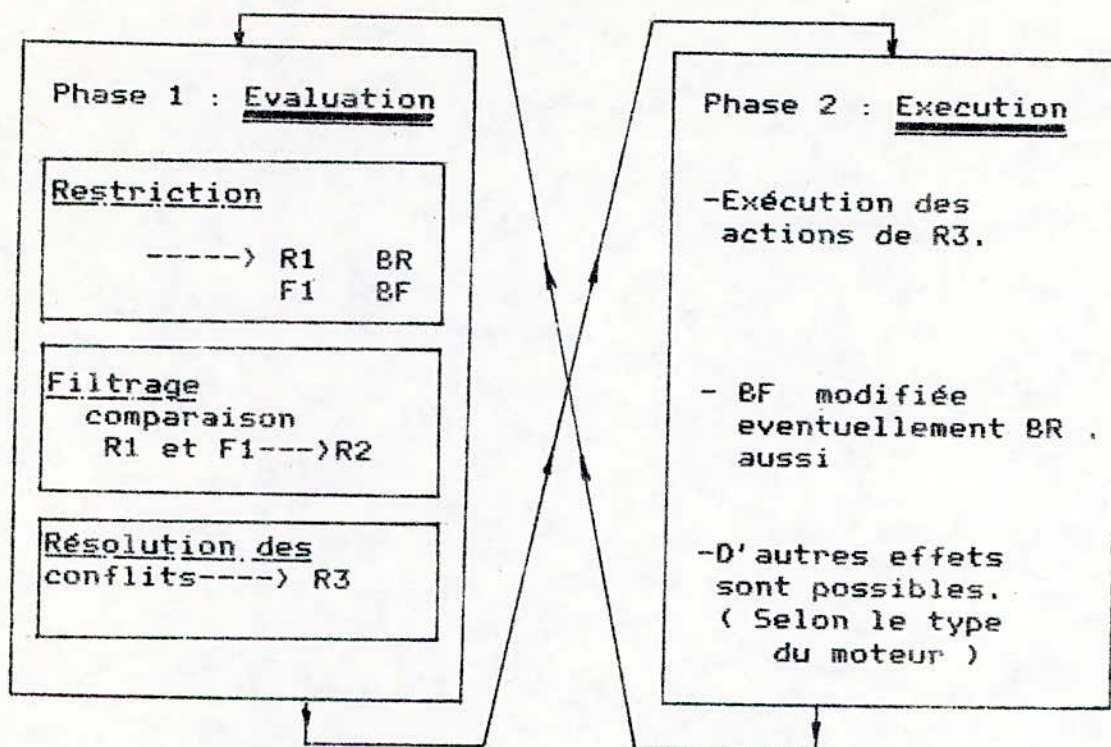
7.3 PHASE D'EXECUTION

Ici le moteur d'inférences met en oeuvre les actions définies par les corps des règles de R3 .Si R3 contient plus d'une règle, la stratégie d'exécution dépend du moteur d'inférences considéré.

7.4 REGIME IRREVOCABLE ET PAR TENTATIVES (BACK-TRAP)

Lorsque l'ensemble R3 est vide , si le moteur d'inférences fonctionne en régime "irrévocable", le moteur s'arrête avec échec. S'il fonctionne en régime par tentatives , il reconsidère l'ensemble des conflits R2 du cycle antérieur et redéfinit un sous ensemble R3 .On dit dans ce cas qu'il effectue un retour arrière ,dans ce cas il annule les déclencheurs sélectionnés à l'étape n , pour les remplacer par ceux de l'étape n-1 .

7.5 REPRESENTATION D'UN CYCLE DE BASE D'UN MOTEUR



-Selon la stratégie :arrêt ou retour-arrière ou autres effets sur le contrôle du moteur .

7-6 MODELE DE COMPATIBILITE

Chaque système offre un modèle de compatibilité entre les faits et les groupes de faits qui composent les déclencheurs. Un modèle de compatibilité peut être selon le système :

a-L'identité:

Ce mode est utilisé par le moteur dit d'ordre 0 (moteur 0) dans ce cas un filtre ne peut être compatible qu'avec une expression de faits parfaitement identique (caractère à caractère).

b-Semi-unification et unification:

Ces deux modes sont utilisés par le moteur d'inférences d'ordre 1 (moteur 1).

- Semi-unification: Les règles peuvent comporter des variables mais pas les faits, les identificateurs de variables dans un filtre sont compatibles avec n'importe quelle constante présente dans un fait.

-Unification : Les règles et les faits peuvent comporter des variables .

7-7 EFFETS DES DECLENCHEMENTS DES REGLES

Les effets des déclenchement de règles sont en général définis comme des transformations de la base de faits . En fait on distingue trois sortes d'effets :

- Instructions de traitement:Effets sur la base des connaissances
- Instructions d'entrée-sortie :Communication externe.
- Instructions de controle :Effets sur le régime du moteur.

8-MODES D'INVOCATION DES REGLES

Il existe quatre modes d'invocation de règles:

a-CHAINAGE AVANT : Un moteur d'inférences fonctionne en chainage avant lorsque:les faits de la base de faits,sur lesquels portent les déclencheurs des règles,représentent des informations dont la valeur de vérité est déjà établie.Les règles sont alors dites "règles en avant " .

Le chainage avant consiste à raisonner des données vers le but .

b-CHAINAGE ARRIERE:Un moteur d'inférences fonctionne en chainage arrière lorsque :

-Certains faits de la base de faits sont considérés comme étant à établir ou à évaluer; ce sont des problèmes ou hypothèses à vérifier ou buts à atteindre .

-Les déclencheurs des règles se reportent uniquement aux problèmes à résoudre .

-Lorsqu'une règle est déclenchée de nouveaux problèmes "définis" peuvent être introduits dans la base de faits .

-Les problèmes auxquels réfèrent les déclencheurs sont considérés comme résolus lorsque le déclenchement de la règle introduit des faits établis ou des problèmes résolus au préalable .Les règles sont alors dites " règle en arrière " . Exploiter de telle règles revient à raisonner des buts vers les faits .

c-CHAINAGE BIDIRECTIONNEL: Un moteur d'inférences fonctionne en chaînage bidirectionnel lorsque il invoque tantôt des règles en avant tantôt des règles en arrière .

d-CHAINAGE MIXTE : Un moteur d'inférences fonctionne en chaînage mixte lorsque:

- Une partie de faits de la base de faits sont considérés comme étant à établir (problèmes à résoudre) et d'autres comme établis.

Les conditions de déclenchement peuvent porter sur des faits de l'une ou de l'autre sorte .

9-LES MODES DE REPRESENTATION

Il existe deux modes de représentation des connaissances:

a-Représentation par espace d'état:L'espace des états correspond à une application de toutes les règles de toutes les facons possibles depuis l'état initial (base de faits initiale) jusqu'à l'état l'objectif correspondant à un succès .

L'état objectif est défini par :

-Une base de faits contenant un fait satisfaisant certaines caractéristiques , pour les moteurs en chaînage avant .

-Une base de faits ne contenant plus de faits à établir ,pour les moteurs en chaînage arrière .

REPRESENTATION



E_i : Etat initial

E_j : Etat final

$E_i E_j$: Application d'une règle R

b-Représentation par espace de sous-problèmes

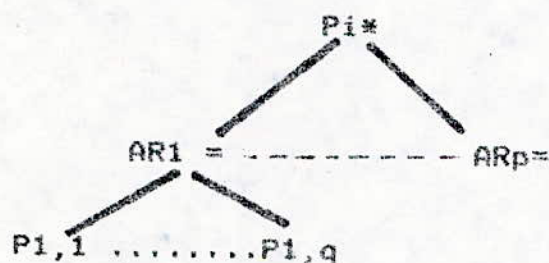
Les règles peuvent être considérées comme des opérateurs de décomposition de problèmes (faits à établir) en sous-problèmes (d'autres faits à établir) . L'espace des sous problèmes est défini par application de tous les opérateurs (règles) de toutes les facons possibles à partir de l'état initial .

L'espace des sous-problèmes peut être représenté par un graphe orienté de type ET-OU .Chaque sommet symbolise soit un problème P_i (sommet problème) , soit une application AR_j d'une règle (ARJ:sommet règle) .

Si AR_1, \dots, AR_n sont toutes des applications de règles qui décomposent P_i (i fixé), l'ensemble des arcs $P_i - AR_j$ détermine une liaison OU.

Si P_1, \dots, P_n sont tous les sous-problèmes de P_i introduits par AR_j , l'ensemble des arcs $AR_j - P_i$ (pour j fixé) détermine une liaison ET.

REPRESENTATION:



Dans cette représentation, un état objectif est atteint si on peut affirmer que le problème initial est résolu au moyen de :

- a-Un sommet problème considéré comme résolu.
- b-Un sommet problème dont un des fils (sommet-règle) au moins est étiqueté résolu.
- c-Un sommet règle dont tous les fils (sommets problèmes) sont étiquetés résolus.

10 STRATEGIES DE DEVELOPPEMENT

Deux stratégies sont utilisées : la stratégie dite "irrévocable" et la stratégie par tentatives.

a-Stratégie irrévocable :
une stratégie est dite irrévocable si les applications des règles choisies lors de la résolution des conflits ne sont jamais remises en cause.

b-Stratégie par tentatives:
Cette stratégie admet des retours arrière. Généralement, les moteurs par tentatives peuvent en cas de besoin fonctionner en régime irrévocable.

11 MONOTONIE

On dit qu'un moteur fonctionne de façon monotone lorsque :

1 - Aucune connaissance , fait établi , ou règle ne peut être retirée de la base .

2 - Aucune connaissance ajoutée à la base n'introduit de contradiction .

Inversement un système est dit non monotone lorsqu'il permet de supprimer ou d'inhiber provisoirement des connaissances (faits ou règles) .

***** CHAPITRE 2 *****

** LES.....STRUCTURES **
** **
** ...DE DONNEES..... **

II.1	INTRODUCTION.....	17
II.2	METHODES DE STRUCTURATION.....	17
2.1	STRUCTURES DYNAMIQUES.....	17
2.2	STRUCTURE DE BASE.....	18
II.3	OPERATIONS SUR UNE LISTE MONODIRECTIONNELLE.....	19

LES STRUCTURES DE DONNEES

II-1 INTRODUCTION

Notre système expert doit comporter une base de connaissances qui permet d'accueillir la connaissance spécifique au domaine d'application. Cette base doit être isolée et organisée de manière à permettre en plus d'un ACCES RAPIDE aux fragments d'informations qui la constituent, une STRUCTURE DE DONNEES EVOLUTIVE qui offre des possibilités de mise à jour telles que :

- Ajout d'une connaissance .
- Modification // // .
- Suppression // // .

II-2 METHODES DE STRUCTURATION DES DONNEES.....

Il est fréquent en informatique d'avoir besoin d'une structure de données permettant de mémoriser les éléments dont le NOMBRE EST INCONNU au moment de l'écriture du programme ou EVOLUE meme constamment au cours de son exécution, c'est le cas, dans notre application, l'association entre une variable déclarée, et ses attributs (son identificateur et son type) est définitivement déterminée dès l'écriture du programme. On ne peut donc pas à l'aide de ces variables (appelées variables statiques) définir une structure de taille non connue au moment de l'écriture du programme.

2.1 LES VARIABLES DYNAMIQUES.

Les structures dynamiques autorisent la définition d'une structure de taille inconnue, en allouant un espace mémoire pendant l'exécution du programme. Contrairement aux variables statiques, les variables dynamiques n'ont pas d'identificateurs et leur déclaration n'est pas faite explicitement. L'accès à ces variables ne se fait pas par un identificateur mais par une référence. La variable où se trouve cette référence est appelée "POINTEUR".

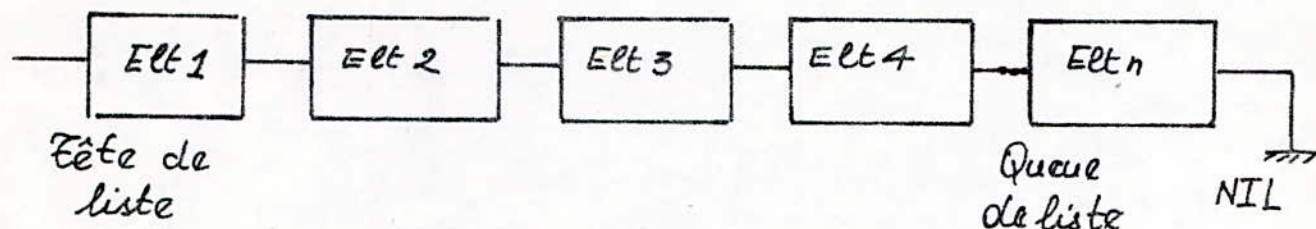
Un pointeur est une variable (statique ou dynamique) dont la valeur repère une variable créée dynamiquement.

2.2 STRUCTURES DE BASE

La liste chaînée est la structure évolutive la plus fondamentale. Elle consiste en la succession d'éléments ordonnés, liés entre eux.

a- Liste monodirectionnelle

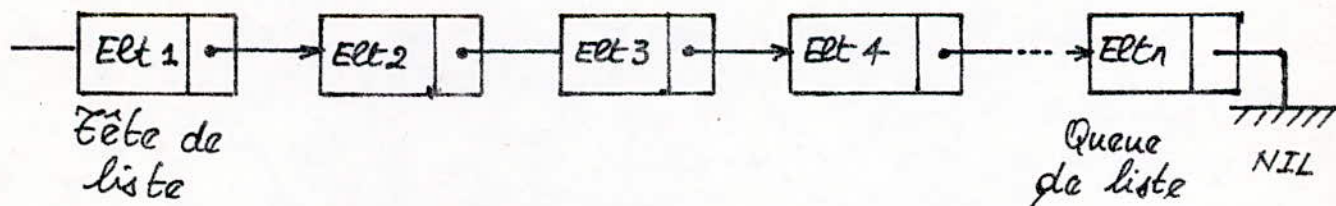
Schématiquement une telle structure a l'allure suivante :



Le pointeur de tête mémorise l'adresse, en mémoire centrale, du premier élément. Chaque élément ou maillon peut être décomposé en deux parties :

- une, contenant la valeur de l'élément (entier, réel, chaîne de caractères) qu'on peut exploiter.

- l'autre, l'adresse de l'élément suivant conformément au schéma ci-dessus



II-3 OPERATIONS SUR UNE LISTE MONODIRECTIONNELLE.....

Plusieurs opérations peuvent être appliquées à une liste :

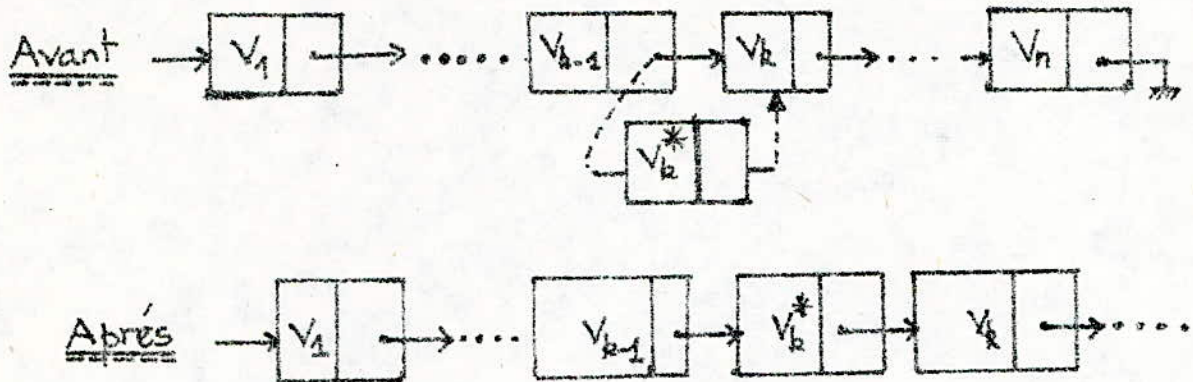
a-recherche d'un élément

la recherche d'un élément se ramène à la recherche de son adresse en mémoire centrale.

Cette recherche se fait par parcours de cette liste à partir de son premier élément. Les pointeurs permettent le balayage de la liste jusqu'à identification de l'élément.

b- Ajout d'un élément

Schématiquement cette opération se fait de la manière suivante :



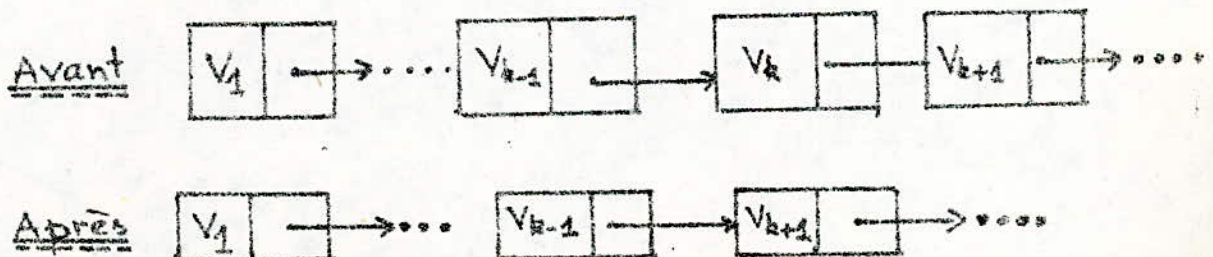
Pour effectuer cette opération , il faut donc :

- 1 - rechercher l'adresse de l'élément $V(k-1)$.
- 2 - supprimer la liaison $V(k-1) - V(k)$.
- 3 - établir la liaison $V(k-1) - V(k^*)$.
- 4 - // // // $V(k^*) - V(k)$.

Et ceci en modifiant les adresses des pointeurs des maillons $V(k-1)$ et $V(k^*)$.

c- Suppression d'un élément

La suppression d'un élément peut être illustrée par le schéma suivant :



Pour effectuer cette opération il faut donc :

- 1 - rechercher l'adresse de l'élément $V(k-1)$.
- 2 - détruire la liaison $V(k-1) - V(k)$.
- 3 - établir / / $V(k-1) - V(k+1)$.

***** CHAPITRE 3 *****

** REALISATION..... DU **
** **
** MOTEUR D'INFERENCEES **

III.1	INTRODUCTION.....	22
III.2	CARACTERISTIQUES DU M.I.....	22
III.3	POSITION DU PROBLEME.....	22
III.4	PRINCIPALES PARTIES CONSTITUANT LE MOTEUR D'INFERENCE.....	24
III.5	LES FICHIERS UTILISES.....	25
III.6	LE SUPERVISEUR.....	27
III.7	LE MODULE DE DIALOGUE.....	29
III.8	LE MODULE D'APPRENTISSAGE.....	47
III.9	LE MODULE DE RAISONNEMENT.....	50

REALISATION DU M.I

III-1 BUT

L'écriture de notre système expert se décompose en deux parties :

- écriture d'un moteur d'inférences.
 - implémentation d'une base de connaissances relative aux pannes T.V, qui sera développée ultérieurement.
- Le moteur d'inférences doit à partir des connaissances disponibles et effectuer un raisonnement afin d'induire ou de déduire de nouvelles connaissances.

III-2 CARACTERISTIQUES DU M-I :

Notre moteur d'inférences doit répondre aux caractéristiques suivantes :

- 1- Le mode d'invocation des règles est le chaînage-mixte. Son principal avantage est de permettre un fonctionnement en chaînage arrière ou en chaînage avant selon la structure de la règle.

La stratégie de recherche de la solution est une stratégie en "profondeur d'abord", c'est à dire que le moteur s'intéresse en premier lieu aux problèmes les plus récents.

Notre moteur d'inférences utilise une stratégie par tentatives, c'est à dire qu'il effectue des retours arrière en cas d'échec dans la voie de raisonnement choisie.

Il fonctionne de façon "monotone". On ne peut pas tirer de la base des connaissances une règle ou un fait établi ; les connaissances ajoutées à la base ne doivent pas introduire de contradiction.

- 2- C'est un moteur "0", le modèle de compatibilité utilisé pour la sélection des règles est l'identité (on n'utilise pas de variables dans les règles et dans les faits).
- 3- Il est écrit en PASCAL. Ce langage est structuré, il offre une allocation dynamique de mémoire et permet d'écrire des procédures "récurssives": (PASCAL ISO : transportabilité et adaptabilité)

III-3 POSITION DU PROBLEME :

Notre moteur d'inférences doit permettre de :

- Vérifier une hypothèse .(mixte)
- Faire des déductions .(avant)

-Faire des inductions . (mixte)

Pour chacun de ces cas , il doit utiliser le mode d'invocation le mieux adapté et les connaissances les plus appropriées (choisir le mode d'invocation et le type de règles) pour arriver à la solution .

3-1 Vérification des hypothèses (Fonctionnement en chainage mixte)

- . Notre moteur doit pouvoir vérifier la validité d'une ou plusieurs hypothèses à la fois .
- . La base de connaissances est composée dans ce cas de :
 - Un ou plusieurs faits à établir (hypothèses à vérifier)
 - Un ou plusieurs faits établis
- . A partir d'une base de connaissance avec :
 - Base de faits non établis: les deux triangles sont semblables.
 - base de faits établis : il ont deux angles égauxLe moteur doit être capable de répondre par : "oui cette hypothèse est vrai".
Le mode d'invocation le mieux approprié dans ce cas est le chainage mixte.

3-2: DEDUCTIONS:

- Le moteur doit dans ce cas , à partir d'un ou plusieurs faits établis, déduire d'autres faits.
- . La base de connaissances dans ce cas :
 - Ne contient pas de faits à établir.
 - Contient un ou plusieurs faits à établir.
 - . A partir du fait "Il pleut " , le moteur devra déduire :
 - La terre est mouillée.
 - Il y'a des nuages.
 - Il faut porter un parapluie.
 - Il faut rouler doucement.
 - La chaussée est glissante.
 - . Le mode d'invocation le plus approprié dans ce cas est le chainage avant.

3-3: LES INDUCTIONS:

Dans ce cas le M-I doit pouvoir résoudre un ou plusieurs problèmes posés .

- . La base de connaissances dans ce cas comporte :
 - Un ou plusieurs faits à établir.
 - Un ou plusieurs faits établis.

III-4 PRINCIPALE PARTIES CONSTITUA

Pour faciliter l'approche , nous avons scindé le M-I en quatre parties essentielles que nous avons nommé respectivement :

- Le superviseur
- Le module de dialogue
- Le module d'apprentissage
- Le module de raisonnement

Pour chacune des parties citées , nous nous sommes fixés des objectifs

4-1 :Le superviseur :

Il a pour but de contrôler le bon fonctionnement du système expert Il suit un cheminement très précis:

- 1 - Au démarrage du M-I , il restaure la base de connaissances (base de règles et base de faits), à partir de la mémoire de masse (disquette ou disque dur dans notre cas)
- 2 - Il utilise le module de dialogue comme moyen de communication avec l'opérateur.
- 3 - Suivant leurs types les données recueillies de l'extérieur sont aiguillées soit vers le module apprentissage soit vers le module raisonnement.
- 4 - Il redonne le contrôle au module de dialogue pour afficher les résultats.
- 5 - En fin il sauvegarde la base de connaissances sur mémoire de masse .

4-2 :LE MODULE DE DIALOGUE :

Cette partie a pour but de communiquer avec l'opérateur dans un langage à la fois très précis et proche du langage du langage naturel , elle sera chargée de :

- 1 - saisir les données quelque soit leur type : (nouvelle règle à insérer ou nouvelle hypothèse à vérifier) .
- 2 - vérifier la validité des données entrées par l'opérateur et émettre un message clair en cas d'erreur .
- 3 - adapter ces données à la structure interne (codage en entier) .
- 4 - afficher les résultats dans une forme lisible (decodage)

4.3 MODULE D'APPRENTISSAGE

Ce module a pour but :

- d'insérer des nouvelles règles .
- de supprimer des règles
- consulter la base de connaissances en particulier la base de règles

4.4 MODULE DE RAISONNEMENT

Ce module utilise le " savoir faire " du système (base de règles pour :

- vérifier une hypothèse
- tirer des conclusions à partir de faits établies

REMARQUE

Dans le programme que nous avons écrit ces modules n'apparaissent pas de façon explicite . Ceci est due au fait qu'elles ont plusieurs parties en commun : elles s'enchevêtrent.

III.5 LES FICHIERS UTILISES

Afin de répondre au critère " base de connaissance isolée " , nous avons prévu trois fichiers pour la sauvegarde sur mémoire auxiliaire . Ce sont les fichiers de REGLES , de FAITS et DICTIONNAIRE .

Deux fichiers supplémentaires nous ont semblé nécessaires pour améliorer le contact Homme-Machine (module de dialogue) ce sont les fichiers RESULTATS et IMPRIMANTE .

1 - FICHIER DES REGLES

Ce fichier permet de sauvegarder la base de règles à la sortie du M.I et de la restaurer au démarrage du M.I . C'est un fichier à accès séquentiel et de type binaire .

2 - FICHIER DICTIONNAIRE

Ce fichier contient tous les mots (ou groupe de mots) qui ont été utilisés par l'utilisateur , dans les règles . Il permet de limiter l'espace mémoire alloué à chaque règle en mémoire centrale et de vérifier la validité syntaxique d'un problème . C'est un fichier à accès direct et de type binaire .

3 - FICHIER DES FAITS

Ce fichier joue deux rôles :

a / à la fin d'une séance de travail , il permet de sauvegarder la base de faits (BFE) .

b / au début d'une séance de travail , il permet de restaurer soit l'ancienne base de faits , soit une nouvelle base de créée par l'utilisateur sous éditeur .

C'est un fichier à accès séquentiel de type text . Il peut donc être consulté en dehors du programme par un " TYPE " ou sous éditeur .

4 - FICHER RESULTATS_____

Ce fichier contient les résultats d'exécution des modules d'apprentissage et de raisonnement . Il peut être consulté en dehors du programme avec un " TYPE " ou sous éditeur . C'est un fichier à accès séquentiel et de type text .

5 - FICHER IMPRIMANTE

Nous avons ouvert ce fichier pour permettre une sortie sur imprimante des résultats d'exécution . L'utilisateur a la possibilité de valider ou d'inhiber cette sortie . Sous MS-DOS le nom de cette sortie est "PRN.XXX" . C'est un fichier à accès séquentiel et de type text .

REMARQUE_____

Nous n'avons pas fixé les noms relatifs à ces fichiers dans le programme . Ceci permet à l'utilisateur de créer ses propres applications : c.à.d définir de nouvelles bases de connaissances et ainsi de réaliser des systèmes experts dans les domaines souhaités

III.6 LE SUPERVISEUR

Il a pour rôle celui d'assurer le bon fonctionnement du système expert grâce à un cheminement très précis, dont les principaux points sont illustrés par l'algorithme ci dessus:

6-1 ALGORITHME

-Au démarrage du MI le superviseur fait appel à la procédure HELLO qui permet :

a- de saisir les noms des différents fichiers : fichier des règles , fichier dictionnaire , fichier des faits , fichier résultats. Le nom du fichier de l'imprimante est dicté par système d'exploitation (MS-DOS) .

b- de restaurer la base de connaissances (base de règles et base de faits) à partir de la mémoire de masse .

c- de déterminer le nombre de mots contenus dans le fichier dictionnaire.

d- d'initialiser les fichier resultats et imprimante.

e- d'appeler la procédure masque qui ouvre une fenêtre et affiche un menu permanent sur la partie supérieure de l'écran.

Ce menu est composé de la:

1 .Sauvegarde :pour sauvegarder la base de connai- SUPERVISEUR
-ssances

2 .Consultation: pour consulter la base de règles.

MODULE

3 .Supprime: pour supprimer une règle.

APPRENTISSAG

4. Append (ajout) :pour ajouter une règle.

MODULE

5. Démontre : pour résoudre une problème

RAISONNEMENT

f- de saisir le choix de l'utilisateur grace au module de dialogue . Le module de dialogue teste la validite de ce choix .

- a. si non, afficher un message d'erreur.
- b. si oui, appeler la routine choisie .
- c. si la routine choisie, est sauvegardée, à la fin de l'exécution de celle-ci ,il y a sortie du programme.
- d. si non le cycle recommence , à partir de "f" par un appel récursif .

Remarque: si lors de la restauration ,un des fichiers n'existe pas ,le programme crée ce fichier pour le préparer à recevoir des données lors de la sauvegarde .

6-2 PROCEDURE SAUVEGARDE

Elle permet de sauvegarder la base connaissance sur mémoire de masse. Les étapes suivies sont :

a- Sauvegarde de la base de règles

les trois listes définies, sauvegardées à tour de rôle sur le fichier de règles . Les pointeurs permettent de balayer les différentes listes du début jusqu'à la fin et de respecter aussi l'ordre .

b- Mise en place de la marque de fin du fichier dictionnaire

le fichier dictionnaire étant à accès direct , aucune instruction ne permet de reconnaître sa fin (il n'y a pas de fin physique). On a donc eu recours à un artifice. Le nombre de mots contenu dans le dictionnaire étant connu ,il suffit de placer une marque de fin artificielle. Cette marque servira lors du MI , pour déterminer ce nombre .

c- Sauvegarde sur fichier des faits

les faits de la base des faits sont sauvegardés sur ce fichier .

d- Fermeture de tous les fichiers

à cette étape on passe les cinqs fichiers utilisés. Ceci n'est pas indispensable , mais permet de libérer des canaux d'E/S utilisés par le système .

III.7 MODULE DE DIALOGUE

Z-1 INTRODUCTION

Le premier problème auquel nous avons été confrontés lors de l'écriture du MI, fût le choix d'un mode de représentation de la règle et des problèmes à résoudre. L'idée la plus simple, est de définir une structure en chaîne de caractères de longueur suffisante pour accueillir le texte de n'importe quelle règle. Cette simplicité apparente cache derrière elle, plusieurs problèmes. Elle constituera un handicap lors de la vérification des hypothèses. En effet rien ne permet de distinguer les parties prémisses et conclusion entre elles et les faits entre eux. On notera, qu'il s'agit là d'un ancien problème en I.A celui de doter l'ordinateur de facultés de compréhension et d'analyse de texte : problème non résolu jusqu'à ce jour.

Z-2 REPRÉSENTATION D'UNE RÈGLE

Avant de chercher la solution, examinons de plus près la structure d'une règle. En chaînage mixte, une règle a ses parties prémisses et conclusion, qui contiennent en plus de faits établis (FE) des faits non établis (FNE). Ceci nous donne le schéma suivant :

< Règle >	=	< Prémisses >	+	< Conclusion >
< Prémisses >	=	< 1 FNE >	+	< Des FE >
< Conclusion >	=	< 1 ou des FNE >	+	< Des FE >

On peut donc recenser quatre parties de type FE ou de type FNE. Il s'agit donc de

- Séparer la partie prémisses de la partie conclusion.
- Distinguer un FE d'un FNE.
- Séparer les FE ou FNE.

Pour répondre à ces exigences, nous avons imposé une syntaxe à l'utilisateur dont les principales bases sont :

- 1-Séparer la partie prémisses de la partie conclusion par le signe " = " .
- 2-Faire précéder un FNE du signe " ? " .
- 3-Séparer les faits établis ou non par une virgule " , " .

Ainsi une règle telle que : "s'il pleut alors il y a des nuages " s'écrit : " il pleut = il y a des nuages "

Pour généraliser une règle formulée explicitement par :

Si	FNE	à	vérifier	sachant	FE 1
					et FE 2
					.
					et FE n
Alors	vérifier	Problème 1	sachant	Fait n+1	
		et Problème 2		et Fait n+2	
		.			
		et Problème p		et Fait m	

doit être rédigée comme suit :

?Problème, Fait 1, ..., Fait n=?Problème 1, ..., problème p, Fait n+1, ..., Fait

Représentation d'un problème

Un problème peut avoir l'une des trois structures suivantes :

1-il comporte des FE et des FNE .

2-// // // FNE .

3-// // // FE .

Il s'agit donc de séparer les faits entre eux et de distinguer un FNE d'un FE .

Pour garder la compatibilité entre les règles , nous avons choisi la même syntaxe : les règles 2 et 3 s'appliquent directement . La règle 1 est inutilisable ici du moment qu'on n'a pas de parties prémisses et conclusion à séparer . Ainsi un problème tel que :

VERIFIER que la THT est défectueuse SACHANT que l'écran est sombre
et qu'on a pas de son .

doit être rédigée sous la forme :

? THT défectueuse ,Ecran sombre ,Pas de son .

Pour généraliser , un problème formulé explicitement par :

VERIFIER	hypothèse 1	SACHANT	Fait 1
	et hypothèse 2		et Fait 2
	.		.
	et hypothèse n		et Fait m

doit être rédigé comme suit :

? Hypothèse 1,hypothèse 2,...,hypothèse n,Fait 1,Fait 2,...,Fait m

7-3 REMARQUE

-Nous avons retenu cette syntaxe car elle offre les critères de clarté et de simplicité et met en évidence les différentes parties d'une règle ou d'un problème à résoudre (Prémisse , conclusion , FE et FNE).

-Pour la rédaction des faits aucune contrainte n'est imposée, ce qui se traduit par un langage clair : Ecran sombre, HP défectueux, etc... Néanmoins nous avons limité le nombre de caractères pour les faits à 20 et celui de toute la règle ou tout le problème résoudre à 160 . Cette dernière restriction est avant tout d'ordre technique . En effet , le micro-ordinateur (RAINBOW 100) sur lequel nous avons développé notre logiciel n'accepte pas une saisie de plus de deux lignes de texte (160 caractères) .

-Une conversion automatique majuscule minuscule est prévue dans dans le programme .

-Pour accueillir le texte de la règle (problème à résoudre) nous avons défini une variable de type chaîne de caractères avec une longueur de 160 caractères .

Au niveau des déclarations on a : CHAINE PACKED ARRAY [1..160] OF CHAR

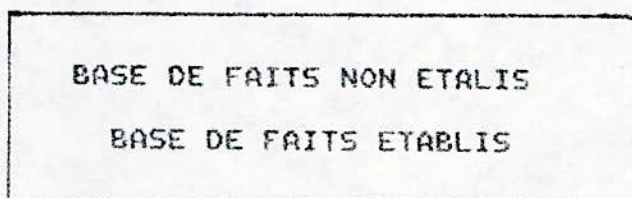
et VAR C:CHAINE 160

Après cette saisie ,le texte subit des opérations de codage et de transformation en entiers dans le but de condenser et d'améliorer le temps d'exécution .

7.3 STRUCTURE INTERNE DES DONNEES

a-Base de faits (B.F)

La base de faits est constitué d'un ensemble de faits établis et d'un ensemble de faits non établis .

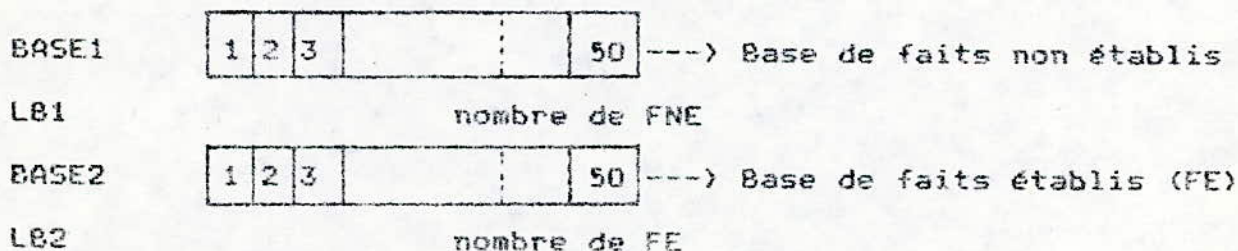


Nous avons défini une structure de vecteur pour supporter ces deux bases , les élément du vecteur étant des entiers qui réfèrent par leur valeur à un fait (les raisons de ce choix seront développées ultérieurement)

Le nombre de FE et de FNE est à priori inconnu et variable dans le temps .Nous avons fixé la dimension des tableaux à 50 ,le nombre est largement suffisant .Dans la pratique n'excède guère une dizaine de faits.Nous avons défini deux variables qui contiennent le nombre de faits non vérifiés présent dans la base de faits non établis (FNE) et le nombre de faits établis présent dans la base de faits établis. Au niveau des déclaration nous avons :

```
TYPE TABLEAU50=ARRAY [1..50] OF INTEGER;  
VAR BASE1,BASE2 :TABLEAU50;  
LB1,LB2 :INTEGER;
```

Scématiquement on a la structure suivante :



b-Structure d'une règle

Une règle est constituée d'une partie prémisses et d'une partie conclusion. Chacune de ces parties contient des FE des FNE. Schématiquement on a :



Pour accueillir toutes ces données, nous avons défini une structure d'enregistrement (RECORD) avec 8 champs, 4 relatives aux différentes parties de la règle, les 4 autres étant destinées à contenir le nombre de faits dans chacune d'elle.

Au niveau du programme, nous avons fait la déclaration suivante:

```
TYPE TABLEAU15 : ARRAY(1..15) OF INTEGER;
REGLES : RECORD
  C1,L1,L2,L3,L4:INTEGER;
  C2,C3,C4 :TABLEAU 15;
END;
```

C1 est le champ correspondant à la partie FNE de la prémisse. Celle-ci contient un fait dans le cas du chaînage mixte et zéro faits dans le cas du chaînage avant. Nous avons donc déclaré une variable simple à cet effet.

L1 indique l'existence ou non de FNE et prend les valeurs 0 ou 1.

C2, C3, C4 correspondent respectivement aux parties FE (de la prémisse), FNE (de la conclusion) et FE (de la conclusion). Nous avons fixé pour ces parties des dimensions assez grandes (15). Dans la pratique on n'excède guère une dizaine de faits.

L2, L3, L4 correspondent respectivement à C2, C3 et C4 et représente le nombre de faits dans ces vecteurs.

Remarque:

Au niveau de la saisie des règles l'utilisateur peut par mégarde dépasser ces limites, nous avons prévu dans le module des messages d'erreurs pour l'utilisateur. La liste de ces messages sera donnée ultérieurement.

c-Structure de la base des règles.

Les règles telles qu'elles ont été définies pouvaient être groupées dans une structure de vecteurs. Ce choix n'a pas été fait car le nombre de règles est inconnu au moment de l'écriture du programme, nous avons donc choisi une structure dynamique la liste mono-directionnelle pour la base des règles. Ce choix autorise des opérations d'ajout et de modification. Pour la création et la gestion de cette liste, nous avons défini deux pointeurs: l'un pour la tête l'autre pour la queue de liste. Nous avons aussi défini une variable qui contiendra le nombre de règles disponibles dans la base de règles.

Les déclarations utilisées sont les suivantes:

TYPE

Elien : ^Règles

Règles : RECORD

PE : Elie;

C1, L1, L2, L3, L4 : Integer;

C2, C3, C4 : Tableau 15;

END;

VAR AD, B0 : Elie;

N : Integer;

VALUE A :=NIL , B:=NIL;

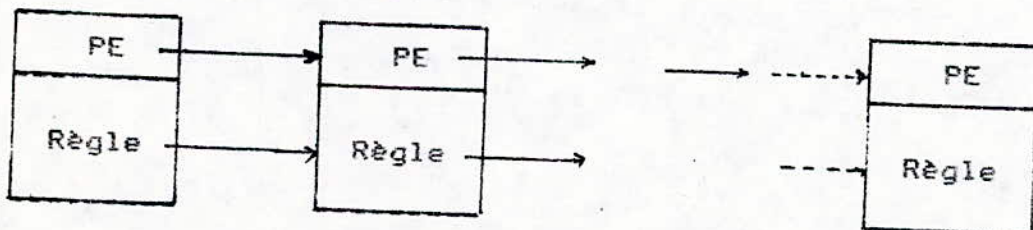
A étant le pointeur de tête de la liste des règles.

B étant le pointeur de queue de la liste des règles.

PE étant le pointeur sur l'élément suivant.

N étant le nombre de règles.

Schématiquement nous avons:



Afin d'améliorer le temps d'exécution nous avons opté pour une structuration en trois sous-listes, dont les détails seront donnés ultérieurement.

d-Pourquoi une structure avec trois listes?

La réponse à cette question ne peut être donnée à ce stade. Il est en effet nécessaire de connaître d'abord le mode de fonctionnement du module raisonnement. Nous pouvons toutefois dire que c'est dans le but d'optimiser la recherche de la solution d'un problème et d'améliorer ainsi le temps de résolution dans le module de raisonnement, que nous avons scindé la base de règles en trois parties.

1ère Liste

La première liste est destinée à recevoir des règles dont la partie conclusion ne contient pas de FNE. c'est à dire des règles qui résolvent les problèmes du type:

Problème =====> Solution

2ème Liste

La deuxième liste est destinée à recevoir les règles dont la partie conclusion contient un seul FNE. c'est à dire des règles qui remplacent un problème par un autre.

Problème 1 =====> Problème 2

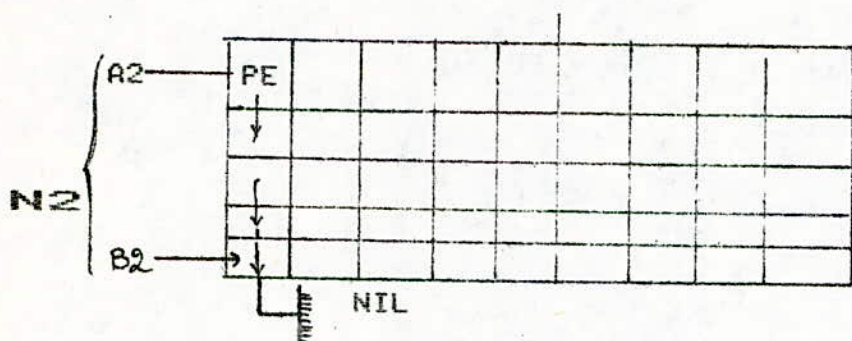
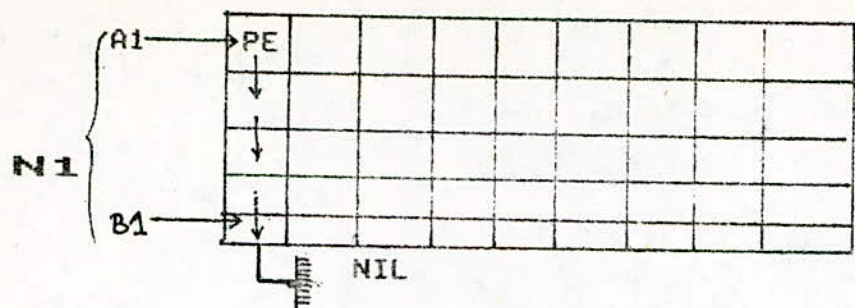
3ème Liste

La troisième liste est destinée à recevoir les règles dont la partie conclusion contient au moins deux FNE. c'est à dire les règles qui décomposent un problème par plusieurs sous-problèmes.

Problème =====> Série de problèmes

Cette structure confère à notre moteur d'inférences une propriété supplémentaire non prévue dans le cahier de charges, nous pouvons donc dire que la base de règles est ordonné.

Des explications plus détaillées sur les avantages de cette structuration seront données dans la partie module de raisonnement.



e-Codage des règles et des problèmes en entiers (INTEGER)

Les raisons de ce codage ont déjà été cités dans le paragraphe "Pourquoi une structure à base d'entiers". Nous allons présenter les moyens et les solutions mis en oeuvre pour y parvenir.

. Codage des règles

La représentation générale au moment de la saisie sur clavier est la suivante :

?Pb, Fait 1, Fait 2, ..., Fait n = ? Pb 1, Pb 2, ..., Pb p, Fait 1, ..., Fait m

L'algorithme que nous avons utilisé peut être schématisé comme ci-dessous

- Séparation des parties droite et gauche grâce au signe "="
on obtient deux fragments

?Pb, Fait 1, Fait 2, ..., Fait n -----> Partie droite

?Pb 1, Pb 2, ..., Pb p, Fait 1, Fait 2, Fait m -----> Partie gauche

- Pour chacune des parties , on sépare les problèmes et les faits on aura donc deux parties:

PARTIE 1

?Problème
Fait 1
Fait 2
· · · · ·
Fait n

PARTIE 2

?Problème 1
?Problème 2
· · ·
?Problème p
Fait 1
· · ·
Fait m

Ceci est possible grâce au signe "?" inséré entre les faits et les problèmes .

- Chaque fait ou problème est considéré à part .Le signe "?" permet de faire la différence .

Sachant qu'on dispose de N mots dans le dictionnaire , pour trouver le code d'un fait :

- On parcourt le dictionnaire depuis le premier mot : deux cas peuvent se présenter :

a- Le mot existe déjà dans le dictionnaire . Dans ce cas on attribut à ce mot l'ordre du mot dans le dictionnaire ,c'est à dire si le mot "THT défectueuse" est trouvé en 10 ème position ,on remplace ce mot par l'entier 10 .

b- Si le mot n'existe pas on l'insère à la position $N + 1$ et on incrémente N . Le code sera bien entendu le nombre de règles .

- On remplit ainsi les champs C1 , C2 , C3 , C4 tout en incrémentant les valeurs correspondantes L1 , L2 , L3 , L4 jusqu' à la fin du codage .

- Des messages d'erreurs sont lancés par le module de dialogue c'est-à-dire :

L1 > 1
L2 > 15
L3 > 15
L4 > 15

(L1 = 0 et L2 = 0) Partie gauche inexistante .

(L3 = 0 et L4 = 0) Partie droite inexistante .

La règle codée est transférée vers le module d'apprentissage qui s'occupera de l'insérer dans la liste qui convient et dans la place qui lui a été prévue .

Pour effectuer ces tâches nous avons écrit quatre procédures que nous avons nommé CODE , SEPRE , MAJUSCULATION et PURGE .

1. Procédure CODE

Elle sépare les parties de droite et gauche de la règle , elle permet aussi de séparer les faits des problèmes. Cette procédure sépare les faits entre eux et les problèmes entre eux. Par appel à la procédure SEPRE la procédure CODE permet d'avoir le code d'un mot .

1. Procédure SEPRE

Elle reçoit un mot de la part de la procédure CODE , elle fait aussi appel à la procédure MAJUSCULATION pour la transformation majuscule minuscule . Elle permet aussi de chercher le code du mot en parcourant le dictionnaire .

1. Procédure MAJUSCULATION

Elle sert à transformer des caractères minuscules en caractères majuscules . Le but est de rendre équivalent le code des mots écrits sous les deux formes .

1. Procédure PURGE 1

L'utilisateur peut introduire le même fait plus d'une fois , la procédure PURGE 1 se charge d'éliminer dans les champs C2 , C3 et C4 les faits supplémentaires .

1. Codage d'un Problème

La représentation générale d'un problème au moment de la

saisie sur clavier est la suivante :

?Pb 1 , ?Pb 2 , ..., Pb p , F 1 , F 2 , ..., F n .

On peut remarquer que cette structure est la même que celle de la partie prémisses . Il est donc logique de penser à utiliser les outils décrits pour le codage de la règle . Toutefois ces outils ne peuvent pas être appliqués directement .

En effet ceci générerait des erreurs dans le cas où le problème à résoudre contient plus d'un fait non établi (L1 > 1) . La solution est de faire en sorte que l'algorithme considère le le texte du problème non pas comme une partie prémisses mais mais comme une partie conclusion . Pour parvenir à ce but , il suffit d'insérer un signe " = " en début du texte .

La transformation donne la structure ci-dessous :

= ?Pb 1 , ?Pb 2 , ..., Pb p , F 1 , F 2 , ..., F n

Une fois l'algorithme exécuté on recueille dans les champs (C3 ,L2) et (C4 ,L4) le problème codé . Il suffit de le transformer dans les variables de destination qui sont (Base1 ,LB1) et (Base2 ,LB2) .

f-Pourquoi une structure à base d'entiers?

Cette structure ne nous a pas été imposée . Elle fut le d'un travail de réflexion . Cette structure offre d'innombrables avantages dont voici les plus importants :

1 - Elle permet de réduire l'espace mémoire utilisé

En effet si chaque fait été défini sous forme de chaîne de caractères , il prendrait vingt octets en mémoire (ceci correspond à la longueur maximale , que nous avons fixé à l'utilisateur). Avec cette structure,chaque fait est représenté par un entier qui prend deux octets de mémoire .On a directement un rapport de 1/10 .

2- Elle permet de diminuer le temps d'exécution

En effet on peut noter que :

- Les structures de chaîne de caractères sont des structures en entiers. Le temps d'accès aux structures compactées (PACKED ARRAY) est classé comme étant long comparé à celui des autres structures .

- Grâce à cette structure , la comparaison de la base des faits avec les filtres des règles revient à une comparaison d'entiers :ceci donne une grande souplesse pour l'écriture des procédures de module de raisonnement et notamment celle concernant la phase de filtrage .

Cette structure,nous oblige, à définir évidemment, une table de conversion des entiers ,vers les chaînes de caractères , c'est à dire le passage d'un nombre tel que 43 vers le fait " T H T défectueuse" . Ceci a été effectué à l'aide d'un fichier à accès direct, que nous avons appelé fichier dictionnaire et dont nous donnerons ultérieurement plus de détails . Ainsi sa création n'a pas été vaine.

En effet ce fichier contenant tous les faits introduits par l'utilisateur , défini de lui même un langage reconnu par le système .

Lors de la saisi des problèmes à résoudre , nous avons inclu une vérification syntaxique très simple de la requête de l'utilisateur , par vérification des mots reconnus par le système et contenus dans ce fichier . Ceci permet un gain de temps appréciable .

Ainsi , si la base de règles contient une centaine de règles ,et si ce contrôle n'a pas été ajouté,le module de raisonnement aurait confronté, les cents règles avec la la base des faits introduite par l'utilisateur .

9-Amélioration du contact homme-machine

Afin d'améliorer le contact homme-machine , nous avons accru le nombre des voix de sorties des résultats à trois et mis ces sorties sous le contrôle de l'utilisateur. Il peut valider celle qui lui convient et inhiber celle qui ne lui convient pas . Dans le cas d'une inhibition totale, on a la vitesse maximale d'exécution.Seules les résultats finaux sont affichés à l'écran . Ces sorties sont :

- Sortie écran : c'est la sortie standard.Nous avons mis par cette sortie un contrôle . On peut obtenir une exécution du MI, avec affichage des résultats intermédiaire (Mode Trace) ou non .

Nous avons aussi amèlioré l'efficacité du contact homme-machine en :

- . affichant un menu permanent à l'écran .
- . affichant les messages importants ,dans des modes qui les mettent en évidence (soulignée , brillance inverse ,vidéo , clignotement) .
- . en cas d'erreur , nous avons prévu en plus de l'avertissement un beep .

Les outils mis en oeuvre sont basés sur les séquences escape de la norme AINSI (American National System Institute) . Cette norme est disponible actuellement sur la plus part des ordinateurs , en particulier elle existe sur le (RAINBOW 100,OLIVETTI M 24 et le VAX 750). notre programme ne perd ainsi rien de sa transportabilité et de son adaptabilité .

Sortie sur imprimante :La plus part des micro-ordinateurs disposent d'une imprimante .Sous MS-DOS cette sortie est considérée comme un fichier texte à accès séquentiel .Le nom de ce fichier est standard : "PRN.XXX". Nous avons donc rajouté cette sortie , tout en offrant à l'utilisateur la possibilité de la valider ou de l'inhiber .

Sortie sur mémoire de masse : nous avons inclu cette

sortie afin de permettre à l'utilisateur de :

- Consulter les résultats en dehors du programme . Ceci est possible en utilisant les commandes TYPE ou EDIT .
- Imprimer ces résultats après une exécution rapide du moteur d'inférences.Ceci est possible grâce à une des deux commandes MS-DOS suivantes :

PRINT/P Nom du fichier résultat

COPY Nom du fichier résultat PRN.XXX

- Imprimer ces résultats après modification de ce fichier sous l'éditeur .Ceci permet à l'utilisateur de garder uniquement les résultats intéressants .

Nous avons juger inutile de mettre un contrôle sur cette sortie. En effet il est toujours intéressant d'avoir une trace de son exécution . De plus ce fichier peut être détruit à tout moment grâce aux commandes du système MS-DOS ERASE et DELETE Nom du fichier .

III.B MODULE D'APPRENTISSAGE

BUT

Ce module a pour rôles :

- d'insérer une nouvelle règle .
- de supprimer une règle .
- de permettre à l'utilisateur de consulter la B.R .

A. INSERTION

La règle n'est pas exploitable directement à sa saisie sur clavier . Elle subit au niveau du module dialogue un prétraitement avant d'être transférée au module APPRENTISSAGE pour être insérée dans une des trois listes à une position précise .

LE MODULE DE DIALOGUE

- a : il saisit le texte de la règle .
- b : il alloue dynamiquement une zone mémoire qui recevra la règle
- c : il code la règle en entiers (opération de compression)
- d : il met le résultat du codage dans la zone mémoire allouée . La règle est ainsi désignée par une préférence de type pointeur qui contient son adresse . Schématiquement on a :

```
-----  
I I I      I I      I I      I I  
Q -----> I I I      I I      I I      I I  
-----  
c1 11  c2 12    c3 13  c4 14
```

- e : si cette règle n'est pas conforme à la syntaxe imposée , le module de dialogue détruit la règle en libérant la zone de mémoire allouée . Il affiche ensuite un message d'erreur . Si cette règle est validée , elle est transférée au module d'apprentissage pour être insérée .

MODULE D'APPRENTISSAGE

- a : il insère la règle introduite dans la liste appropriée . Si la partie CONCLUSION de la règle :
 - . ne contient aucun FNE , alors la règle est insérée dans la liste 0 (première liste)
 - . contient un FNE , alors la règle est insérée dans la liste 1 (deuxième liste)
 - . contient plus d'un FNE , alors la règle est insérée dans la liste 2 (troisième liste)

b : La méthode classique consiste à insérer la règle en début ou en fin de liste . Nous avons rejeté ce choix non ordonné . Il est préférable de trier les règles en se basant sur la valeur de C1 (partie FNE de la prémisses de la règle) . Ceci permet de minimiser le temps nécessaire à l'étape de filtrage . La procédure AJOUTE réalise cela . Cette procédure compare le filtre C1 de la nouvelle règle avec les filtres des règles déjà présentes dans la B.R , par balayage de cette dernière .

Plusieurs cas peuvent se présenter :

1 - Si la liste est vide , elle appelle la procédure ALLONGE qui insère directement la règle .

2 - Si $C1(1 \text{ ère règle}) > C1(\text{ nouvelle règle})$, on insère la nouvelle règle en début de liste .

3 - Si tous les éléments ont leur $C1 < C1(\text{ nouvelle règle})$, on insère la règle en fin de liste .

4 - Si la situation suivante se présente :

$C1(\text{ REGLE } K) < C1(\text{ NOUVELLE REGLE})$

$C1(\text{ REGLE } K+1) > C1(\text{ NOUVELLE REGLE})$

on insère la règle à la position $K+1$.

B. SUPPRESSION D'UNE REGLE

Le module dialogue permet de saisir le rang (K) de la règle à supprimer . Si le rang est correcte ($K < 1$ OU $K > NO+N1+N2$) alors le module de dialogue lance un message d'erreur . Dans le cas contraire ce nombre est envoyé à la procédure supprime .

PROCEDURE SUPPRIME :

Pour supprimer une règle , il est nécessaire de connaître l'adresse de la règle qui la précède (voir chapitre STRUCTURE DE DONNEE) . La recherche de cette adresse est confiée à la FONCTION ADRESSE . Cette fonction utilise quatre paramètres qui

P1 : pointeur de tête de la liste .

P : pointeur de queue de la liste .

N : nombre d'éléments dans la liste .

K : rang de l'élément à supprimer .

Dans notre cas l'utilisateur introduit une seule donnée pour faire référence à une règle dans une des trois listes . Trois cas peuvent se présenter :

. $1 <= \text{RANG} <= \text{NO}$ alors $K = \text{rang} - 1$, $P1 = A0$, $P = B0$, $N = \text{NO}$

. $\text{NO} < \text{RANG} <= \text{N1}$ alors $K = \text{rang} - \text{NO} - 1$, $P1 = A0$, $P = B0$, $N = \text{N1}$

. $\text{N1} < \text{RANG} <= \text{N2}$ alors $K = \text{rang} - \text{NO} - \text{N1} - 1$, $P1 = A0$, $P = B0$, $N = \text{N2}$

La fonction " ADRESSE " calcule l'adresse de l'element precedent. Cette adresse est utilisée par la procedure OTER pour supprimer effectivement la règle .

La procedure SUPPRIME , libere ensuite l'espace occupé par la règle grace à l'instruction Pascal (DISPOSE).

Le module Dialogue affiche un message qui informe l'utilisateur que la règle a ete detruite .

C. CONSULTATION DE LA BASE DE REGLES

Le module de dialogue permet de saisir le choix de l'utilisateur concernant les sorties desirées (ecran, imprimante, memoire de masse) .Le controle est ensuite passé au module apprentissage dont la partie maitresse est la procedure CONSULTATION (voir listing) . Nous donnons ci dessous l'algorithme générale de fonctionnement de cette procedure .

PROCEDURE CONSULTATION

```

Pour i allant de 1 jusqu'a NO+N1+N2 faire
  Debut
    Q=A0 ( on commence par la première liste )
    si i=NO+1 alors Q=A1 ( passage à la deuxième liste )
    si i=NO+N1+1 alors Q=A2 ( passage à la troisième liste )
    afficher i
    print(Q) ( impression de la règle )
    Q=Q^.PE ( passage à la règle suivante )
  fin
```

PROCEDURE PRINT

Cette procedure affiche le texte de la règle en respectant la syntaxe imposée .Elle restitue donc la règle telle qu'elle à ete introduite par l'utilisateur . Pour cela elle appelle la procedure ECRIT_mot qui effectue une conversion des entiers en chaine de caracteres lisibles par l'utilisateur , grâce à l'utilisation du fichier dictionnaire .

Pour respecter la syntaxe imposée , la procedure :

- fait precéder les FNE du signe '?'
- sépare les faits par ','
- sépare la partie premisses de ma partie conclusion par le signe '='

PROCEDURE ECRIT_NOT

Cette procedure utilise le fichier dictionnaire pour afficher le mot codé par un nombre n .Trois instructions Pascal suffisent pour réaliser cela :

- seek(f,n) (positionnement à l'enregistrement n)
- get(f) (transfère de l'enregistrement à la mémoire centrale)
- write(f^a) (affichage et impression)

III. 2 MODULE DE RAISONNEMENT

INTRODUCTION

Ce module utilise le savoir du système (base de règles) pour :

- . Vérifier une hypothèse (fonctionnement en drainage mixte)
- . Dédire des faits (fonctionnement en drainage avant)
- . Idire des faits (fonctionnement en drainage arrière)

Le module de dialogue, permet de saisir, le texte du problème à résoudre et de le coder en étape de compression ou transformatin en entier. Le résultat est contenu dans les variables vecteurs : Base 1 (FNE) et base 2 (FE) et dans les variables entières L8 1 (nombre de FNE) et L8 2 (nombre de FE)

Le module de dialogue, vérifie si l'opérateur veut utiliser la base de faits contenue dans la mémoire de masse (fichier des faits).

Si la réponse est positive, alors les faits sont insérés dans base 2. Un appel de PURGE 2 permet d'éliminer les faits qui apparaissent plus d'une fois.

Grâce au module de dialogue, l'utilisateur fait le choix des sorties de résultats (écran, imprimante, mémoire auxilliaire), le contrôle est ensuite donné à la procédure DEMONTRE

1. Procédure DEMONTRE

Dans le cas d'un problème à résoudre avec des FNE (fonctionnement en chaînage mixte, ou en chaînage arrière) elle fait appel à la procédure VERIFIE. Au retour elle utilise le module de DIALOGUE, pour informer sur la validité de ses hypothèses

Dans le cas d'un problème avec seulement des FE (fonctionnement en chaînage avant) la procédure insère le filtre C1 (fait non établis) dans la partie prémisse de chaque règle BF comme FNE puis appelle la procédure VERIFIE. En cas de succès elle affiche les nouveaux faits établis, puis insère en nouveau C 1 de la prochaine règle.

2. Procédure VERIFIE

- Elle réalise, les étapes d'évaluation par appel des fonctions "EQUAL" et "CAS" qui détermine l'ensemble des conflits (des règles compatibles avec la BF filtrage sur C1.

- Elle fait appel à sub-vérifie qui effectue un filtrage sur C2 puis réalise l'étape d'exécution par appel de la procédure REMPLACE.

a. Etape d'évaluation

1- Phase de filtrage sur C1 :

Pour chacune des trois listes (listes 0, 1, et 2) la procédure VERIFIE fait appel à la fonction "EQUAL" qui affecte aux variables POINTEURS Q0, Q1 et Q2 les adresses des premières règles compatibles dans liste 0, liste 1, listes 2.

Elle fait appel ensuite à la fonction "CAS" qui affecte aux variables I0, I1 et I2 le nombres de règles compatibles respectivement dans les listes 0, 1, et 2 .

Rmarque : I0, I1, I2 et Q0, Q1, Q2 sont des variables internes à la procédure .

2- Phase de filtrage sur C2 et exécution :

Elle envoie les paramètres (I0, Q0) à SUB-VERIFIE .

Ainsi la recherche de la solution est ordonnée (on commence par les règles les plus prometteuses : pas de FNE à la partie conclusion).

Au retour s'il y a succès, alors on quitte cette procédure (retour à la procédure qui a effectuée l'appel).

S'il y a echec on envoie les paramètres (I1 , Q1) à SUB-VERIFIE (ce sous ensemble de règles remplace un problème par un autre).

Si au retour on a succès on quitte la procédure .

Si echec on envoie les paramètres (I2, Q2) à SUB-VERIFIE (ce sous ensemble de règles remplace un problème par une série de sous problèmes). A ce stade l'ensemble R1 est vide, on quitte la procédure dans les deux cas echec ou succès .

3. Procédure SUB-VERIFIE

a- Elle mémorise grâce à des variables internes la base de faits .

b- Elle mémorise grâce à une variable interne , l'adresse de la règle utilisée. De cette façon en cas de retour arrière, l'étape de restriction est effectuée automatiquement.

c- Elle sélectionne la première règle disponible dans le sous ensemble R1 (la procédure VERIFIE détermine trois sous ensembles R1(0), R1(1) et R2(2) qui correspondent respectivement à la liste 0, la liste 1 et la liste 2).

d- Elle appelle la procédure REMPLACE qui effectue un filtrage sur C2 . Si le résultat est positif cette même procédure réalise l'étape d'exécution .

e- Au retour de la procédure REMPLACE , si la base de problème est vide (succès), elle redonne le contrôle à VERIFIE (procédure APPELLANTE).

f- Si la base de problèmes n'est pas vide , elle appelle la procédure VERIFIE (passage au cycle n + 1). Au retour si on a succès , il y a retour au cycle supérieur , sinon la procédure SUB-VERIFIE :

- Restaure l'ancienne base de faits mémorisée .

- Sélectionne la règle qui suit (étape de restriction) .

- SI aucune règle n'est disponible on a ECHEC et RETOUR ARRIERE vers la procédure VERIFIE , sinon le cycle recommence avec la nouvelle règle depuis l'étape (d) .

4. Procédure REMPLACE

a- PHASE DE FILTRAGE SUR C2

A cette étape la procédure effectue un filtrage sur C2. C.a.d , elle vérifie si tous les FE de la prémisses font partie de la base de règle .

exemple : si la SF contient :

? ALIMENTATION, INT DEFECTUEUSE, écran sombre, pas de son,

une règle telle que :

? ALIMENTATION, ECRAN SOMBRE=.....

sera sélectionnée alors que la règle :

? ALIMENTATION, SON NORMALE=.....

b-PHASE DE RESOLUTION DES CONFLITS

A cette étape , si la phase de filtrage a été concluante, la procédure retient cette règle pour la phase de filtrage. Sinon , l'ensemble R1 est diminué de la règle non compatible et le contrôle est redonné à sub-verify .

c-PHASE D'EXECUTION

FUSION DES FAITS ETABLIS

La procédure insère les FE de la partie conclusion de la règle dans la base de faits établis . Elle appelle la procédure PURGE2 qui élimine les répétitions de FE .

. FUSION DES FAITS NON ETABLIS

La procédure insère les FNE de la partie conclusion de la règle dans la base des FNE . Elle appelle ensuite la procédure PURGE2 élimine toutes les répétitions de FNE .

. DESTRUCTION DES FNE DE LA BFE QUI EXISTE DANS LA BASE DES FE
A cette étape , la procédure élimine tous les FNE de la BF qui existe déjà dans la BFE .

La phase d'exécution est terminée à ce stade . Il y a retour vers la procédure sub-verifiée .

5. PROCEDURE PURGE2

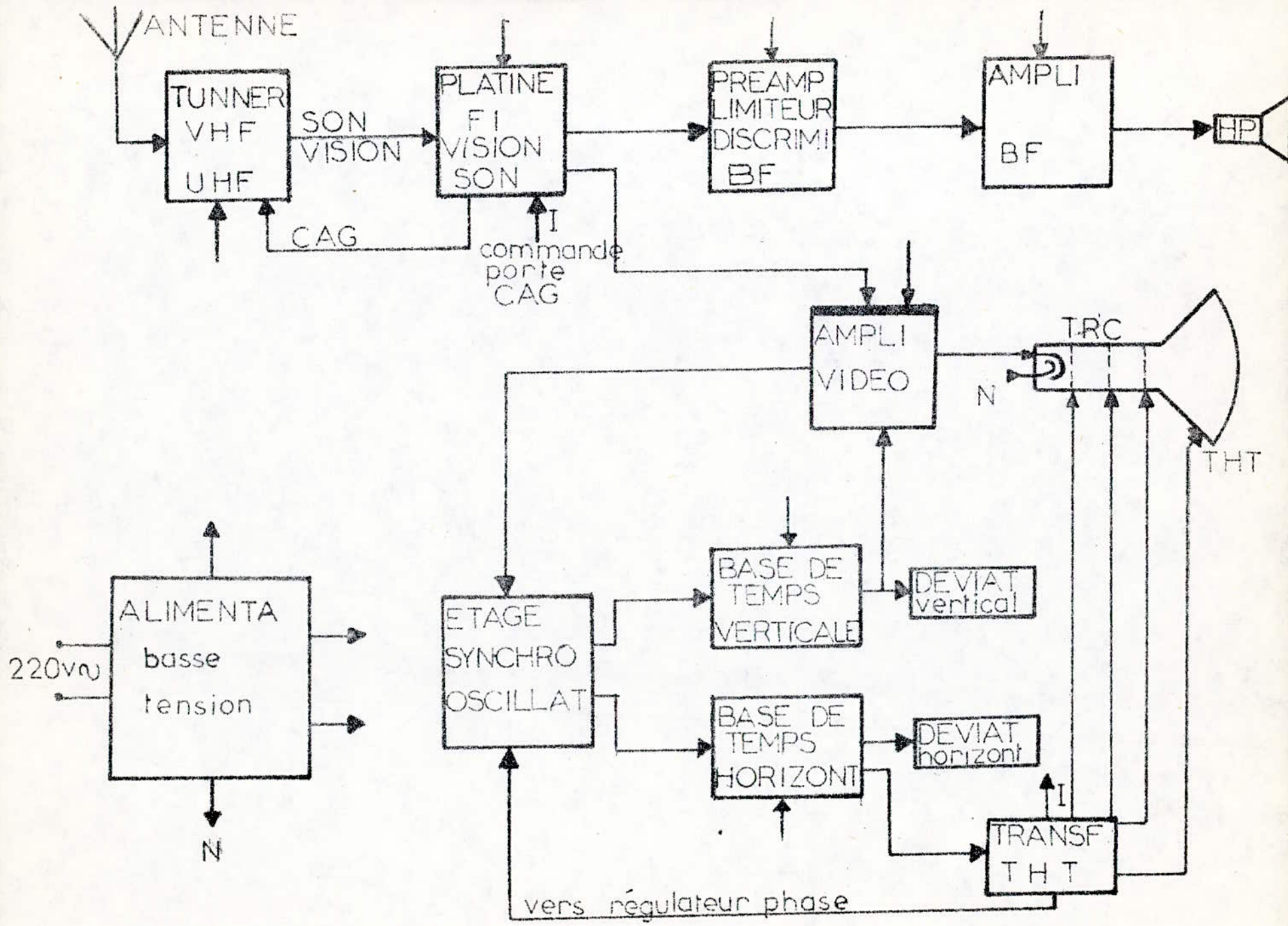
Cette procédure a pour rôle celui d'éliminer les répétitions de faits de faits afin de minimiser le temps d'exécution .

. AU NIVEAU DE LA SAISIE DE LA BF

Si l'utilisateur introduit le même fait plus d'une fois , ceci n'affecte pas les résultats au niveau du module raisonnement . Toutefois , le temps d'exécution sera plus long et notamment au niveau de la phase de filtrage et de la phase d'exécution .

. AU NIVEAU DE L'ETAPE D'EXECUTION

Quant on insère les FE dans la BFE ou les FNE dans la BFNE , des faits peuvent apparaître plus d'une fois . Ceci est due au fait que les champs C3 ET C4 de la règle peuvent contenir des faits qui existent respectivement dans la BFE et la BFNE .



..schema synoptique

III-1 INTRODUCTION

La majorité des réparateurs de télévisions ont accédé à la profession sans connaissances approfondies de cette technique. Ils possèdent, en général, juste quelques rudiments de formation au départ, puis accumulent un certain savoir par la pratique quotidienne, se limitant dans la plus part des cas au remplacement de l'élément défectueux. Les causes véritables de pannes sont souvent ignorées et l'équipement dérisoire des ateliers ne permet aucunement d'en faire les investigations nécessaires. Dans la majorité des cas, lors des opérations de réparation, les hypothèses hatives ne donnent aucun résultat.

Nous nous proposons dans ce chapitre de ressenser les différentes causes des pannes les plus fréquentes, afin d'élaborer une base de connaissances exploitable par notre moteur d'inférences.

Cette base de connaissances intégrée au moteur d'inférences constituera le noyau du système expert d'aide au diagnostic de pannes de télévision.

III-2 SCHEMA SYNOPTIQUE ET FONCTION
DES DIFFERENTS BLOCS.....

Dans cette partie, seule une description succincte de chaque bloc qui entre dans la constitution du téléviseur sera donnée.

Le téléviseur est constitué de :

a- L'antenne :

Elle permet de collecter des signaux électromagnétiques en provenance de l'émetteur.

b- Le tuner ou "synthéonisateur" :

Le tuner constitue le système de changement de fréquence. A la sortie du tuner la fréquence du signal utile est changée en fréquence intermédiaire.

c- Etage amplificateur son et vision :

Cet étage, en plus de l'amplification permet la séparation vidéo et son.

d- L'amplificateur basses fréquences de puissance :

Le signal du son est amplifié par cet étage qui délivrera un signal destiné à exciter le haut parleur.

e- Platine synchro - oscillateur :

Elle permet la séparation des tops de synchronisation. Elle sépare aussi les tops de synchronisation lignes et trames qui

seront ensuite véhiculés vers les bases de temps correspondantes .

f- Base de temps horizontale :

Elle génère des signaux en dents de scie ,qui excitent la bobine de déviation horizontale , et permet l'obtention de la THT .

g- Bases de temps verticale :

Elle génère des signaux en dents de scie qui excitent la bobine de déviation verticale .

h- Alimentation :

Le récepteur TV est équipé d'une alimentation basse tension qui délivre, à partir du secteur ,les différentes tensions nécessaires aux différents étages .

III.3 ETUDE DES PANNES ENGENDREES PAR CHAQUE BLOC.....

1-ALIMENTATION

Un défaut, dans l'alimentation, entraîne automatiquement un arrêt ou un mauvais fonctionnement de tous les étages du téléviseur : les principales causes de cette panne sont :

a,b- Une des diodes du pont redresseur se trouve défectueuse, le secondaire du transformateur court-circuité, tous les étages ne sont plus alimentés et cela se traduit par un écran sombre et une absence de son .

c- Le condensateur de filtrage présente des fuites importantes de courant .Ce défaut se manifeste au niveau de l'écran par une image retrécie en hauteur et en largeur, il est à noter que ce phénomène peut être engendré par une tension de secteur très faible .

d- Si un des fusibles se trouve détérioré alors on peut avoir les cas suivants :

- Absence de tension au niveau du secondaire ,cela se manifeste par un écran sombre et une absence totale de son .

- Le son existe mais l'écran reste sombre .

2-TUNER

Dans cette partie, on se limitera au tuner V H F et aux conséquences de son mauvais fonctionnement .

a- Si les diodes V A R I C A P sont défectueuses , l'étage intermédiaire est hors d'usage, on ne reçoit ni son ni image.

b- Si l'oscillateur local est détérioré , à la sortie du tuner, on récupère le signal reçu par l'antenne mais avec une fréquence supérieure à la fréquence intermédiaire . Cela se manifeste par l'absence de l'image et du son .Le meme résultat serai observé si l'oscillateur local était mal polarisé .

c- Si l'oscillateur local présente une dérive thermique ,après un certain temps ,l'oscillateur chauffe puis le son et l'image se trouvent noyés dans des parasites .

d- La défectiosité ou la mauvaise polarisation de l'étage mélangeur se traduit par l'absence totale du signal contenant l'information son et image .

e- Si la tension de C A G est nulle , ce qui veut dire que l'étage H F n'est pas polarisé , le signal est bloqué au niveau de l'entrée et par conséquent il y a absence du son et de l'image .

f- Si la tension de C A G est soit trop faible , soit trop élevée ; le point de fonctionnement de l'étage H F se trouve déplacé , cela engendre une amplification insuffisante , l'image sera à peine visible et le son à peine audible .

g- Si l'une des diodes, du filtre sélecteur de bandes, se trouve détériorée , le filtre est court-circuité alors le récepteur ne reçoit plus les émissions . Cela est caractérisé par l'absence du son et de l'image .

h- Si l'une des diodes V A R I C A P est défectueuse , l'étage intermédiaire est mis hors d'usage . On ne reçoit plus ni son ni image .

3- PLATINE FI

a- Si le préamplificateur est détérioré , il n'y a ni son ni image

b- Si le circuit oscillant du limiteur se trouve défectueux, on aura uniquement l'image .

c- Si le signal qui traverse le limiteur est faible, le son sera perturbé .

d- La défectuosité de la platine synchro-oscillateur se traduit par l'absence du son et de l'image .

e- Si le filtre de sortie vidéo est défectueux , l'image reçue est perturbée par des parasites .

f- Si le déphaseur est détérioré , il n'y a pas d'image .

g- Si la porte de commande se trouve bloquée ou détériorée , la C A G restera bloquée , l'étage H F n'est plus polarisé, il se manifeste une absence de son et d'image .

4-ETAGE B F

- a- Si l'un des circuits intégrés est mal polarisé le signal BF n'est pas traité correctement, le son se trouve complètement noyé dans du souffle.
- b- Si l'amplificateur limiteur fonctionne mal, les parasites qui accompagnent le signal BF, ne sont pas éliminés cela se traduit par un son à peine audible.
- c- Si le potentiomètre du volume est détérioré alors il y a absence de son.
- d- Si l'un des fils qui relie l'étage BF au haut parleur est coupé, il y a absence de son.
- e- Si le HP est défectueux, il n'y a pas de son.
- f- Si le démodulateur à coïncidence est détérioré alors le son est anormal.

5-AMPLIFICATEUR VIDEO

- a- La défectuosité de l'un des transistors qui équipent l'étage vidéo se traduit par l'absence de l'image.
- b- Si les bobines qui déterminent la bande de fréquence voient leurs valeurs diminuer avec le temps, la bande passante diminuera, l'image ne sera plus nette à cause de de l'absence des fréquences élevées qui traduisent les fins détails de l'image.
- c- Si les paramètres des transistors diminuent avec le temps l'étage ne transmet plus toutes les fréquences et l'image manque de définition (l'image est floue).

6-ETAGE SYNCHRO-OSCILLATEUR

- a- Si l'image paraît déchirée, cela est dû à la valeur de la fréquence horizontale qui est différente de 15625 Hertz (cas du 625 lignes). On peut rendre l'image normale en agissant minutieusement sur le potentiomètre de réglage de la fréquence horizontale.
- b- Si le déphasage, entre l'impulsion de synchronisation et le signal dent de scie, délivré par la base de temps, est grand alors il y a apparition au niveau de l'écran d'une

bande noire verticale à droite ou à gauche de l'écran . Cette anomalie peut être éliminée en agissant sur le potentiomètre de réglage de la fréquence verticale .

c- Si le séparateur ne fonctionne pas, il n'y a pas de tops de synchronisation .

d- Si le séparateur vertical est défectueux, il n'y a pas de synchronisation verticale, Cela se traduit au niveau de l'écran par l'apparition d'une ligne horizontale brillante .

e- Si le comparateur de phase est détérioré et si l'oscillateur présente des glissements, la synchronisation devient irrégulière et l'image donnée par le récepteur apparaît déchirée .

f- Si l'étage de sortie horizontal ne fonctionne pas, il n'y a pas de déviation horizontale et l'écran reste sombre .

7-BALAYAGE VERTICAL

a- Si le circuit de correction en " S " se trouve défectueux l'image sera déformée et étirée suivant la hauteur .

b- Si le potentiomètre de stabilité verticale ne permet pas d'arrêter le défilement de l'image alors l'oscillateur trame n'est pas à la bonne fréquence .

c- Si le potentiomètre permet d'arrêter le défilement de l'image et que celle-ci reste instable, alors il y a absence de tops de synchronisation .

d- Si la tension de polarisation est faible alors l'image se trouve "tessée" dans le bas de l'écran .

8-TRANSFORMATEUR DE THT

Important : S'il n'y a pas de balayage horizontal, le transformateur de THT ne délivrera aucune tension .

a- Si les tensions délivrées par le transformateur sont faibles, cela se traduit par une image terne, rétrécie en largeur et en longueur .

b- Si la THT est coupée par suite d'un court-circuit partiel l'écran restera sombre .

c- Si la THT est faible, l'écran ne s'éclaircira que très faiblement .

d- Si la tension de l'une des grilles , est nulle le faisceau électronique est perturbé par la divergence des électrons , et cela entraîne une image floue .

e- Si le téton de THT présente un mauvais contact , il se produira un léger amorçage qui créera des bandes verticales plus ou moins larges sur l'écran, du côté où se trouve le téton.

9-BALAYAGE HORIZONTAL

a- Si le circuit driver est détérioré , il n'y aura pas de transmission des tops de synchronisation , cela va engendrer une absence de l'image et l'écran restera sombre .

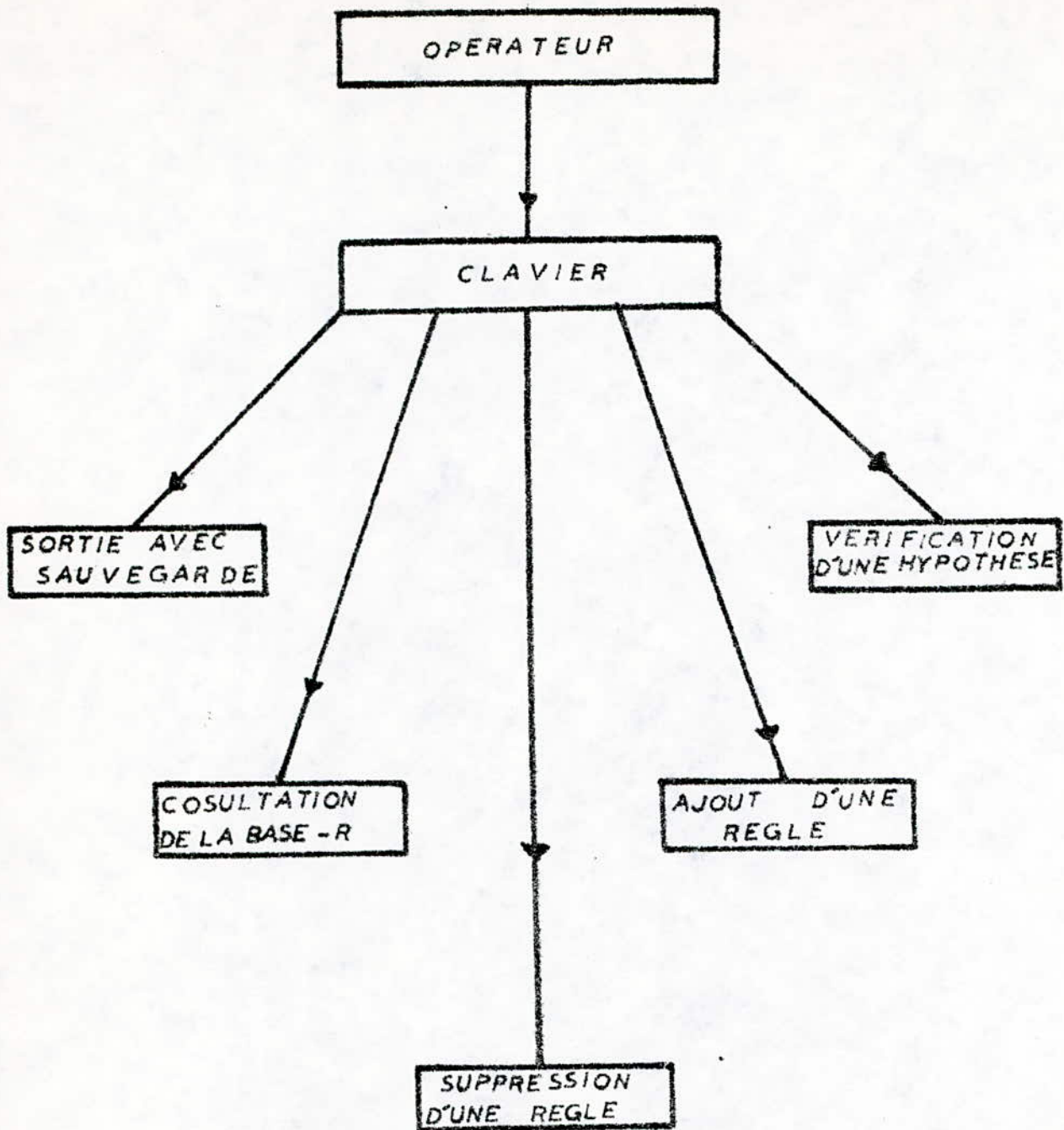
b- Si le circuit de linéarité est mauvais , l'image sera tassée au milieu de l'écran et étirée sur les cotés, suivant la largeur .

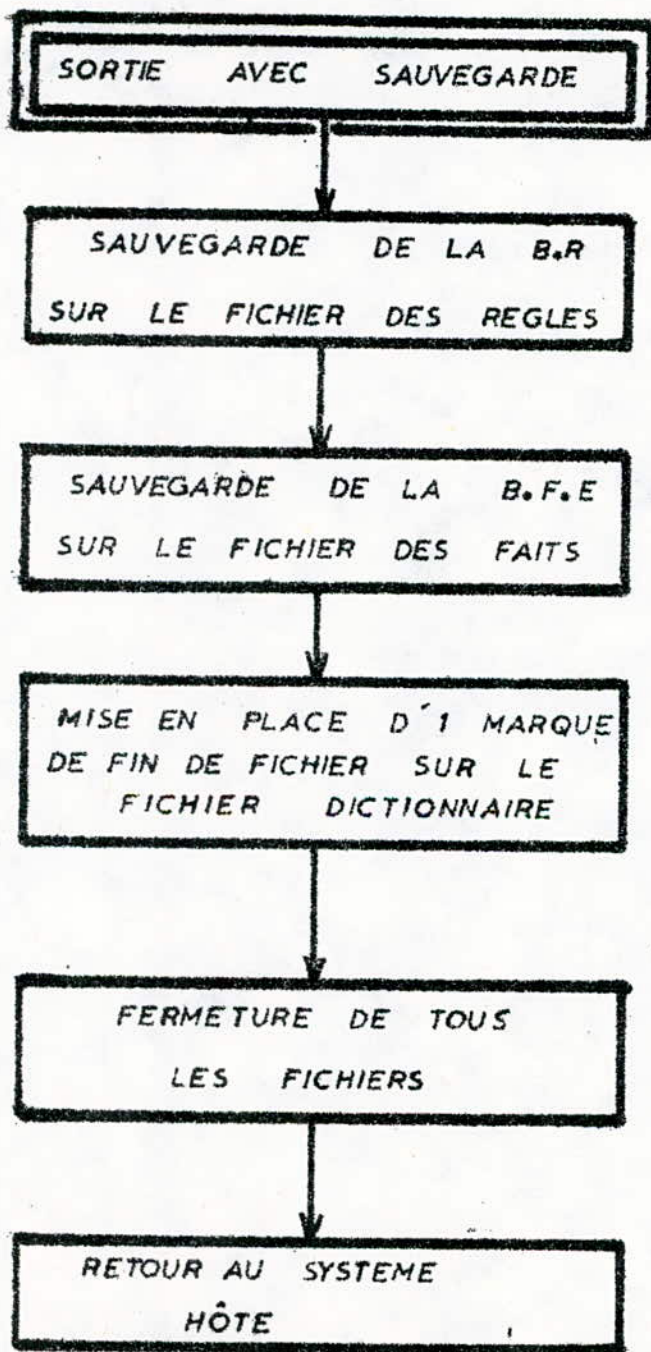
c- Si le condensateur de récupération présente des fuites importantes , cela se traduit au niveau de l'écran par des ondulations sur le côté gauche de l'écran .

d- Si la tension qui alimente l'étage final est très élevée , l'image augmentera en largeur .

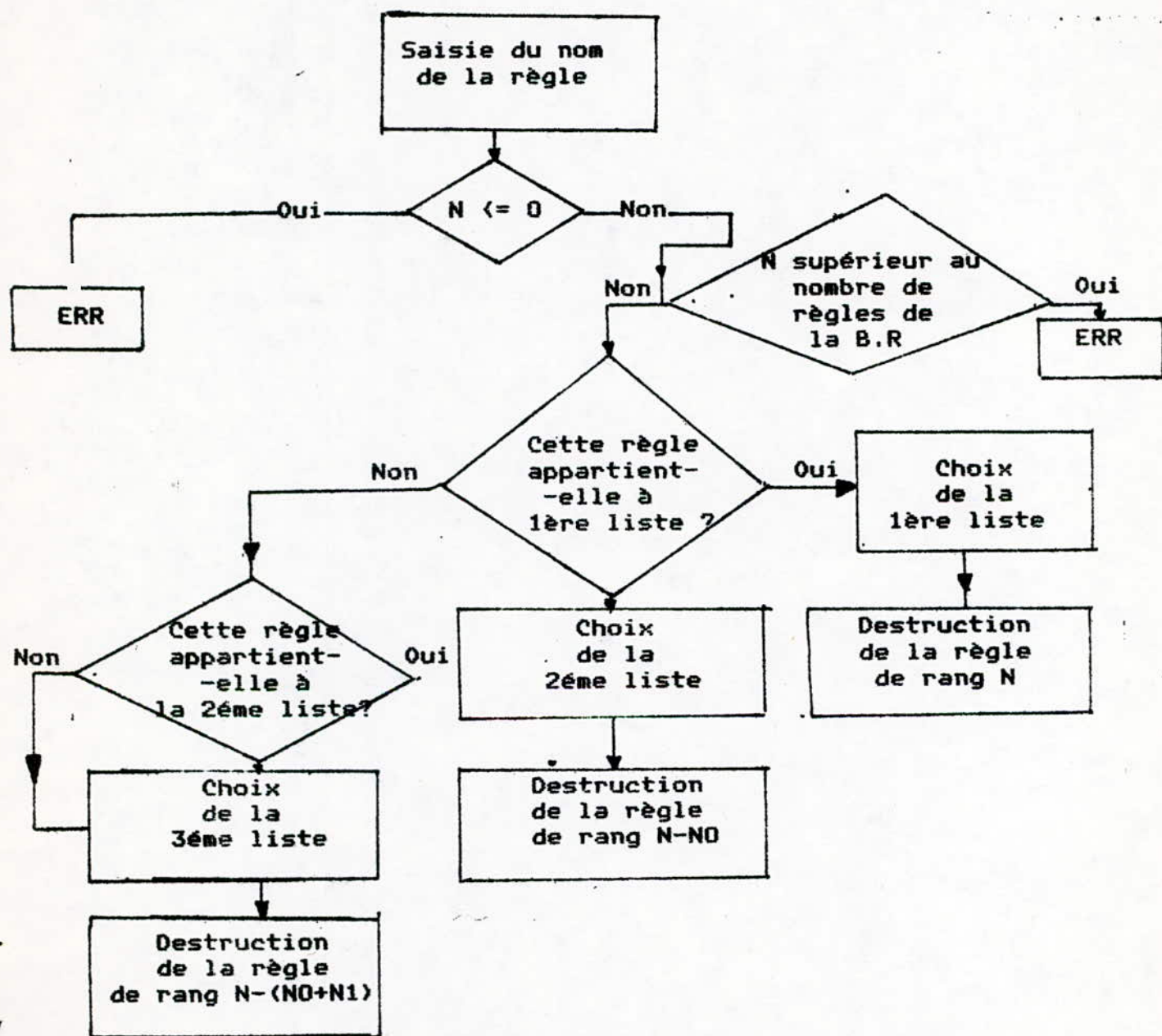
e- Si le condensateur de récupération présente un mauvais contact , on aura une image floue et pale par intermittence .

**
**
**
**
** ALGORITHME GENERAL
** DE FONCTIONNEMENT DU
** MOTEUR D' INFERENCES
**
**





SUPPRESSION D'UNE REGLE



NO : Longueur de la 1ère liste.
 N1 : Longueur de la 2ème liste.
 N2 : Longueur de la 3ème liste.

Remarque: A l'affichage les règles sont numérotées globalement et ne sont pas séparées en sous-listes .

*** CONSULTATION *** DE LA *** B.R ***

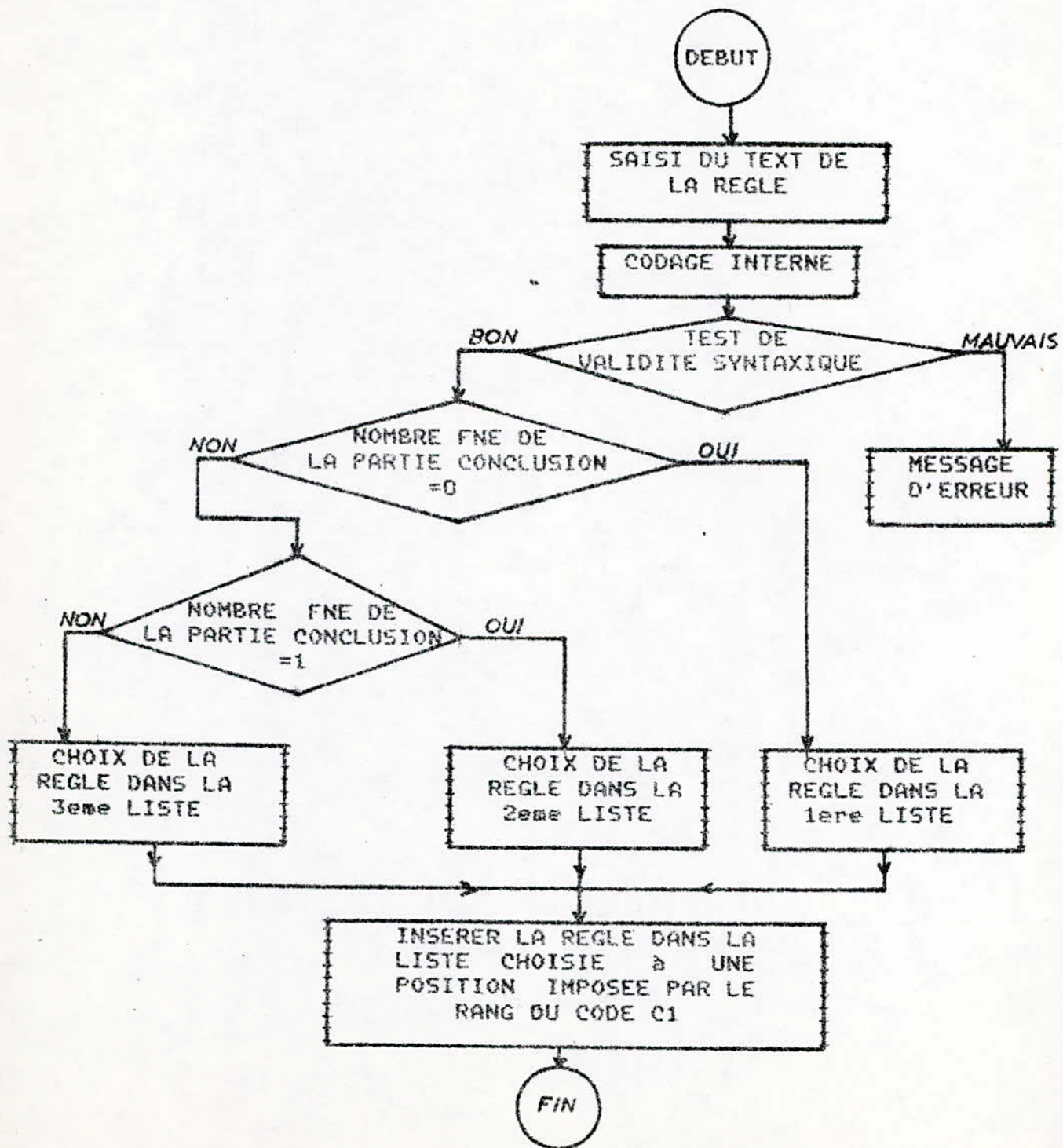
BASE DE REGLE SOUS
CODAGE INTERNE

* DECODAGE *

- . à chaque code numerique associe, grace au dictionnaire,
le mot qui lui correspond
- . Mettre un ' ? ' avant chaque fait non etabli
- . Mettre une virgule entre les faits successifs
- . Mettre ' = ' entre la partie premisses et la partie conclusion

- . Affichage la 1° REGLE
- . Passage à la regle suivante jusqu'a la fin de B.R

AJOUT D'UNE REGLE

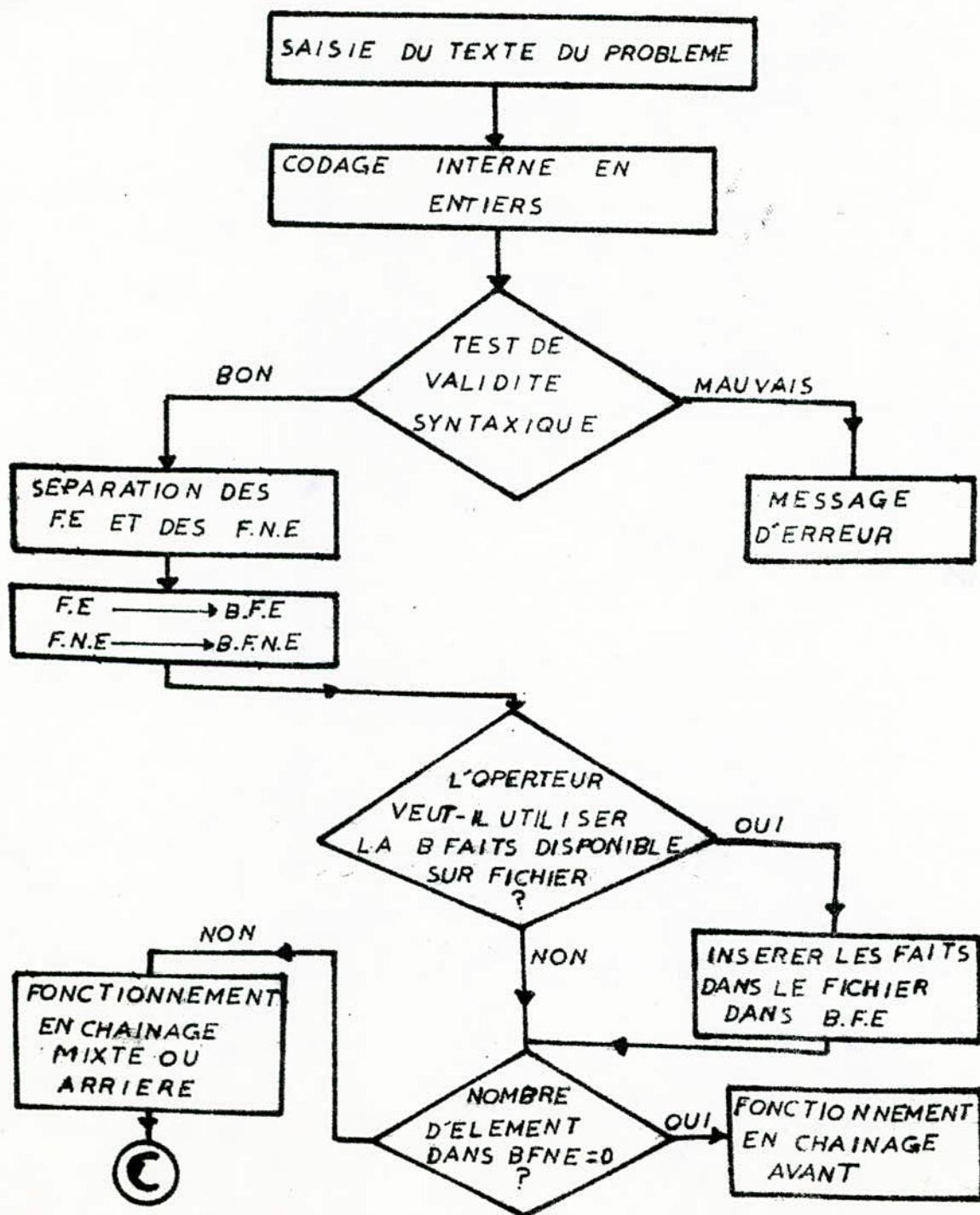


F.E : FAIT-ETABLI

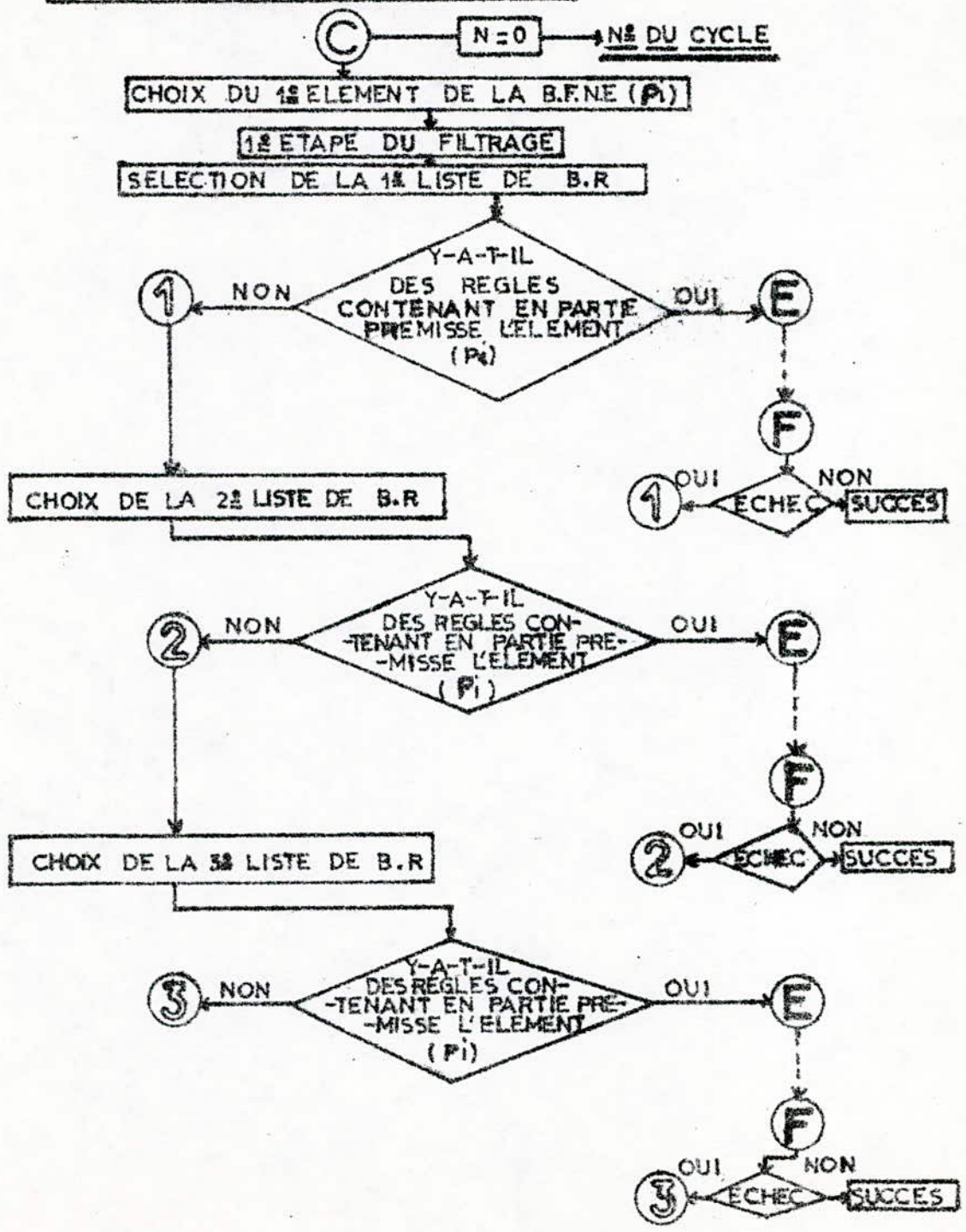
B.F.E:BASE DES FAIT ETABLI

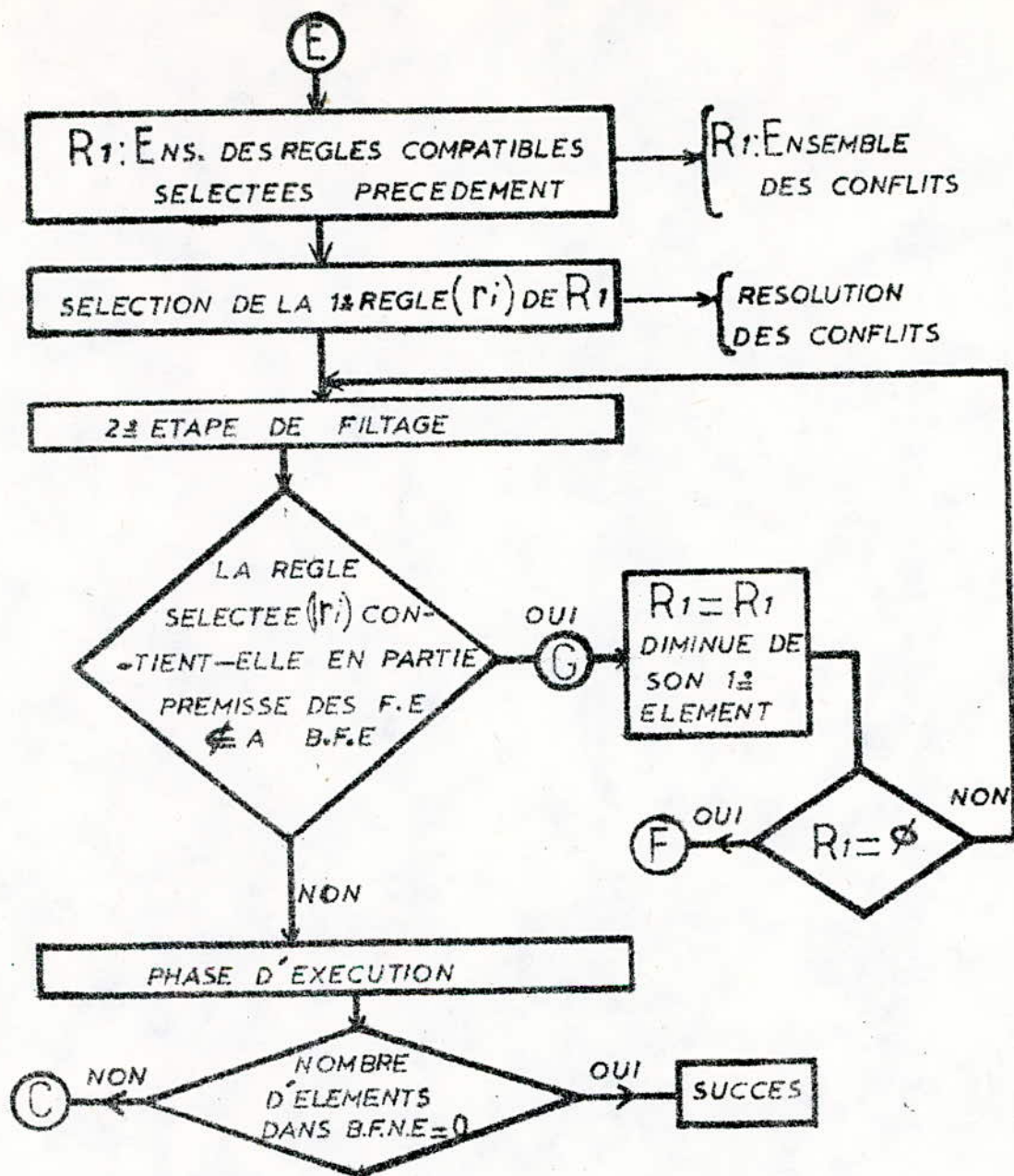
F.N.E: FAIT NON ETABLI

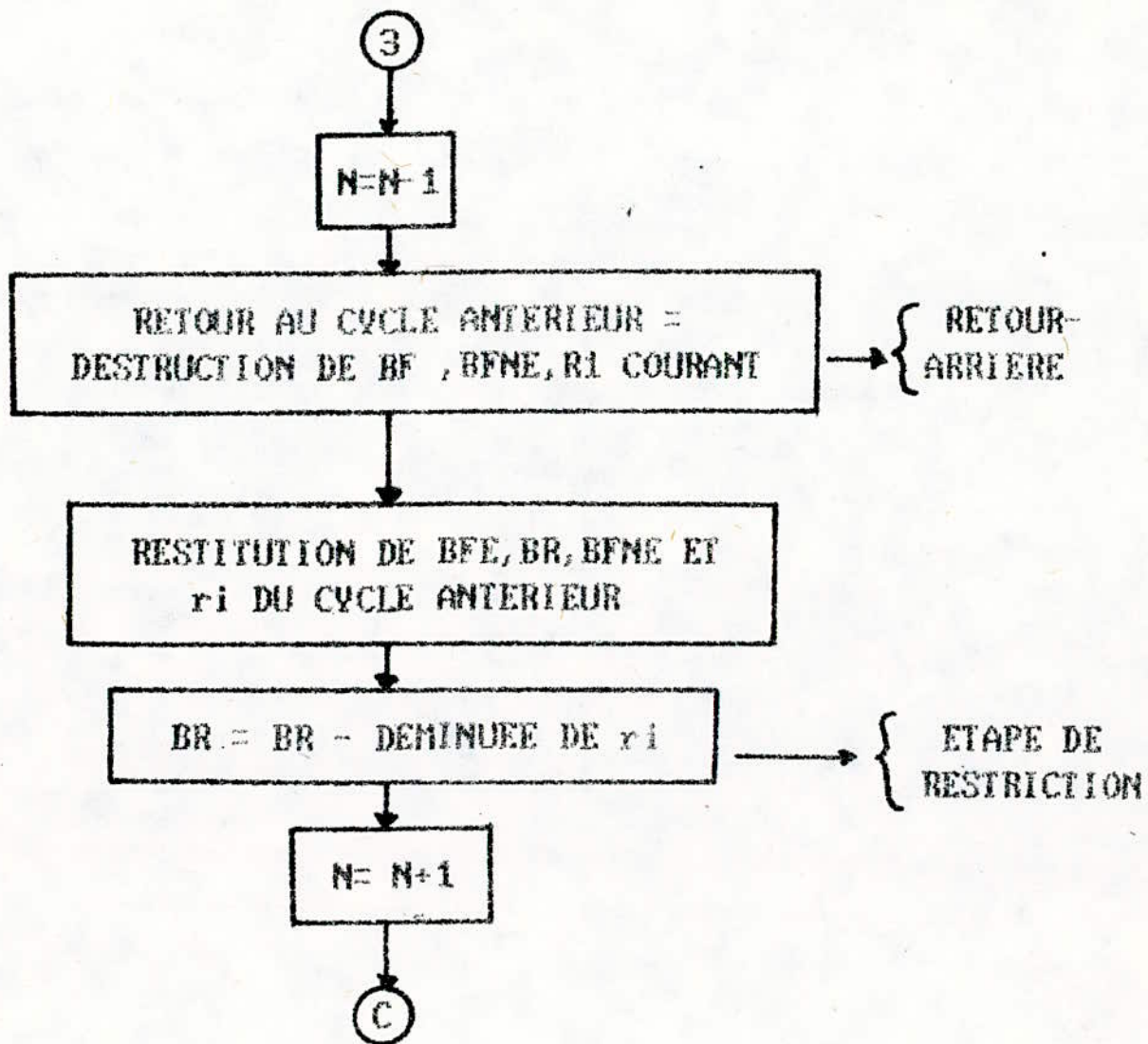
BFNE:BASE DES FAIT NON ETABLI

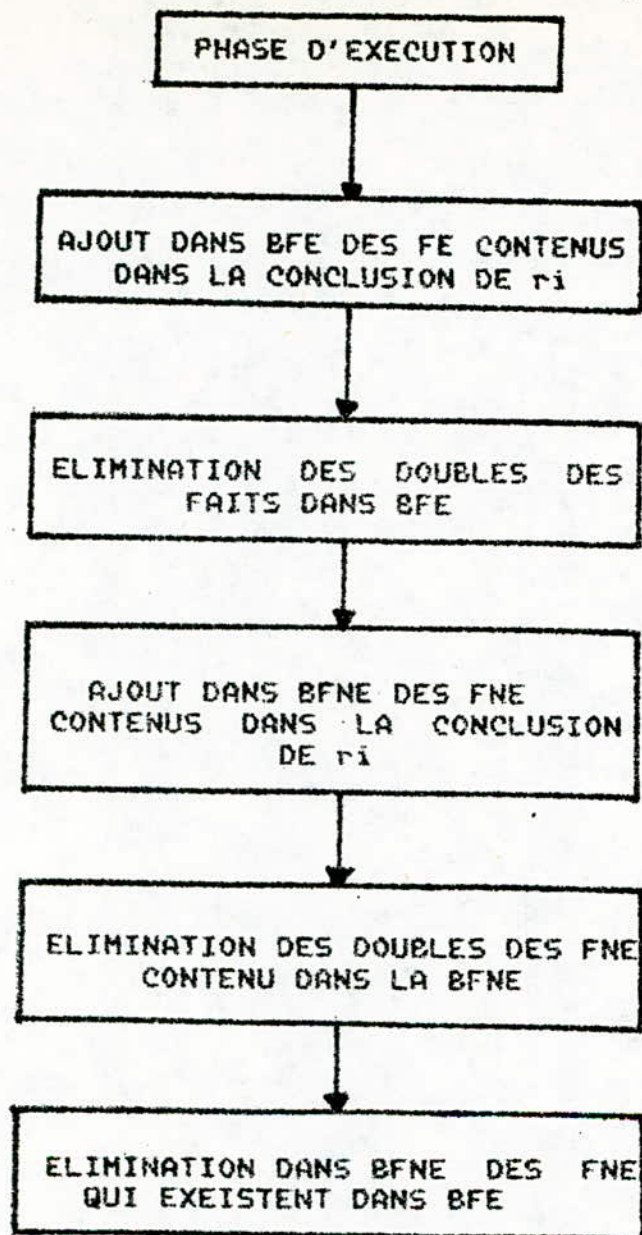


FONCTIONNEMENT EN CHAINAGE-MIXTE

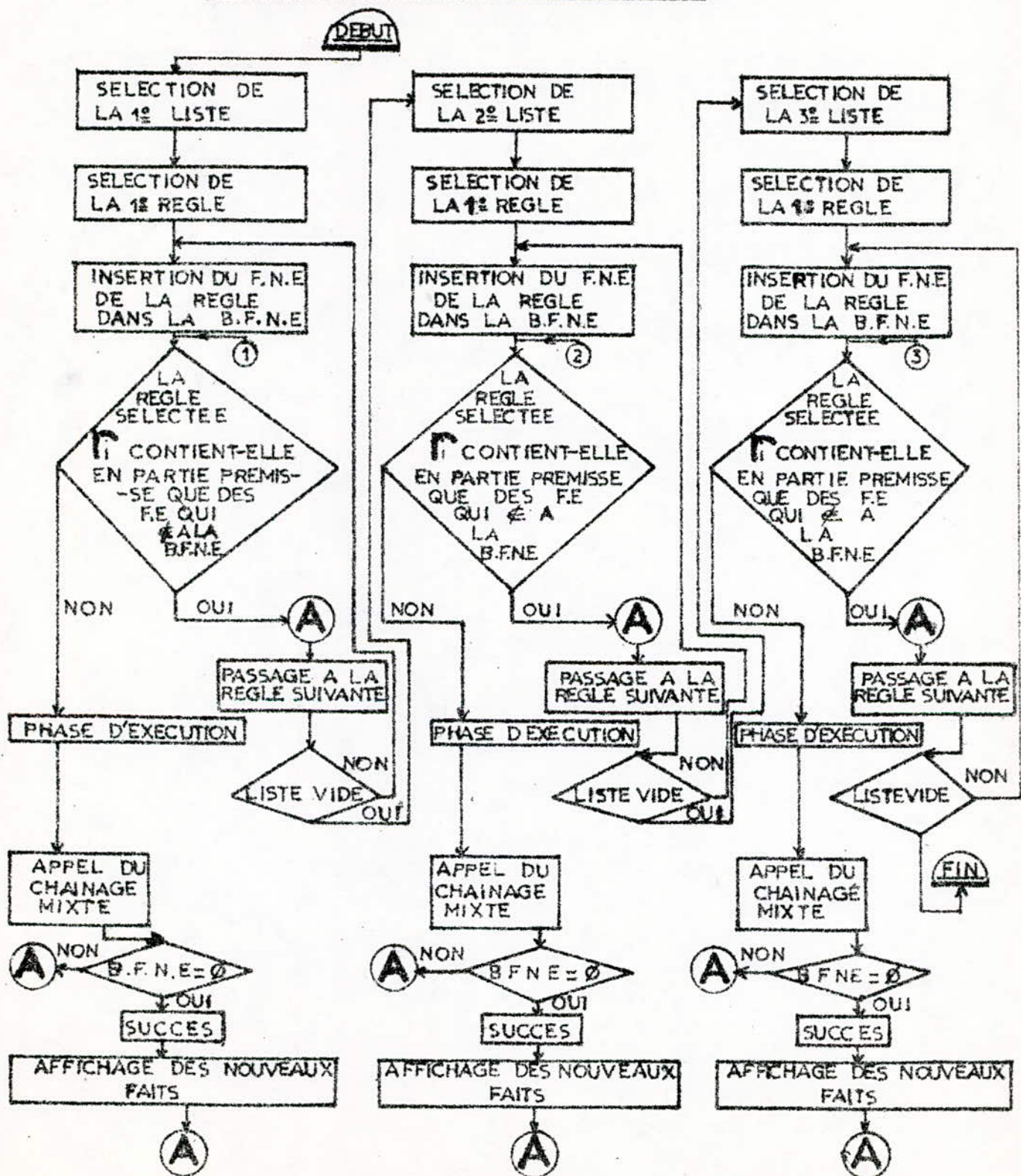








FONCTIONNEMENT EN CHAINAGE AVANT



***** CHAPITRE 5 *****

** IMPLEMENTATION D'UNE **
** BASE DE CONNAISSANCES **
** RELATIVE AP PANNES DE **
** TELEVISION **

V.1 METHODE.....

72

V.2 LISTES DES SYMPTOMES.....

55

IMPLEMENTATION D'UNE BASE DE CONNAISSANCES RELATIVES AUX PANNES DE TELEVISION.....

V.1 METHODE

La réalisation du système expert d'aide au diagnostic des pannes de télévision, passe par deux étapes.

- écriture du moteur d'inferences.
- implementation de la base de connaissances relative aux pannes de television.

nous avons recensé les pannes les plus fréquentes dans le domaine de récepteurs T.V (voir chapitre IV).

Il s'agit de coder ces informations et d'organiser une base de connaissances (dans ce domaine), exploitable par notre moteur d'inferences.

Ceci se fait par création d'un fichier contenant les règles recensées dans le format et la syntaxe imposés par le module de dialogue.

. Soit la règle :

si il n'y pas de son et s'il n'y a pas d'image et si l'ecran est sombre alors verifier le fusible et le secteur et verifier le pont redresseur.

pour introduire cette règle en respectant la syntaxe nous écrivons :

NI SON,NI IMAGE,ECRAN SOMBRE=FUSIBLE+SECTEUR,PONT REDRESSEUR.

. Si l'utilisateur rentre dans la base de faits établis :

NI SON ,NI IMAGE,ECRAN SOMBRE

Le moteur d'inférence répond

JE DEDUIS :

FUSIBLE+SECTEUR
PONT REDRESSEUR

cette règle est rédigée pour permettre un fonctionnement en chainage avant.

Nous avons rédigé d'autre règles pour permettre un fonctionnement en chainage mixte et en chainage arrière . Ceci permet de résoudre des problèmes et de faire des inductions .

V.2 LISTES DES SYMPTOMES PERMETTANT D'IDENTIFIER LES ETAGES DEFECTUEUX

- 1- Ni son ,ni image ,Ecran sombre.....
- 2- Ni son ,ni image ,Ecran éclairé.....
- 3- Son anormal ,image correcte.....
- 4- Son brouillé ,image brouillée.....
- 5- Son normal ,Ecran sombre.....
- 6- Son normal ,pas d'image ,Ecran éclairé.....
- 7- L'ecran présente une bande noire sur le coté gauche.....
- 8- Image réduite en largeur.....
- 9- Image concentrée au centre et étalée suivant la largeur.....
- 10- Image tassée dans le bas de l'ecran.....
- 11- Image ondulée.....
- 12- Image tassée horizontalement (ou dilatée).....
- 13- L'image se retrécit en largeur.....
- 14- Bandes noires verticales.....
- 15- Instabilité horizontale.....
- 16- Ligne horizontale pleine.....
- 17- L'image présente une bande noire au bas de l'ecran.....
- 18- Image réduite en largeur.....
- 19- Image concentrée au centre et étirée suivant la hauteur.....
- 20- Déformation des lignes horizontales.....
- 21- Image étalée verticalement.....
- 22- Le haut de l'image est tassé.....
- 23- Image floue.....
- 24- L'image manque de définition.....
- 25- Instabilité verticale.....
- 26- L'image manque de concentration.....
- 27- Image retrécie en hauteur et en largeur.....
- 28- Image peu lumineuse (terne).....
- 29- Image peu contrastée.....
- 30- Image floue manquant de netteté.....
- 31- Image fantôme (Echo).....
- 32- Retour de trames visible.....
- 33- Effet de brouillard sur l'ecran.....
- 34- Variations périodiques de luminosité.....
- 35- Moirage de l'ecran.....
- 36- La stabilité ne tient pas dans le temps.....
- 37- Image trop lumineuse.....
- 38- Image en forme de trapèze.....
- 39- Déformation en tonneau ou en coussin.....

LES ATTRIBUTS GRAPHIQUES ANSI

- ESC [2J efface l'écran (le curseur garde sa position initial)
- ESC [J efface la partie inferieure de l'écran depuis la position du curseur .
- ESC [l1;l2r ouvre une fenêtre de la ligne l1 à la ligne l2 dans laquelle s'effectue le scrolling (l1 < l2-1) .
- ESC [Y;Xh positionne le curseur à la ligne Y et la colonne X de l'écran .

-
- ESC [0m écriture normal .
- ESC [1m écriture sur fond blanc .
- ESC [4m écriture soulignée .
- ESC [5m écriture clignotante .
- ESC [7m écriture en inverse vidéo .

-
- ESC #6 écriture en double largeur .
- ESC #5 écriture en simple largeur .
- ESC #3 écriture de la partie superieur d'un caractère .
- ESC #4 écriture de la partie inferieur d'un caractère .

-
- ESC [7h écran avec fond blanc .
- ESC [7l écran avec fond noir .

Remarque : cette liste n'est pas limitative : d'autre codes existent . Nous n'avons présenter que les codes que nous avons utilisé .

G L O S S A I R E

Quelques termes importants en intelligence artificielle et en informatique utilisés dans cette thèse :

Algorithme	Description pas à pas d'une solution d'un problème. Transcrit dans un certain langage, un algorithme donne lieu à un programme.
Base de connaissances	Partie d'un système expert contenant les faits et inférences valides dans domaine de compétence du système.
Base de données	Ensemble de faits d'un domaine.
Heuristique	Règles formelles de raisonnement, définies comme "l'art de raisonner" par G. Polya.
Lisp	Langage le plus populaire en I.A.
Moteur d'inférences	Stratégie de raisonnement permettant d'utiliser les connaissances d'un domaine pour résoudre les problèmes.
"Pattern-matching"	Processus par lequel une solution à analyser satisfait un ensemble de critères caractérisant une solution type. Une phrase en langage naturel par rapport à une grammaire est un exemple; un groupe de symptômes vis-à-vis d'une maladie en est un autre. La satisfaction est souvent partielle.
Prolog	Deuxième langage le plus populaire en intelligence artificielle.
Représentation symbolique	Abstractions représentant des objets concrets.

G L O S S A I R E

(S U I T E)

**Structures
de données**

Organisation d'un ensemble de données;
vecteurs, tableaux, listes, ect...

Système

Contrairement à l'usage en "théorie des
des systèmes". Système est employé
dans cette thèse comme synonyme de
programme .

C O N C L U S I O N

Nous pensons avoir répondu totalement aux exigences du cahier de charge imposé par notre promoteur du point de vue M.I .
Nous avons jugé utile d'apporter les améliorations suivantes :

- . possibilité de fonctionnement en chaînage avant du M.I .
- . possibilité de sortie des résultats sur imprimante .
- . stockage des résultats d'exécution sur la mémoire de masse pour une consultation antérieure .
- . utilisation d'une structure à base d'entiers permettant la réduction d'un facteur de 10 de l'espace mémoire utilisée ainsi que de minimiser le temps d'exécution .
- . structuration des règles en trois listes , ce qui confère à notre moteur d'inférence une propriété supplémentaire : la base de règles est ordonnée .

PERSPECTIVES

- . adaptabilité de notre M.I à la réalisation de système expert dans le domaine désirée
- . possibilité d'extention de notre M.I à l'ordre " 1 "
- . amélioration du langage de dialogue avec la machine .
- . interactivité entre le système et l'utilisateur .

Nous ne terminons pas cette étude sans remercier vivement tous les gens qui ont contribué à l'élaboration de notre travail qui n'est qu'une approche du passionnant domaine de " L'I.A " et plus particulièrement des systèmes experts .

Nous espérons que notre travail sera une amorce au développement de cette discipline à l'ENP .

BIBLIOGRAPHIE

- 1: INTRODUCTION AUX SYSTEMES
EXPERTS
MICHEL GONDRAN
EDITION EYROLLES
- 2: PROGRAMMES D'INTELLIGENCE
ARTIFICIELLE EN BASIC
MG.MONTEIL & R.SCHOMBERG
EDITION EYROLLES
- 3: INTRODUCTION A L'INTELLIGENCE
ARTIFICIELLE SU MICRO ORDINATEUR
MIKE JAMES
EDITION EYROLLES
- 4: L'INTELLIGENCE ARTIFICIELLE
PROMESSES ET REALITES
ALAIN BONNET
INTEREDITIONS
- 5: DICTIONNAIRE DE L'INFORMATIQUE
EDITIONS LAROUSSE
- 6: STRUCTURES DE DONNEES ET
ALGORITHMES FONDAMENTAUX
PROFESSEUR PAIR
ENSIM . NANCY
- 7: PASCAL
TECHNIQUES DE PROGRAMMATION AVANCEES
R.LIGNELET
EDITIONS EYROLLES
- 8: RAPPORT D'ACTIVITE _JUN 85
RAPPORT D'ACTIVITE _MAI 86
K.KASDI
E.N.P
- 9: DEPANNAGE , MISE AU POINT ,
AMMELIORATION DES TUC (NB)

HYPOTHESE INITIALE : ?H,B,C

***** CYCLE 1 *****

ON A 2 REGLES COMPATIBLES

No: 1 ?H,W=Z

No: 2 ?H=?A,L

J'UTILISE LA REGLE : ?H,W=Z
CETTE REGLE -----> NE CONVIENT PAS

J'UTILISE LA REGLE : ?H=?A,L
CETTE REGLE -----> CONVIENT
LA BASE DES FAITS DEVIENT---->>?A,L,B,C

***** CYCLE 2 *****

ON A 1 REGLES COMPATIBLES

No: 1 ?A=?P,C,K

J'UTILISE LA REGLE : ?A=?P,C,K
CETTE REGLE -----> CONVIENT
LA BASE DES FAITS DEVIENT---->>?P,C,K,L,B

***** CYCLE 3 *****

ON A 2 REGLES COMPATIBLES

No: 1 ?P,L=W

No: 2 ?P=?D,?E,B

J'UTILISE LA REGLE : ?P,L=W
CETTE REGLE -----> CONVIENT
LA BASE DES FAITS DEVIENT---->>W,C,K,L,B

***** CYCLE 4 *****

PAS DE FAITS A VERIFIER =====> SUCCES

LISTE DES REGLES

No: 1 ?P, L=W
 No: 2 ?H, W=Z
 No: 3 ?E=?D, M
 No: 4 ?A=?P, C, K
 No: 5 ?H=?A, L
 No: 6 ?P=?D, ?E, B
 No: 7 ?A, B=?D, ?G

LISTE DES MOTS RECONNUS PAR LE SYSTEME

No: 1 P
 No: 2 D
 No: 3 E
 No: 4 B
 No: 5 A
 No: 6 G
 No: 7 C
 No: 8 K
 No: 9 M
 No: 10 H
 No: 11 L
 No: 12 W
 No: 13 Z

HYPOTHESE INITIALE : ?H, B, C

***** CYCLE 1 *****

ON A 2 REGLES COMPATIBLES

No: 1 ?H, W=Z
 No: 2 ?H=?A, L

J'UTILISE LA REGLE : ?H, W=Z
 CETTE REGLE -----> NE CONVIENT PAS

J'UTILISE LA REGLE : ?H=?A, L

CETTE REGLE -----> CONVIENT
 LA BASE DES FAITS DEVIENT--->>?A, L, B, C

***** CYCLE 2 *****

ON A 1 REGLES COMPATIBLES

No: 1 ?A=?P, C, K

CETTE REGLE -----> CONVIENT
LA BASE DES FAITS DEVIENT--->>?P,C,K,L,B

***** CYCLE 3 *****

ON A 2 REGLES COMPATIBLES
No: 1 ?P,L=?W
No: 2 ?P=?D,?E,B

J'UTILISE LA REGLE : ?P,L=?W

CETTE REGLE -----> CONVIENT
LA BASE DES FAITS DEVIENT--->>?W,D,K,L,B

***** CYCLE 4 *****

ON A 0 REGLES COMPATIBLES

RETOUR VERS LE CYCLE : 3

J'UTILISE LA REGLE : ?P=?D,?E,B

CETTE REGLE -----> CONVIENT
LA BASE DES FAITS DEVIENT--->>?D,?E,B,C,K,L

***** CYCLE 4 *****

ON A 0 REGLES COMPATIBLES

RETOUR VERS LE CYCLE : 3

RETOUR VERS LE CYCLE : 2

RETOUR VERS LE CYCLE : 1

RETOUR VERS LE CYCLE : 0

CETTE HYPOTHESE EST NON VERIFIEE