

Tesc
الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

19/90

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

REALISATION D'UN SYSTEME
EXPERT EN TURBO-PROLOG POUR
LA RECONNAISSANCE DE LA PAROLE

Proposé par :

Mr. BOUSSEKSOU

Etudié par :

K. AOUA

Promotion : Septembre 1990

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ECOLE NATIONALE POLYTECHNIQUE
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

REALISATION D'UN SYSTEME
EXPERT EN TURBO-PROLOG POUR
LA RECONNAISSANCE DE LA PAROLE

Proposé par :
Mr. BOUSSEKSOU

Etudié par :
K. AOUA

Promotion : Septembre 1990

REMERCIEMENTS

Je tiens à remercier tous ceux qui ont concouru à la réalisation de cet ouvrage, plus particulièrement mes enseignants et en premier lieu mon promoteur Mr B. BOUSSEKSOU qui m'a beaucoup aidé par ses précieux conseils. Ont droit de citation aussi, ma famille et mes amis qui ont partagé avec moi, jour après jour, péniblement certes, mais patiemment les différents soucis liés à mon travail.

Une fois encore et à tout ce beau monde, je dis MERCI.

DEDICACES

A ma mère avec l'espoir qu'elle en cueillera honneur et
fierté .

Première partie

"Introduction à l'intelligence artificielle et outils logiques."

- 3 - Chapitre I: Buts et concepts fondamentaux de l'intelligence artificielle.
- 11 - Chapitre II: Outils logiques.

II.1-Le calcul des propositions

II.2-Le calcul des prédicats

II.3-Les logiques non classiques

Chapitre III: La représentation des connaissances et des mécanismes de raisonnement.

Définition:

Une connaissance déclarative ne contient pas son mode d'emploi, par contraste, une connaissance procédurale indique explicitement comment elle est utilisées.

III.1-Les règles de productions

III.2-Les réseaux sémantiques

III.3-Les objets structurés.

Deuxième partie

"Théorie des systèmes experts".

Chapitre I: Les systèmes dits "experts".

Chapitre II: Moteurs d'inférences.

1-Types de mécanismes d'inférences.

2-Logique associées aux moteurs.

3-Mécanisme de fonctionnement.

4-Cycle de base d'un moteur d'inférence.

Troisième partie

"Systèmes expert d'aide à la décision dans la reconnaissance de la parole SEDARP"

Chapitre I: Prise d'expertise.

1-Présentation du domaine de la reconnaissance de la parole.

2-Extraction et représentation des connaissances:

a-Extraction

b-Représentation.

Chapitre II: Conception de SEDARP

- 1-Shéma général de SEDARP.
- 2-Structure de données utilisées en TURBO-PROLOG.
 - a-Dictionnaire
 - b-Base de faits
 - c-Base de connaissances
 - d-Trace.
- 3-Interface utilisateur/SEDARP
 - a-Interface d'entrée.
 - b-Interface de sortie.
- 4-Interface expert/SEDARP
 - a-Interface d'entrée
 - b-Interface de sortie.
- 5-Moteurs d'inferences
 - 5.1-Objets trités par le moteur d'inférences
 - 5.2-Conception du moteur d'inférence
 - a-Module selection
 - b-Module filtrage
 - c-Module résolution de conflits
 - d-Module déclenchement d'une règle.

Quatrième partie

Conclusion.

CHAPITRE I

BUTS ET CONCEPTS FONDAMENTAUX DE L'INTELLIGENCE ARTIFICIELLE

1) INTRODUCTION :

L'intelligence artificielle est d'un intérêt interdisciplinaire et d'une importance pratique qui vont croissant. Des personnes d'origines et de professions très diverses découvrent de nouvelles idées et de nouveaux outils dans cette science toute jeune . Des psychologues théoriciens ont développé de nouveaux modèles de l'esprit basés sur les concepts fondamentaux de l'I.A . Des systèmes basés sur les symboles et le traitement de l'information les linguistes sont aussi intéressés par ces notions de base aussi bien que par le travail sur l'I.A, dans l'informatique linguistique dont le but est de fournir des programmes qui puissent réellement comprendre le langage humain. Et les philosophes, en considérant les progrès, les problèmes et le potentiel de ce travail, vers une intelligence non humaine ont parfois trouvé de nouvelles perspectives pour les éternels problèmes de la nature, de l'esprit, et de la connaissance.

Dans d'autres sphères d'activités on approche souvent pour la première fois l'I.A sous la forme d'un système expert qui est utilisé expérimentalement dans son propre domaine (

interprétation de données en chimie, intégration symbolique, diagnostics de maladies infectieuses, études des séquences d'A.D.N , débugage de systèmes informatiques, ingénierie de structure, dessins des processeurs ...).

Avec un coût de l'informatique qui continue de chuter, de nombreuses applications nouvelles deviennent possibles étant donné que pour beaucoup d'entre elles, il n'y a pas de cadre mathématique, de tels domaines appliqueront inévitablement des modèles symboliques et des associations de symboles techniques.

2- QU'EST-CE QUE L'I.A :

L'intelligence artificielle est une discipline qui s'intéresse à des problèmes dont la combinatoire est tellement vaste que la solution optimale (si elle existe) ne peut être atteinte dans des délais raisonnables de temps ou à cause des limitations de places en mémoire (humaine ou mécanique). Le nombre de situations possibles sur ces échiquiers, par exemple, est estimé à 10^{420} , ce qui rend impossible tout examen exhaustif de ces situations ou toute mémorisation de l'ensemble de ces positions avec leurs valeurs associées.

L'I.A est la partie de l'informatique dont le but est d'élaborer des systèmes intelligents, c-à-d, des systèmes qui possèdent les caractéristiques que nous associons avec

l'intelligence dans le comportement humain (le langage de compréhension, l'apprentissage, le raisonnement, le solutionnement de probleme, etc...)

Dire qu'un programme résoud un problème ou prend une décision au vu description d'une certaine situation signifie que le programme trouve lui-même la méthode de résolution à employer grâce à une vaste gamme de mécanismes de raisonnements qui lui ont été incorporés. Ceci est une évolution importante par rapport à l'informatique classique où le raisonnement est fait par l'homme et où l'ordinateur est utilisé en priorité de sa capacité de calcul.

Les systèmes d'I.A expérimentaux comprennent des programmes qui

:

- a) Résolvent des problèmes difficiles en chimie, biologie, géologie, ingénierie et médecine au niveau des spécialistes.
- b) Manipulent des systèmes robotisés pour réaliser des tâches utiles, répétitives avec des moteurs et des systèmes sensitifs.
- c) Répondent à des questions posées de façon simple en anglais (en arabe ou toute autre langue naturelle comme on les nomme).

3 QU'EST-CE QUE L'INTELLIGENCE :

Bien que le sens du mot " artificielle " soit suffisamment clair, le problème avec l'expression I.A est qu'il est (

difficile d'être précis sur le sens du mot " intelligence ". La réponse traditionnelle donnée par les psychologues : "l'intelligence est ce que mesurent les tests d'intelligences ", est totalement dépourvue d'intérêt dans le domaine de l'intelligence informatique. Nous pouvons cependant mettre en relief un certain nombre de critères entrant dans la composition de l'intelligence tels que la capacité d'abstraction ou de généralisation, la capacité de faire des analyses entre différentes situations, la faculté de s'adapter à de nouvelles situations, la faculté de corriger ses erreurs afin d'améliorer ses performances futures, etc... Ces tâches intelligentes peuvent être aussi variées que comprendre des histoires écrites ou dites en langage naturel, jouer aux échecs ou au bridge, résoudre un problème de mathématique ou un puzzle, écrire un poème établir un diagnostic médical, etc...

4)- LES CARACTERISTIQUES D'UN PROGRAMME D'INTELLIGENCE

ARTIFICIELLE :

Les critères sont définis, d'une part par les types des problèmes abordés (ils requièrent une certaine intelligence et on n'en connaît pas de solution en général), d'autre part par les méthodologies employées pour faire face à ces problèmes en s'aidant de ce l'on sait de l'intelligence humaine.

a) Représentation symbolique:

les méthodologies employée pour faire face à ces problèmes en s'aidant de ce l'on sait de l'intelligence humaine.

a) Représentation symbolique:

La première caractéristique des programmes d'I.A. est le fait qu'ils manipulent principalement des symboles autres que numériques.

b) Heuristique:

La deuxième caractéristique des programmes d'I.A. est le fait qu'ils s'attaquent à des problèmes dont on ne connaît pas un algorithme (description de la suite à entreprendre afin d'arriver à une solution) du fait que l'on ne connaît pas d'algorithme, on essaye des "heuristiques" c'est à dire des méthodes informelles non garanties de succès. Une démarche heuristique consiste, face à un problème, à essayer un chemin en gardant la possibilité d'en essayer d'autres si celui qui paraissait prometteur n'a pas conduit rapidement à une solution.

c) Une représentation des connaissances :

Les programmes d'I.A. contrairement aux programmes de statistiques possèdent une représentation des connaissances c-a-d une correspondance entre le monde extérieur et un système symbolique permettant de raisonner. Ces connaissances sont généralement examinables et compréhensibles en "termes humains" car les symboles employés pour la représentation sont très

rarement numériques.

d) Données incomplètes du problème:

Une quatrième caractéristique fondamentale des programmes d'I.A. est leur capacité de fournir une solution, même si toutes les données ne sont pas disponibles au moment de la résolution du problème. Il arrive souvent en médecine qu'un examen de laboratoire n'ait pas encore été fait et qu'une décision soit malgré tout à prendre sans attendre le résultat de cet examen. La conséquence de données incomplètes est simplement que les conclusions seront moins "sûre" ou moins bonnes (éventuellement fausses dans certains cas). Il est fréquent dans le monde réel que l'on doive prendre une décision sans disposer de toutes les données du problème ce qui nous conduit également à tirer des conclusions qui s'avèrent parfois erronées par la suite. Il arrive aussi que la "présence" de données incomplètes soit inhérente au problème à résoudre. C'est le cas des jeux de cartes, le bridge par exemple, ou l'on ne connaît jamais deux des quatre jeux et où l'on doit faire des hypothèses "incertaines" sur la répartition des cartes inconnues en utilisant notamment des probabilités de distribution utilisées par les informations provenant des annonces.

e) Données conflictuelles :

Une cinquième caractéristique de l'I.A. est de s'accommoder de

deuxième explication possible est celle qui nous intéresse ; il y a erreur d'observation et l'un des deux phénomènes A ou B n'est en fait pas présent.

f) Capacité d'apprentissage:

Un autre critère d'intelligence est la faculté d'apprendre, c'est à dire d'améliorer ses performances en tenant compte des erreurs passées. La faculté d'apprentissage est liée à la notion de généralisation, de découverte d'analogie et de faculté d'omission sélective.

LES OUTILS LOGIQUES

INTRODUCTION :

Les différents formalismes de la présentation des connaissances se déduisent de la logique des propositions (logique d'ordre 0) et celle de la logique des prédicats (logique d'ordre 1).

II-1) LE CALCUL DES PROPOSITIONS :

Le calcul de propositions se définit d'une part par sa syntaxe régissant l'ensemble des assertions exprimables dans le langage et d'autre part par ses règles d'inférences décrivant comment on peut créer de nouvelles assertions à partir d'anciennes. Toute assertion proprement formulée, appelée proposition est affectée de l'une des deux valeurs possibles VRAI ou FAUX .

Exemples : " EDISON est le lauréat du prix NOBEL en médecine " " Sicile fait partie de l'Italie ". Il est possible de représenter des propositions plus complexes que celles-ci en

utilisant les connectives logiques fréquemment employées (ET,OU,NON,IMPLIQUE,EQUIVALENT).

La table de vérité ci-dessous définit les cinq connectives

Le calcul de propositions repose sur la règle dite du "modus ponens" qui veut que si $(P \rightarrow Q)$ et que P est VRAI alors Q est VRAI.

Cela s'exprime plus formellement par : $(A \wedge (A \Rightarrow B)) \Rightarrow B$

Il y a aussi les règles de MORGAN:

$$\neg (A \wedge B) \leftrightarrow (\neg A) \vee (\neg B)$$

$$\neg (A \vee B) \leftrightarrow (\neg A) \wedge (\neg B)$$

Le calcul de propositions est insuffisant pour exprimer de nombreuses assertions utiles en intelligence artificielle. Notamment, pour décrire des faits à propos d'objets du monde réel, nous avons besoin de particulariser des objets (notion d'instantiation), de préciser la portée de certaines assertions (est ce que c'est vrai pour tous les individus d'un certain type ?). Le calcul des prédicats est une extension du calcul des propositions.

Il comporte les notions de prédicats et de symboles de quantification universelle d'existensibilité. Son intérêt principal par rapport au calcul des propositions est l'introduction de la notion de variable. Un prédicat est une fonction prenant un ou plusieurs arguments, et retournant l'une des valeurs VRAI ou FAUX, par exemple le prédicat PHONEME défini par : (PHONEME (X) : " X est un PHONEME ") Par convention on utilisera les dernières lettres de l'alphabet X, Y, Z pour les variables et les premières les premières lettres A, B, C pour les constantes. Un prédicat peut recevoir plusieurs arguments, ainsi le prédicat :DONNER (X,Y,Z) peut signifier : X donne Y à Z. Les fonctions généralisent la notion de prédicat car elles peuvent prendre n'importe quelle valeur symbolique ou numérique. Par exemple, la fonction CAPITALE (X) prend la valeur ALGER quand X = ALGERIE. On peut composer

fonction et prédicat, mais pas de n'importe quelle façon. Par exemple, si PHONEME est le prédicat et CAPITALE la fonction précédemment définie, PHONEME (CAPITALE (X)) a un sens bien que généralement faux, car cette assertion CAPITALE(X) donne par exemple Londres si X = Angleterre et ensuite PHONEME (LONDRES) donne FAUX. Par contre CAPITALE (PHONEME (X)) n'a pas de sens car l'évaluation de PHONEME (X) donne VRAI ou FAUX ce qui conduit à quelque chose d'absurde.

Introduction des quantificateurs:

Le quantificateur universel " \forall " et le quantificateur " \exists " permettant de préciser la portée de toute assertion par exemple

:

" Toute voyelle est un phonème " peut s'exprimer par : $(\forall X) (VOYELLE (X) \Rightarrow PHONEME (X))$.

Et " toute langue a une grammaire ", par :

$(\forall X) (\exists Y) (LANGUE (X) \Rightarrow GRAMMAIRE (Y) \wedge POSSEDE (X,Y))$.

** Règles d'inferences:*

Les règles d'inférences permettent de créer de nouvelles règles à partir d'assertions déjà existantes; les deux principales sont le " modus ponens " défini précédemment dans le calcul

des propositions et la " spécialisation universelle " consistant à remplacer une variable quantifiée par une constante ; [(X)(P(X)) ==> P(A)], on appelle ce processus substitution.

* Exemple de problème à résoudre:

Soit le problème suivant qui nous mène d'un état initial a un état final. à etat final .

état iniale

état finale

Donc T est sur M

T est libre

0 est libre

On peut représenter l'état initial:

sur (T,M)

sur TABLE(M)

sur TABLE(0)

SI LIBRE (T)

LIBRE (0)

sur (M,0)

La situation finale : BF sur (0,T)

sur TABLE (T)

Le programme disposera des règles suivantes:

1) LIBRE (X) ==> (y) sur (y,x)

2) sur (y,x) OTER(y,x) ==> LIBRE(X) sur (y,x)

3) LIBRE (x) LIBRE (y) EMPILER (x,y) ==> sur (x,y)

Donc comme on a les cubes M,0 ce programme va donc essayer d'appliquer la règle (3) en substituant X=M et y=0, d'ou on a

LIBRE (M)

SB1 LIBRE (0)

EMPILER(M,0)

Il va commencer par essayer d'établir LIBRE(M), ce qui l'amène à appliquer la règle (2) en substituant X=M ceci amène à établir de nouveaux sous-buts :

SB2 SUR (y,M)

OTER (y,M)

SUR (y,M) est facilement établi en substituant y=T, puisque SUR (T,M) est connu par hypothèse, ce qui amène à l'action

OTER(T,M). Première véritable action à mener dans SB1, nous avons LIBRE(M) qui est maintenant établi ce qui conduit à la deuxième action qui est maintenant possible.

EMPILER (M,0)

Le deuxième but à atteindre (Voir BF) est SUR (0,T) ce qui peut être réalisé

LIBRE(0)

SB3 LIBRE(T)

EMPILER(0,T)

En utilisant la règle (1)

LIBRE(0) ==> SUR (y,0) ==> SUR (M,0), mais ce qui d'après la règle (2), exige OTER(M,0) ce qui défait l'action déjà réalisée. Cette fois ci, le sous-but SUR(0,T) va entraîner la bonne action EMPILER(0,T) et l'on pourra ensuite EMPILER (M,0). L'examen de cette contrainte dès le départ, aurait pu éviter l'action inutile qui a provoqué un échec et un retour en arrière ; l'introduction de cette "heuristique" (voir, si la réalisation d'une tâche ne bloque pas celle d'une autre) est ce qui différencie une recherche guidée par des règles de haut niveau, d'une recherche aveugle. Beaucoup de problèmes de nature fortement déductive, comme celui-ci, ou bien le problème de tours d'HANOI, s'exprime bien en logique du premier ordre; par contre, sortir de ces jeux logiques, la plus part des problèmes de la vie qu'ont à résoudre les êtres humains sont

d'une nature beaucoup plus inductives.

L'interprétation de données médicales d'un patient entre dans cette catégorie. Elle recueille la manipulation d'informations incertaines, voire erronée qui nécessite des mécanismes de raisonnement complexes tel qu'émettre une hypothèse peu sûre à confirmer graduellement par d'autres informations, détecter une erreur possible. C'est pourquoi d'autres formalismes ont été développés pour mieux prendre en compte les aspects inhérents aux raisonnements humains.

II-3 LES LOGIQUES NON CLASSIQUES :

La logique du premier ordre celle que nous utilisons le plus souvent pour résoudre des problèmes de mathématiques, de physiques et d'informatique, semble toujours être le plus simple et la plus appropriée pour manipuler des valeurs numériques, ce n'est probablement pas elle que nous utilisons dans nos raisonnements quotidiens, ceux qui font appel au " bon sens ", au " flair ", à l'intuition. Or cette logique dite classique s'avère inadaptée à la représentation de connaissances évoluant dans la temps, incomplètes, imprécises, subjectives..., et des relations entre ces connaissances.

C'est pour cela que d'autres logiques ont existé, comme la logique floue inventée par LOTFI ZADEH en 1965 qui est en fait, une généralisation des logiques multivalentes.

II-4 CONCLUSION :

Dans ce chapitre, on a essayé de résumer les fondements mathématiques sur lesquels repose tout système "intelligent".

Dans les chapitres qui vont suivre, nous essayerons de faire les liaisons éventuelles de chaque concept " d'un système intelligent " avec l'élément de logique sur lequel il est basé.

CHAPITRE III

LA REPRESENTATION DES CONNAISSANCES ET DES

MECANISMES DE RAISONNEMENT

DEFINITION :

Une connaissance déclarative ne contient pas son mode d'emploi par contraste une connaissance procedurale indique explicitement comment elle est utilisée .

III-1 > LES REGLES DE PRODUCTIONS :

INTRODUCTION:

Une règle de production est un couple situation-action, ce qui signifie que chaque fois qu'une situation est reconnue (partie de gauche de la règle), l'action est exécutée (partie de droite de la règle); très souvent l'action à entreprendre est une conclusion à tirer mais ce n'est pas une nécessité. Il n'y a à priori pas de contrainte sur les formes qui peuvent prendre les situations ou les actions . Un système à base de règles de production comporte généralement trois parties :

- (1) La base des règles de productions.
- (2) Une ou plusieurs structures de données que l'on appelle souvent base de faits qui contient des faits connus ou des définitions utiles dans le domaine.
- (3) L'interpréteur des faits et des règles de production, c'est à dire le mécanisme qui décide quelle règle s'applique et qui déclenche l'action correspondante. Les faits et les règles obéissent à une syntaxe qui est connue par l'interpréteur ; ce dernier peut donc les manipuler logiquement . C'est à dire établir la véracité (ou non) de certaines assertions., en établir de nouvelles ou en supprimer. La distinction entre données et faits n'est pas toujours très nette et peut dépendre des auteurs . Ici, les faits sont plutôt des connaissances permanentes du programme alors que les données concernent un problème particulier à résoudre. La séparation conceptuelle entre règles de production mises sous forme aussi déclarative (et donc aussi lisible) que possible et leur mécanisme d'interprétation (appelé souvent moteur d'inférences) n'a pas toujours été aussi nette que maintenant. Les premiers systèmes à règles de production étaient souvent programmés sous une forme procédurale qui contenait à la fois la connaissance et l'interprétation de cette connaissance. Même à l'heure actuelle, certains systèmes comportent des règles qui tiennent autant du mécanisme de contrôle des inférences que des

inférences elles-mêmes. La séparation nette entre les deux et un progrès méthodologique qui a une grande influence sur les idées et les outils développés actuellement en infomatique. Plus le formalisme des règles de production est simple et uniforme plus il se prête à des capacités d'apprentissage puisque la détection de régularité dans les règles favorisent les possibilités de généralisation (ou au contraire de spécialisation) des règles existantes. En revanche, un formalisme trop simple peut être contraignant vis-à-vis des types d'assertions que l'on désire représenter dans un domaine. Il n' y a pas de compromis " idéal " entre la complexité du formalisme des règles et celui de l'interpréteur des règles.

LA BASE DES REGLES:

Le principe de base de la programmation en règles de production est que chaque règle est un morceau indépendant de connaissances. C'est-à-dire qu'elle contient toutes les conditions de son application. Un premier corollaire de ceci est qu'il n'existe pas de mécanisme ailleurs que dans la règle empêchant son application par une condition cachée. Un deuxième corollaire est qu'une règle ne peut jamais en appeler une autre : en quelque sorte les règles " ne se connaissent pas " entre elles. Seul, l'interpréteur " sait ce qu'il fait " vis-à-vis

des règles ; il est en quelque sorte le chef d'orchestre du système. Dans un système "pur", de plus, les règles ne sont pas ordonnées; n'importe laquelle peut se déclencher à priori à un moment donné. Les règles sont mises en vrac, sans savoir à l'avance comment elles seront utilisées. C'est l'interpréteur qui décide dans quel ordre les appliquer suivant certains critères (voir les détails de ces mécanismes dans la section " conflit de résolution "); il peut également n'avoir aucune stratégie et les appliquer dans l'ordre dans lequel il les trouve.

Les avantages de cette méthode de représentation peuvent être résumés ainsi: Le système est facile à modifier du fait de sa grande modularité; il est "évolutif". Les ajouts, suppressions ou modifications d'une règle ne remettent pas en cause l'architecture du système; il n'y a aucune précaution à prendre du point de vue d'effets secondaires possibles.

Normalement, plus le système possède de règles (correctes s'étend), plus il est puissant et plus ses conclusions seront fines.

Il à noter que la programmation en règle de production n'est pas la seule à être modulaire; celle à base d'objets qui est décrite dans le chapitre suivant l'est également, mais d'une autre façon le danger principal contre lequel il faut lutter lorsqu'un système évolue est la perte de sa cohérence logique;

à partir d'un certain volume et d'un grand nombre de règles (disons quelques centaines au moins). Il est difficile de vérifier "manuellement" si une nouvelle règle n'est pas redondante voire contradictoire avec une règle existante; c'est la raison pour laquelle les systèmes sophistiqués sur le plan de l'acquisition des connaissances possèdent des mécanismes puissants pour aider le concepteur du système à ne pas introduire d'incohérences (interfaces amicales et détection de ressemblances entre règles).

CLASSIFICATION DES INTERPRETEURS :

Vers l'avant ou vers l'arrière:

Une première façon de caractériser un moteur (interpréteur) provient de la manière dont il essaie d'appliquer les règles en fonction des faits qu'il examine. S'il considère des DONNEES ou FAITS déjà établis et qu'il regarde s'ils satisfont la partie gauche des règles (la partie prémisse), on dit qu'il fonctionne vers l'avant ou qu'il est guidé par les données. S'il considère les BUTS possibles à atteindre en examinant les parties de droite des règles (la partie action) et qu'il essaie de vérifier les règles concluant vers ces buts (et uniquement celles-là), on dit qu'il fonctionne vers l'arrière. Cette distinction est strictement équivalente à celle des

interpréteurs de grammaire qui sont soit ASCENDANTE (équivalents de VERS L'AVANT) car on part des données soit DESCENDANTS (équivalent de VERS L'ARRIERE) car on part des buts possibles.

Aucune de ces stratégies n'a fait la preuve de sa supériorité du point de vue efficacité de façon universelle. La stratégie vers l'avant possède l'avantage de mieux contrôler dans quel ordre on veut acquérir les données qui vont éventuellement satisfaire les parties prémisses des règles. La stratégie vers l'arrière possède l'avantage de mieux cerner les buts que l'interpréteur a quelque raison de vouloir atteindre car il n'applique que les règles conduisant vers ces buts, mais il est par contre difficile de prévoir dans quel ordre elles seront appliquées.

Suivant les stratégies de résolutions de conflits:

La notion de " résolution de conflits " se réfère au comportement de l'interpréteur lorsque, dans une situation donnée, plusieurs règles peuvent se déclencher; il doit alors prendre la décision, soit de les appliquer dans un certain ordre (ou non), et de n'en déclencher que quelques unes (ou toutes).

Le problème de l'ordre des règles est à plusieurs niveaux; le système peut avoir classé a priori les règles dans un certain ordre d'application; la stratégie la plus couramment utilisée

est alors par ordre décroissant des forces des conclusions ou de force de premisses si le système fait usage de coefficient de vraisemblance. Les règles peuvent être aussi classées à la main lors de leur introduction dans le système; une telle procédure est en général une faute méthodologique car elle a pour conséquence une diminution de la modularité des règles qui est la vertu habituellement reconnue du formalisme des règles de production. Le développeur peut cependant avoir de bonnes raisons de faire ainsi dans des cas particuliers où l'efficacité du système prime sur l'aspect évolutif.

Cependant, le principe de modularité dit qu'après l'exécution d'une règle, n'importe quelle autre peut être déclenché. Revenons donc au cas où c'est l'interpréteur qui classe les règles dans un certain ordre; dans ce cas la modularité n'est pas altérée puisque c'est le système qui s'occupe d'insérer toute nouvelle règle à la bonne place. Bien entendu, un tel classement n'est véritablement intéressant que si la recherche de toutes les règles s'appliquent à un moment données n'est exhaustive et que ne seront donc appliquées que les règles considérées comme les plus importantes. Une telle stratégie peut être considérée comme un " comportement intelligent " car toute sélection de critères de décision reflète un raisonnement plus complexe que la recherche aveugle de toutes les possibilités (ceci suppose bien entendu

que la complexité (là où elle est nécessaire) figure parmi les critères d'intelligence).

Un exemple plus simple mais concret:

L'exemple suivant montre une base de règles décrivant des animaux ou groupes d'animaux en termes de certaines caractéristiques. Le but du système est de découvrir le ou les animaux correspondant à cette description. La base des règles est volontairement incorrecte et insuffisante afin de montrer la facilité d'amélioration de systèmes basés sur ce formalisme. D'autre part, elle est peu réaliste en ce sens que les conclusions à tirer ne sont pas aussi sûres dans la réalité. Des mécanismes manipulant des poids traduisant la plausibilité des assertions seraient nécessaires. Les exemples réalistes et intéressants utilisant des inférences sont malheureusement difficiles à montrer complètement, car il font rapidement appel à de nombreuses règles dont les conclusions s'enchaînent. L'exemple ci-dessous est complètement caractérisé par une base de règles, l'algorithme décrivant l'interpréteur et les faits au départ.

BASE DE REGLES : (R1) si Allaites-enfants Alors Mammifère
(R2) si A-des-plumes Alors Oiseau
(R3) si Donne-fourrure ou Mammifère Alors
vit-dans-forêt.

(R4) si Oiseau-ne-vole et

ne-vit-pas-pas-dans-la-forêt Alors

Manchot.

(R5) si vit-dans-forêt et très lourd Alors Ours

(R6) si très-lourd et Mammifère Alors Baleine.

INTERPRETEUR

1- Trouver toutes les règles dont les conditions sont vraies.

2. S'il y en a plus d'une, ne pas prendre en compte celles qui dupliquent une propriété (déjà connue) ("RESOLUTION DE CONFLIT")

3. Exécuter l'action de plus petit numéro: s'il n'y en a pas sortir ("ACTION")

4. Recommencer

FAITS AU DEPART ET BUTS POSSIBLES

FAITS =(allaites-ses-petits, Très lourd)

BUTS POSSIBLES =(Ours, baleine, Manchot)

La stratégie de l'interpréteur ainsi définie est la plus simple que l'on puisse concevoir; il examine continuellement la base de règles, déclenchant l'action correspondant à la règle qui s'applique à condition que celle-ci ne porte pas sur une conclusion déjà établie.

A la première itération, seul R1 est applicable; la conclusion MAMMIFERE est ajoutée la base de fait devenant:

FAITS =(allaites-ses-petits, Très lourd, Mammifère)

A la deuxième itération les trois règles R1,R3,R6 sont applicables: R1 est rejetée car elle duplique une propriété déjà connue donc c'est R3 (plus petit numéro) qui est exécutée.
FAITS :(allaites-ses-petits, Très lourd, Mammifère, Vit-dans-forêt).

A la troisième itération, quatre règles R1, R3, R5, R6 sont applicables; R1 et R3 sont rejetées: restent R5 et R6 donc R5 est exécutée.

FAITS :(allaites-ses-petits, Très lourd, Mammifère, Vit-dans-foret, Ours)

A la quatrième itération enfin, R6 (en attente depuis longtemps) est exécutée et le système s'arrête.

FAITS :(allaites-ses-petits, Très lourd, Mammifère, Vit-dans-foret, Ours, Baleine)

On peut remarquer tout suite que l'influence au niveau logique et discriminant de la base des règles a fait que nous avons deux animaux répondant à la description initiale; il est facile de voir que la caractérisation de la baleine est très incomplète. D'autre part, il y a probablement d'autres mammifères très lourds vivant dans la forêt que l'ours et il manque certainement des règles correspondantes. Montrons simplement comment on peut régler provisoirement le problème de la double conclusion. La règle R6 en est responsable; il manque manifestement une clause supplémentaire dans la partie

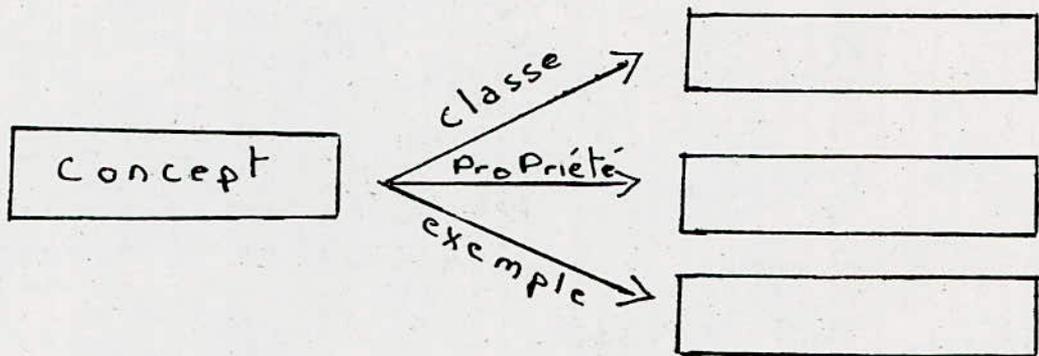
prémisse; R6 pourrait devenir par exemple:

(R6) Si très-lourd et mammifère et Vit-dans-l'eau Alors Baleine
Au cours d'une nouvelle exécution du programme après
modification de R6, le seul changement sera la non exécution de
R6 et nous obtiendrons seulement "Ours" comme conclusion.

Remarquons que la règle R3 non plus n'est pas bonne et qu'elle
serait à modifier. La leçon à tirer de cet exemple est que la
modification du programme a pu être faite en améliorant une
structure déclarative (il y a en général des éditeurs prévus à
cet effet dans tout langage de haut niveau) et non pas en
ayant à chercher une instruction bien cachée au milieu d'un
programme, qui fait partie de la mise au point classique de la
programmation traditionnelle.

III-2 > LES RESEAUX SEMANTIQUES

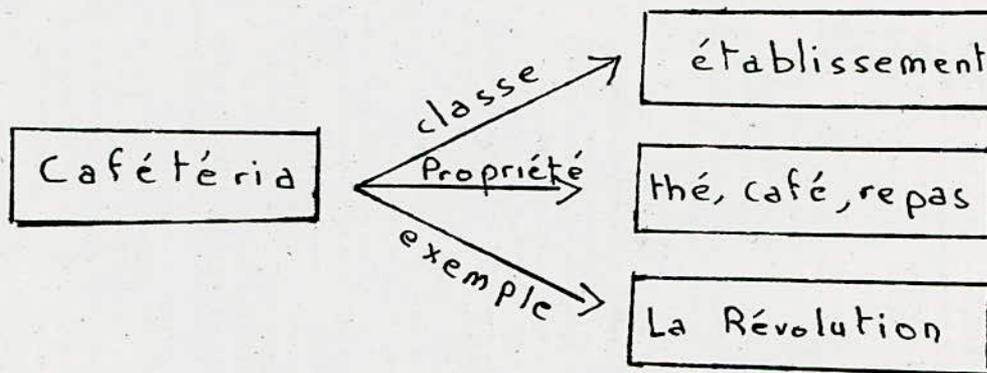
Les réseaux sémantiques, dont la structure en arbre n'est qu'un
cas particulier, permettent une représentation des
connaissances beaucoup plus riche. Ces graphes, se composent de
noeuds, représentant des objets, événements ou concepts et
d'arcs, représentant toutes sortes de relations entre eux.



(fig - 1)

Les propriétés et les attributs se transmettent de noeud en noeud en "glissant" le long du réseau, comme des associations d'idées.

Ces graphes font ressortir l'ensemble des interconnexions entre les composantes significatives (sémantique) en mémoire. Ils permettent de représenter aussi bien les connaissances statiques: "une cafétéria est un établissement ou l'on sert du thé, du café et des repas légers".



(fig - 2)

Que des connaissances dynamiques telle que : "Si le moteur ne démarre pas et que le démarreur tourne, alors regarde les bougies puis l'allumage ". Les réseaux sémantiques peuvent être schématisés selon deux formes différentes. L'une (voir fig ci-dessous) renferme un ensemble de registres; chaque registre est identifié par un numéro, et chacun fait référence d'un certain nombre d'autres registres. Ainsi le registre 102 fait référence aux registres 101, 103, 104, etc ...

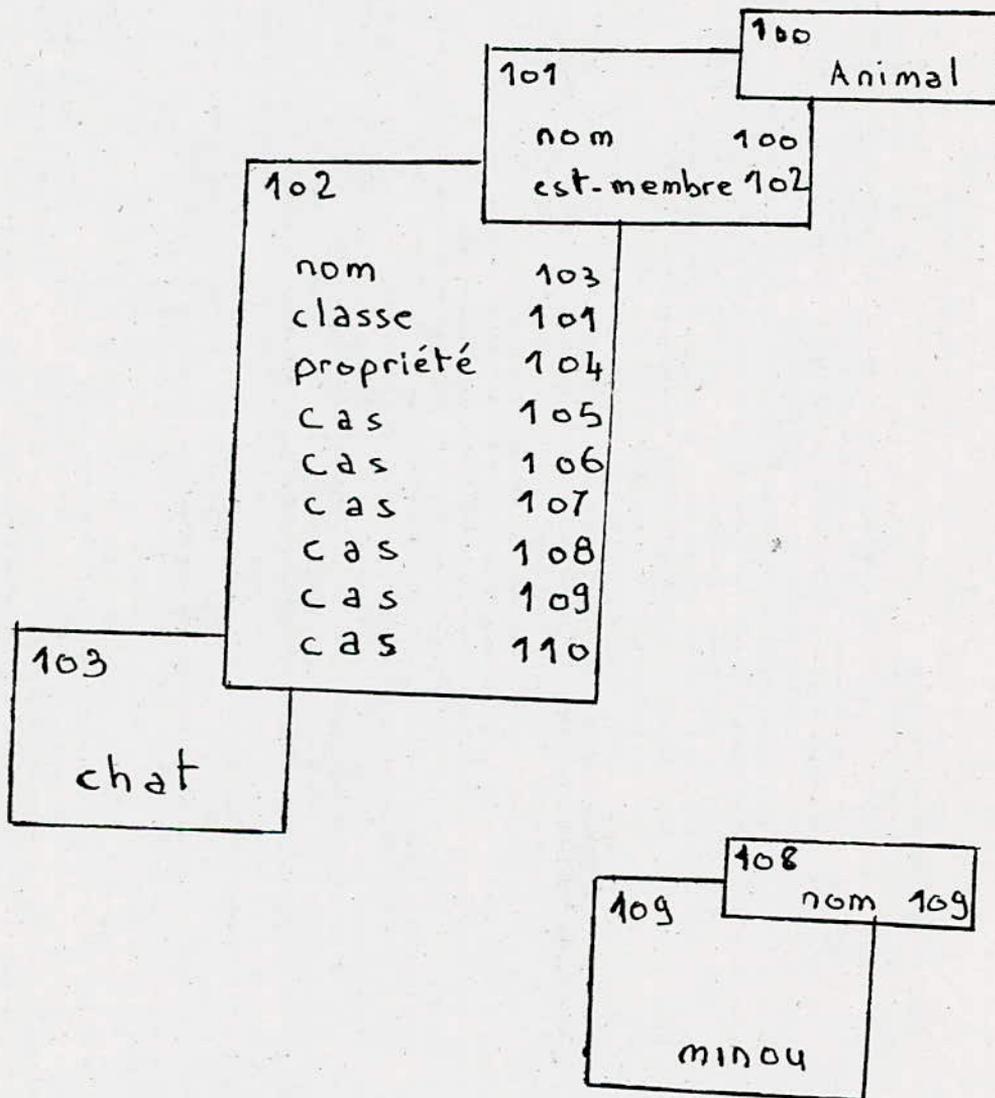
La clareté des relations entre objets est un avantage majeur des réseaux sémantiques : le mode de représentation graphique facilite en effet la compréhension, lorsqu'il s'agit pour l'utilisateur d'examiner le contenu de la base de connaissances. C'est précisément l'inverse pour les règles de production: Les relations ne sont pas explicites puisque, pour les trouver, il faut comparer, règle après règle, les conditions de l'une avec les conséquences de l'autre.

III-3) LES OBJETS STRUCTURES :

Il existe d'autre façon de représenter les connaissances de manière structurée. Les objets structurés sont une extension des réseaux sémantiques, incorporant des connaissances procédurales, mais non inférentielles. Certains auteurs en citent un grand nombre de catégories, d'autres préfèrent les regrouper sous 2 ou 3 dénominations différentes. C'est ainsi que l'on peut distinguer: les frames, les schémas, les scénarios, les scriptes, les prototypes, les objets, les formes et classes, les acteurs, les types abstraits...

Les schémas sont souvent assimilés aux frames, bien que ces derniers soient attribués à Minsky et que les premiers fassent plutôt référence à des travaux sur la mémoire. Ils ont été

Le registre 103 contient l'information au sujet du nom d'un item auquel on s'est référé dans 102 ...



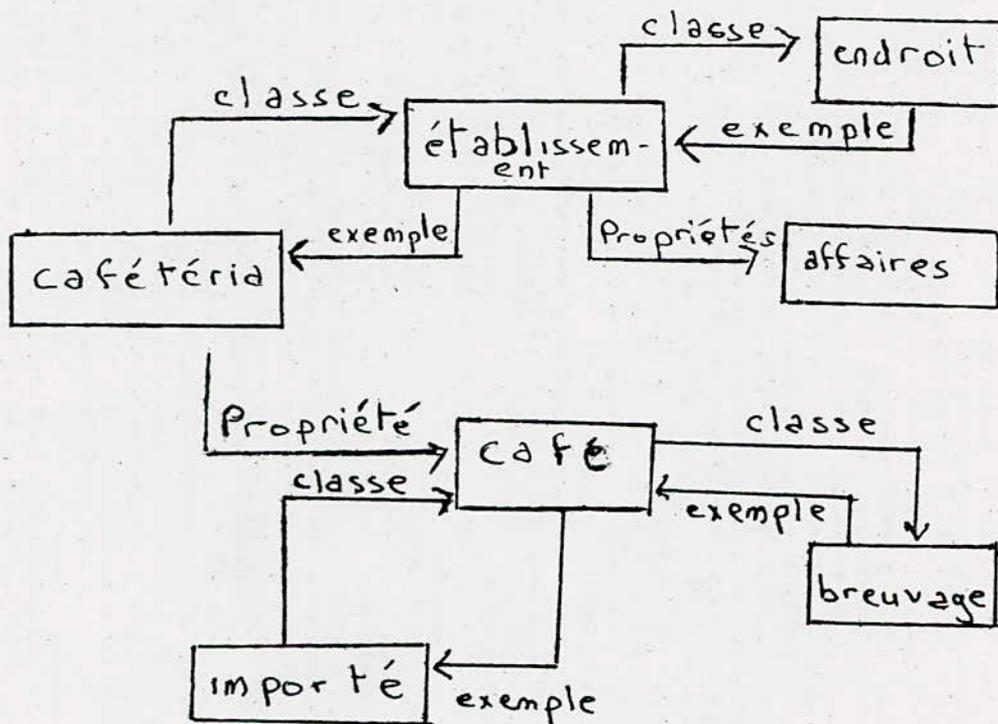
(Fig - 3)

particulièrement par J.Ferber (L.R.I., Orsay). Ce sont des outils de représentation hiérarchique des connaissances, permettant de structurer la base de règles. Les schémas se présentent sous forme d'attributs ou "slot" qui décrivent toutes les caractéristiques liées à une entité conceptuelle. Chacun contient un groupe de règles et un groupe de faits qui concernent un sous ensemble d'un problème complexe.

Les schémas sont reliés entre eux par les liens hiérarchiques: tout schéma est une spécialisation d'un autre schéma dont il affine la description. Un schéma peut être instancié plusieurs fois, et de ce fait les règles peuvent être utilisées pour des objets différents. Cela permet d'écrire des règles générales pouvant être instanciées pour chaque cas particulier devant être traité.

Chaque attribut (ou slot) d'un schéma peut désigner des rubriques telles que: "élément de ", "date", "lieu", "couleur"... A chaque slot correspond une description appelée "valeur du slot", comprenant diverses facettes (valeur par défaut , valeurs possibles, intervalle de validité...) (fig). Celles-ci peuvent contenir aussi bien des données que des procédures (métaconnaissances).

La (fig-4) illustre une autre façon de schématiser l'information. Les noeuds, dans cette représentation, correspondent aux registres de la forme précédente. Ici, les relations entre les noeuds ont des noms qui permettent la distinction des leurs différentes significations.



(fig - 4)

CONCLUSION

Il est difficile de faire une comparaison entre les différents modes de représentation car le choix d'une méthode dépend du domaine considéré.

Certaines modes de représentations des connaissances satisfont mieux quelques critères par rapport à d'autres représentations, mais il n'existe pas une représentation idéale s'adaptant à tous les domaines.

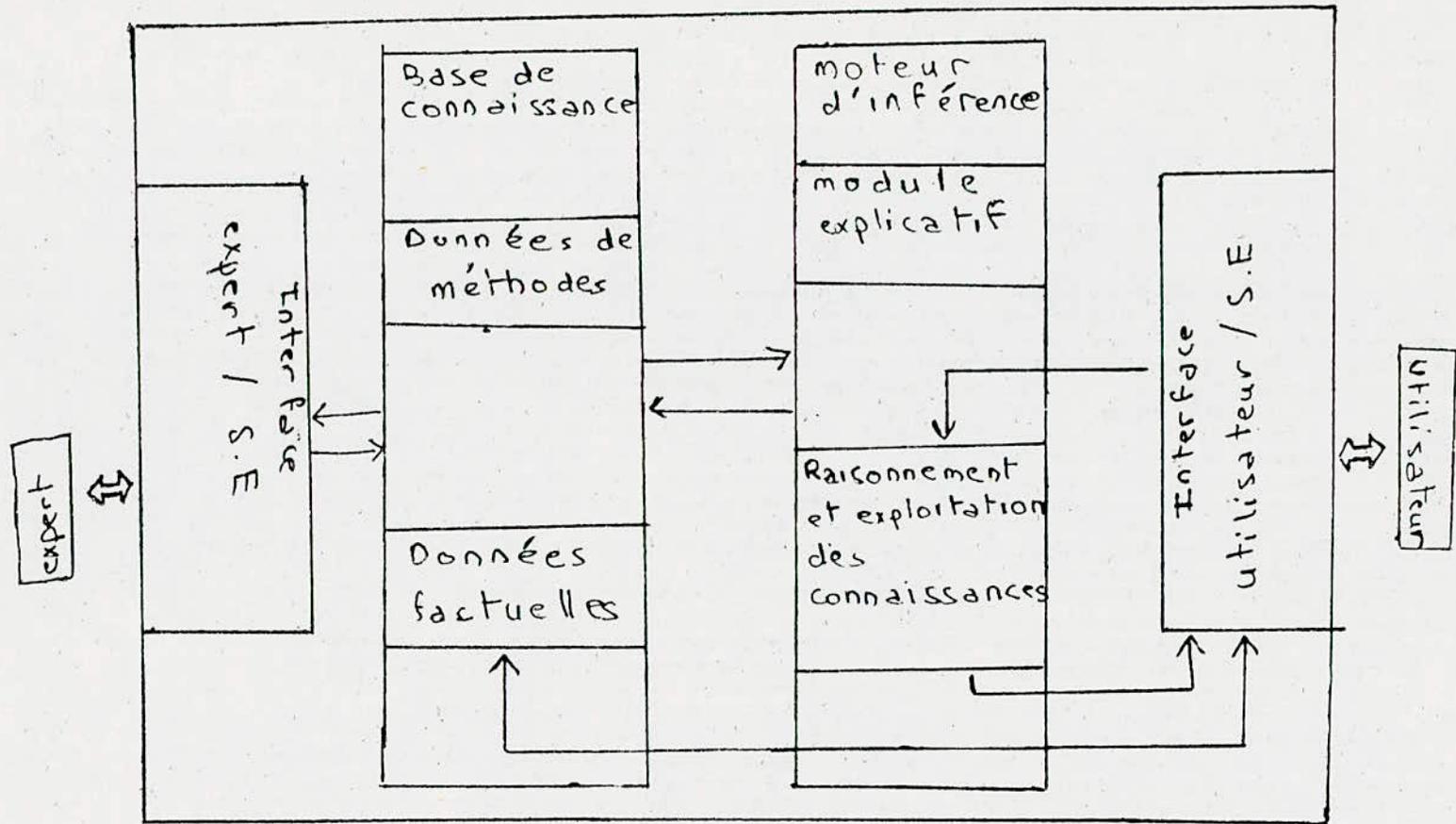
" THEORIE DES SYSTEMES EXPERTS "

CHAPITRE I

LES SYSTEMES DITS EXPERTS.

La méthodologie des systèmes experts (S.E), consiste à séparer le plus complètement possible la connaissance d'un domaine de la façon d'utiliser cette connaissance. Cela correspond à considérer toute la connaissance d'un domaine comme des données pour le programme d'informatique.

Pour concevoir un système expert, il faut, d'une part, transférer à la machine les connaissances d'un expert humain dans une spécialité donnée et, d'autre part, donner à cette machine les moyens de raisonner comme un expert humain. A ces deux fonctions correspondent les deux parties essentielles qui forment un système expert, la base de connaissances et le moteur d'inférence, c-à-d la partie qui met en oeuvre les éléments de la base de connaissances (fig 1).



(fig-1) Anatomie d'un système expert.

1- Base de connaissances :

Elle est constituée de deux parties distinctes:

1-a Données factuelles:

Elles correspondent à une situation précise.

1-b Données de méthodes

Elles correspondent à l'ensemble des connaissances d'un expert dans un domaine limité.

2- Environnement de développement d'un système expert:

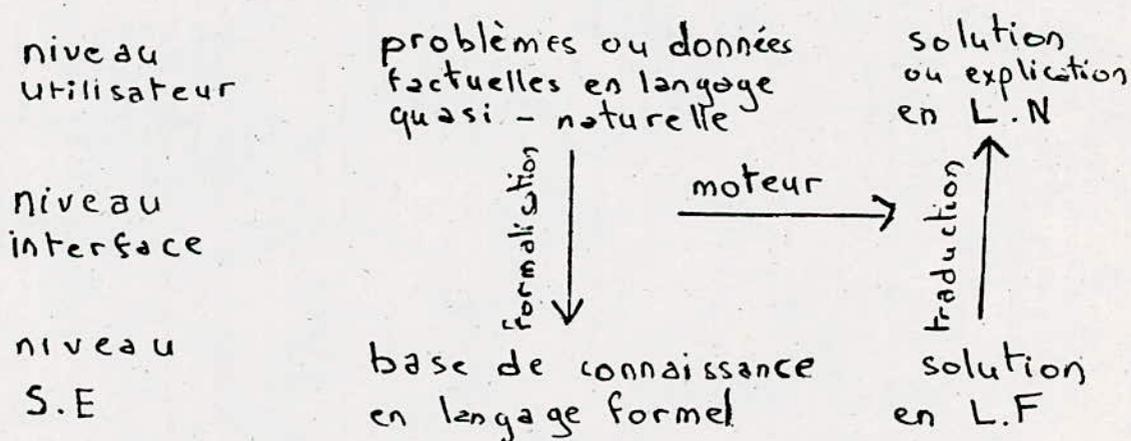
Pour qu'une base de connaissance quelconque puisse être utilisée de manière efficace, il faut lui associer des possibilités d'interfaçage avec le monde extérieur (utilisateur, experts ...) ainsi qu'un mécanisme de raisonnement qui exploite ces connaissances.

L'environnement de développement d'un système expert (E.D.S.E) n'est pas spécifique à une application donnée.

a- Interfaçe U/S.E (utilisateur / système expert)

Ce sont des programmes d'interprétation d'un langage quasi-naturel et de mise en forme de réponses. Ainsi l'utilisateur peut demander des explications au moteur d'inférences, suivre son raisonnement dans un langage qui lui est compréhensible et inversement lui fournir des informations.

Le schéma suivant illustre cette communication.



b- Interface E/S.E : (Expert / S.E)

Elle permet à l'expert d'accéder à la base de connaissances (plus exactement aux données de méthodes) pour modifier ou supprimer des informations ainsi dans le cas d'un ajout de connaissances supplémentaires, le processus de communication se schématise de la manière suivante :

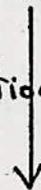
Niveau
expert

Niveau
interface

Niveau
S.E

données de méthodes
en langage quasi -
naturel.

formalisation



base de
connaissances.

c- Moteur d'inférence:

Il est formé de deux parties:

c-1 Module explicatif :

Cette partie du moteur d'inférence se charge de garder une "trace " du plan de résolution pour un problème donné, afin d'être apte à fournir éventuellement (via l'interface U/S.E) une explication de son raisonnement.

c-2 Mécanisme de raisonnement et exploitation de connaissances:

Ce module doit être capable d'assurer les fonctions suivantes:

- sélection d'hypothèses
- obtention de données
- combinaison de données factuelles et données de méthodes pour produire des conclusions.

c- 2 - 1) Sélection d'hypothèses:

Quelques moteurs d'inférences s'appuient sur un ensemble d'hypothèses prédéterminées et vérifient chacune d'elles séparément, (absence de données factuelles initiales). D'autres génèrent un ensemble d'hypothèses résultant de l'observation de données factuelles initiales et vérifient ces hypothèses seulement.

Enfin l'efficacité d'un moteur d'inférence est d'autant plus accrue qu'il s'adapte aux deux situations suivantes: absence ou

c- Moteur d'inférence:

Il est formé de deux parties:

c-1 Module explicatif :

Cette partie du moteur d'inférence se charge de garder une "trace " du plan de résolution pour un problème donné, afin d'être apte à fournir éventuellement (via l'interface U/S.E) une explication de son raisonnement.

c-2 Mécanisme de raisonnement et exploitation de connaissances:

Ce module doit être capable d'assurer les fonctions suivantes:

- sélection d'hypothèses
- obtention de données
- combinaison de données factuelles et données de méthodes pour produire des conclusions.

c- 2 - 1) Sélection d'hypothèses:

Quelques moteurs d'inférences s'appuient sur un ensemble d'hypothèses prédéterminées et vérifient chacune d'elles séparément, (absence de données factuelles initiales).

D'autres génèrent un ensemble d'hypothèses résultant de l'observation de données factuelles initiales et vérifient ces hypothèses seulement.

Enfin l'efficacité d'un moteur d'inférence est d'autant plus accrue qu'il s'adapte aux deux situations suivantes: absence ou

caractéristiques , parmi elles :
existence de données initiales .

c- 2 - 2) Obtention de données:

Pour confirmer ou infirmer une hypothèse, le moteur d'inférence requiert des données du monde extérieur. Ces données peuvent être puisées directement de la base de connaissances, en questionnant l'utilisateur (via l'interface U/S.E), par l'activation de procédures externes, ou encore en consultant une base de données système expert).

c - 2 - 3) Combinaisons de données factuelles et données de méthodes:

Selon les différentes approches possibles de modélisation du raisonnement humain, le moteur d'inférence exploite les connaissances appropriées au problème posé dans le but d'aboutir à une (des) conclusion(s).

CHAPITRE II

MOTEUR D'INFERENCE

Définition 1 :

On appelle système de production ,tout système composé d'une base de connaissance et d'un moteur d'inférences.

Définition 2 :

On appelle base de connaissance ,toute base de connaissance dont les données factuelles et données de méthodes sont représentées soit par des formules de la logique propositionnelle, soit par des formules du calcul des prédicats .

Les données factuelles sont alors appelées "faits" et sont regroupées dans une base de faits (BF).

Les données de méthodes sont alors appelées "règles" et sont regroupées dans une base de règles (BR) .

Remarques :

Chaque règle fournie par les experts est rédigée de telle sorte que le texte obtenu définit :

- i) non seulement les effets de la règle .
- ii) mais aussi les conditions de déclenchement de celle-ci .

Définition 3 :

On appelle moteur d'inférences ,un programme qui exprime les

stratégies et heuristiques d'exploitation des règles .

Un moteur d'inférence est caractérisé par plusieurs caractéristiques , parmi elles :

- le type de mécanismes d'inférence ;
- la logique associée au moteur ;
- le mécanisme de fonctionnement .

II-1) TYPES DE MECANISMES D'INFERENCE :

Il existe plusieurs types de mécanismes d'inférences :

- le chaînage avant ;
- le chaînage arrière ;
- le chaînage mixte .

a) Chaînage avant :

On part des faits connus (données de l'utilisateur) et on déclenche toute règle dont les prémisses sont satisfaites . On ajoute les faits conséquents à la base de faits . Les règles sont alors dites "règles en avant" et exploiter de telles règles correspond à un mécanisme depuis "les données vers les buts".

b) Chaînage arrière :

Un moteur d'inférence fonctionne en chaînage arrière lorsque :

- Certains faits de la base de faits sont considérés comme étant à établir (on les appelle souvent "hypothèses" ou

"buts").

-On regarde les règles qui ont le but fixé dans leur conséquence : on considère chacune de ces règles et si l'une d'elles a toutes ses prémisses dans la base de faits ,on a un succès , sinon on considère les prémisses comme de nouveaux et on recommence . Les règles sont alors dites "règles en arrière" et exploiter de telles règles correspond à un raisonnement "depuis les buts vers les données" .

c)Chainage mixte:

Certains systemes fonctionnent en chainage mixte :

- Une partie des faits de la base de faits est considérée comme établie (ce sont les faits proprement dits).

-Les conditions de règles peuvent porter simultanément sur les faits de l'une ou de l'autre sorte .

Remarques :

-Lorsque le moteur d'inférences effectue le mécanisme de chainage avant ou chainage arrière ,cela revient à explorer un arbre par l'une des méthodes vues au chapitre précédent .

-le principe du chainage avant repose sur la règle d'inférence appelée "modus ponens" .

-le principe du chainage arrière repose sur la règle d'inférence appelée "modus tollens" . On considère que la négation du but est ajoutée aux faits et on recherche par applications successives du modus tollens la négation d'un

fait, ce qui donnera une contradiction et donc la démonstration du but .

-le choix du type de chaînage approprié est important , il dépend d'une part du domaine d'application du système expert , d'autre part de la manière de raisonner de l'expert .

Nous donnons quelques exemples d'utilisation du chaînage avant et du chaînage arrière .

Exemple 1 : En début de consultation du système expert, il peut être intéressant de déduire tout ce qu'on peut de ce que l'on sait déjà : C'est un chaînage avant .

Exemple 2 : Le problème a beaucoup de solutions réalisables , on aimerait en trouver une rapidement moyennant une stratégie : c'est un chaînage avant .

Exemple 3 : Les possibilités finales sont peu nombreuses (diagnostic , selection , classification ...) , le chaînage arrière est souvent adéquat .

II-2-LOGIQUE ASSOCIEE AU MOTEUR :

Dans le cas d'un système de production, les prémisses et les conséquences des règles peuvent être soit :

- Des propositions logiques : le moteur d'inférences est alors d'ordre zéro .
- Des prédicats avec variables : le moteur d'inférences est alors d'ordre un .

II-3-MECANISME DE FONCTIONNEMENT:

Le moteur d'inférences enchaîne des cycles de travail comportant chacun deux phases : la phase évaluation et la phase exécution .

Lorsque le moteur d'inférences est lancé , la base de connaissances contient les informations représentatives de l'énoncé du problème à traiter .

- Les faits avérés et faits à établir constituent la base de faits .

Les connaissances opératoires sur le domaine constituent la base de règles .

En phase d'évaluation , le moteur d'inférences détermine s'il existe dans la base de règles , des règles à déclencher au vue de l'état courant de la base de faits , et si oui quelles sont ces règles .

En phase d'exécution le moteur d'inférences déclenche les règles retenues par l'évaluation .

1-PHASE D'EVALUATION :

La base de connaissances ou plus exactement les règles représentent en principe toute l'expertise d'un expert dans un domaine donné . De ce fait , elles correspondent à une masse considérable de connaissances qu'il faut utiliser de façon judicieuse et efficace .

a) Séléction (ou restriction) :

La base de connaissances ou plus exactement les règles représentent en principe toute l'expertise d'un expert dans un domaine donné .De ce fait ,elles correspondent à une masse considérable de connaissances qu'il faut utiliser de façon judicieuse et efficace .

La difficulté est de décider à tout moment de l'ensemble des règles appropriées au problème posé .Il est donc indispensable d'instaurer une notion de "métaconnaissance " :ce qualificatif s'applique généralement à la faculté que l'on a de " savoir ce que l'on sait " ,sans insister sur le problème philosophique de déterminer si la métaconnaissance fait partie de la connaissance ,nous utiliserons cette distinction,ce qui peut avoir conceptuellement un intérêt lorsqu'on décrit un système.

Instaurer une métaconnaissance peut se faire en subdivisant

le domaine en sous domaines plus spécifiques appelés " espaces de connaissances " .

Dans le cas d'un système de productions, il est très fréquent d'utiliser des métarègles .

Les métarègles introduites par R.DAVIS sont des règles agissant sur les règles " objets " .Par exemple les metarègles peuvent conseiller d'utiliser des règles d'inférences avant d'autres ,compte tenu de la situation .Cela peut permettre d'éliminer d'un seul coup un ensemble de règles inadéquates dans de tels types de situation .

Soit R1 l'ensemble des règles résultant de la phase sélection.

b) Filtrage :

Le moteur d'inférences compare la partie déclencheur de chacune des règles de R1 par rapport à l'ensemble F1 des faits.

Un sous ensemble R2 de R1 rassemble alors les règles jugées compatibles avec F1, c'est à dire celles dont les conditions de déclenchement ont été jugées satisfaites .Soit R2 l'ensemble des règles ainsi obtenues.(R2 est appelé ensemble de conflits).

c) Résolution de conflits :

Une fois l'étape du filtrage effectuée, il est fréquent que plusieurs règles de R2 soient candidates à l'utilisation, dans ce cas plusieurs stratégies permettent le choix de l'une d'entre elles. Trois grandes familles de structures de contrôle se rencontrent actuellement dans la littérature:

c-1. La recherche exhaustive:

Essai systématique de toutes les règles.

c-2. Le choix par évaluation:

Selon plusieurs critères possibles :

- sélection des premières règles rencontrées par rapport à un ordre statique .
- sélection des règles qui comportent le moins de conditions dans leurs prémisses .
- sélection de la règle la plus précise (coefficient de crédibilité) .
- etc

c-3. Le contrôle par métarègles :

Consiste à instaurer des métarègles d'un niveau de détails très spécifique .

2 -) PHASE D'EXECUTION :

La phase d'exécution constitue la deuxième partie de chaque cycle : le moteur d'inférence commande la mise en oeuvre des actions définies par les règles de R3 .

si R3 est vide, il n'y aura pas de phase d'exécution pour le présent cycle .

la conduite à adopter lorsque R3 compte plus d'une règle peut être très différente d'un moteur à l'autre, dépendant de la stratégie de contrôle du moteur .

3 -) STRATEGIE DE CONTROLE :

on distingue deux types majeurs de stratégies de controle :

- stratégie irrévocable .
- stratégie par tentatives .

a) stratégie irrévocable :

Dans ce type de régime de controle, la règle sélectionnée est appliquée de façon irrévocable c'est à dire sans considération ultérieure des autres règles de R3 ,quelque soit l'issue de l'application de la règle élue .

Dans un tel régime,il n'est pas nécessaire de garder un point de retour .

b) stratégie par tentatives :

Une règle est élue (aussi bien arbitrairement que pour d'autres raisons) et un point de retour est gardé pour une application d'autres règles de R3 en cas d'échecs .

Ce type de stratégie de controle peut être réalisé par deux techniques : -Backtracking

-Graphe de recherche et controle

-Backtracking : les effets de la règle sélectionnée sont oubliés et une autre règle (point de retour) est sélectionnée à la place .

-Graphe de recherche de contrôle :il repose sur le même principe que le Backtracking avec une structure d'arbre et

une stratégie de recherche (graphe ET / OU...).

4- CYCLE DE BASE OU MOTEUR D'INFERENCE :

La résolution d'un problème par un moteur d'inférence se caractérise par l'enchaînement de plusieurs cycles de base illustrés par le schéma suivant :

CHAPITRE I

PRISE D'EXPERTISE

I) PRESENTATION DU DOMAINE DE LA RECONNAISSANCE DE LA

PAROLE :

a) Aspects spécifiques de la compréhension de la parole:

Un souci constant des recherches faites en intelligence artificielle a été de développer des interfaces "agréables" entre utilisateurs et systèmes informatiques. Par exemple dans le cadre d'accès à des bases de données. La parole étant l'un de nos moyens les plus naturels pour communiquer, de nombreuses recherches ont été menées dans ce domaine.

Le problème de la compréhension de la parole est plus ardu que celui de la compréhension du langage écrit (ou tapé sur un clavier); ceci est dû à plusieurs facteurs dont les principaux sont :

- a) Le signal résultant de l'émission de parole comporte des "bruits" c'est à dire des non informations à éliminer;
- b) La prononciation des locuteurs n'est pas parfaite et varie d'un individu à l'autre;
- c) La prononciation d'un locuteur donné est variable suivant son état (psychologique physiologique);
- d) La prononciation d'un phénomène voire d'un mot n'est pas la

même suivant que ces derniers sont isolés ou en contexte;

e) La frontière entre les mots successifs d'une phrase n'apparaît pas sur le signal vocal; il peut très bien y avoir des intervalles de silence au milieu d'un mot et aucun intervalle entre deux mots successifs;

f) Des mots totalement différents peuvent avoir la même prononciation; c'est le problème des homonymes par exemple : ver, verre, vers, vair, vert.

Il résulte de tout cela qu'un système de compréhension de parole reçoit en entrée un treillis d'éléments phonétiques correspondants à la transcription de la phrase prononcée. Ce treillis contient de nombreuses erreurs (actuellement de l'ordre de 30 % dans le meilleur des cas), ce qui introduit un indéterminisme important dans les traitements ultérieurs.

De ce treillis une démarche ascendante permet d'extraire un treillis de mots possibles résultant des diverses combinaisons acceptables en fonction du lexique de l'application. Le rôle de l'analyseur peut être de trouver un chemin cohérent à travers ce treillis de mots en utilisant les diverses sources d'informations disponibles. Ce processus purement ascendant s'avère vite insuffisant pour des langages d'une certaine complexité et doit être complété par un processus descendant guidé par les informations de haut niveau (sémantique, pragmatique, syntaxe, etc...).

de deviner les intentions de celui qui parle afin de lui répondre de façon appropriée et pas seulement au niveau d'interprétation immédiate.

La grande difficulté dans la compréhension de la parole provient de deux sources d'incertitude et d'erreurs inhérentes au "traitement" du message de la communication à analyser.

La première est due au locuteur. La traduction de sa pensée en Les appareils disponibles dans le domaine actuellement s'attaquent surtout au problème de la reconnaissance de mots isolés. Même pour ce problème qui est beaucoup plus facile que la reconnaissance de discours continu, les meilleurs systèmes existants reconnaissent les mots isolés avec un taux de succès de 99,5 % dans des vocabulaires de l'ordre de 120 mots pour un locuteur après apprentissage.

Des systèmes de reconnaissance de mots enchaînés (prononcés de façon continue sans construction syntaxique) sont également opérationnels depuis quelques années.

Différents niveaux de connaissance

En allant du niveau "signal" au niveau le plus élevé de la compréhension, les types de connaissances nécessaires pour un système complet de compréhension de la parole sont :

- a) Phonétique: représentation des caractéristiques des sons dans tous les mots du vocabulaire;
- b) Phonologique : règles décrivant les variations de

prononciation lorsque les mots sont prononcés dans des phrases (phénomènes de coarticulation, assimilation).

c) Morphémique : règles décrivant comment les morphèmes (plus petite unité " signifiante ") sont combinés pour former des mots (formation des pluriels, conjugaison des verbes);

d) Prosodique : règles décrivant les fluctuations en accent et intonations, par exemple la montée du ton par une phrase interrogative;

e) Syntaxique : règles de formation des phrases;

f) Sémantique : prise en compte de la signification des mots et des phrases pouvant servir de contraintes utiles à

l'élimination de constructions syntaxiques théoriquement possibles mais peu probables.

g) Pragmatique : règle de la conversation permettant par exemple de deviner les intentions de celui qui parle afin de lui répondre de façon appropriée et pas seulement au niveau d'interprétation immédiate.

La grande difficulté dans la compréhension de la parole provient de deux sources d'incertitude et d'erreurs inhérentes au " traitement " du message de la communication à analyser.

La première est due au locuteur. La traduction de sa pensée en sons s'accompagne d'erreurs telles que le choix des mots, l'articulation, répétitions inutiles, certains sons bizarres (raclement de gorge), autant de particularités (variables avec l'individu) qui dégradent le message idéal.

La deuxième difficulté est due à l'auditeur, qui doit effectuer la transformation inverse du locuteur c'est à dire le passage du signal aux intentions du locuteur. Au cours de cette transformation, l'interlocuteur fait des erreurs de jugement

dues à des règles imprécises de compréhensions. Notons que la communication entre être humains implique également de nombreuses demandes de répétitions ou d'éclaircissement.

B) Les différentes ambiguïtés des langues naturelles:

a) L'ambiguïté lexicale:

Certains mots comme " côte " , " avocat " , " suite " , ont un sens différent selon le contexte . Ainsi la question "est-ce-que la pêche était bonne ?" peut s'adresser à un pêcheur , mais aussi à une autre personne qui vient de manger un fruit .

b) L'ambiguïté structurale dépasse le stade du mot isolé . Par exemple , il est possible de définir "il faut le pouvoir" de deux manières différentes , selon qu'on considère "pouvoir" comme un nom et "le" comme l'article qui s'y rapporte ou bien "pouvoir" comme un verbe et "le" comme pronom neutre objet , qui représente l'idée continue dans la proposition précédente . Ainsi selon que l'on analyse grammaticalement, cette phrase de la première ou de la deuxième façon, elle n'aura pas du tout le même sens. On retrouve ce même problème dans la phrase " la petite brise la glace " (voir figure (1)).

L'ambiguïté de structure profonde est encore plus subtile : elle apparaît lorsqu'il est possible d'interpréter la même phrase de deux façons différentes sans que la grammaire apparente en soit modifiée pour autant. Par exemple, d'après la phrase " omar fait manger les poulets " , on sait que quelque chose est en train de manger quelque chose . Cependant on ne peut pas savoir si les poulets sont en train de manger quelque

B) Les différentes ambiguïtés des langues naturelles:

a) L'ambiguïté lexicale:

Certains mots comme " côte " , " avocat " , " suite " , ont un sens différent selon le contexte . Ainsi la question "est-ce-que la pêche était bonne ?" peut s'adresser à un pêcheur , mais aussi à une autre personne qui vient de manger un fruit .

b) L'ambiguïté structurale dépasse le stade du mot isolé . Par exemple , il est possible de définir "il faut le pouvoir" de deux manières différentes , selon qu'on considère "pouvoir" comme un nom et "le" comme l'article qui s'y rapporte ou bien "pouvoir" comme un verbe et "le" comme pronom neutre objet , qui représente l'idée continue dans la proposition précédente . Ainsi selon que l'on analyse grammaticalement, cette phrase de la première ou de la deuxième façon, elle n'aura pas du tout le même sens. On retrouve ce même problème dans la phrase " la petite brise la glace " (voir figure (1)).

L'ambiguïté de structure profonde est encore plus subtile : elle apparaît lorsqu'il est possible d'interpréter la même phrase de deux façons différentes sans que la grammaire apparente en soit modifiée pour autant. Par exemple, d'après la phrase " omar fait manger les poulets " , on sait que quelque chose est en train de manger quelque chose . Cependant on ne peut pas savoir si les poulets sont en train de manger quelque

phrase : " lorsqu'une belle lune met fin à une morne journée ,
la suivante sera plus belle " , s'agit-il d'une plus belle
journée ou d'une plus belle lune?

Le rôle du contexte :

La plus part des phrases qui sont ambiguës pour l'ordinateur nous paraissent cependant très claires, grâce à notre connaissance du contexte. Ainsi, la phrase " il m'a fait mourir de rire " n'est pas ambiguë car chacun sait que l'auteur de cette phrase est toujours vivant. De même, " il y a un homme à l'entrée avec un chapeau vert " n'est pas une phrase ambiguë car chacun sait que l'entrée d'une pièce ne porte pas de chapeau. Dès qu'on se penche sur la question on s'aperçoit que sans cette connaissance du contexte, la plus part des phrases sont ambiguës.

C) La reconnaissance de la parole :

La reconnaissance de la parole s'insère dans le cadre plus générale de la reconnaissance des formes visuelles et tout particulièrement des caractères de l'écriture.

Le problème se complique lorsque les images sont floues ou légèrement déformées par rapport aux images de références. La notion d'identification entre une image donnée et une image type doit être remplacée par celle de parenté ou de voisinage, qualifiée par la grandeur dénommée distance: l'image donnée appartient à la classe de l'image type la plus voisine. Intervient une autre complication lorsque les images à reconnaître sont imbriquées les unes dans les autres ou simplement juxtaposées c'est le cas de l'écriture manuscrite, de la parole, des photographies tout système de reconnaissance est constitué de trois organes distincts; un capteur fournit les " images " à reconnaître ; un organe de prétraitement effectue les cadrages, lissage, normalisation ou l'extraction de paramètres; et un dispositif de décision et de classification mémorise les critères de classification des formes, et les règles de parenté des images déformées.

1-2-Extraction et représentation des connaissances:

La tâche d'extraction des connaissances est en fait, du ressort de "l'ingénieur cognitif", personne formée spécialement dans ce but.

Nous avons assuré son rôle, auprès d'experts confirmés dans la reconnaissance de la parole.

Les premiers échanges nous ont permis de nous familiariser avec le domaine choisi, d'avoir une vision globale des facteurs intervenant, et de réfléchir à la manière d'aborder la reconnaissance de la parole.

-Comment maîtriser l'ampleur du domaine ?

-Comment rester dans le cadre des objectifs généraux ?

Comment étendre ou restreindre le domaine ?

Les premiers contacts montrent qu'il est difficile pour un expert d'évaluer ce que l'on attend de lui, de même qu'il a été difficile pour nous d'expliquer exactement ce que nous recherchions dans un domaine inconnu.

Il semble bon à ce stade d'apporter à l'expert une information générale sur les systèmes experts, et leurs intérêts.

a-Extraction des connaissances:

Durant cette étape la difficulté réside dans le fait:

-D'éviter que le sujet abordé ne sorte des limites du domaine.

-La recherche d'objets de la connaissance manipulés par l'expert et leur correspondance avec une des représentations étudiées dans la partie théorique.

-Déterminer les éléments essentiels qui guident le raisonnement de l'expert.

b- Représentation des connaissances:

Le formalisme de la base de connaissances retenu est celui des règles de productions, pour les raisons suivantes:

- Elles permettent à l'expert d'exprimer par un moyen naturel ses connaissances?*
- Elles aident l'expert à organiser son savoir, ses stratégies de raisonnement.
- Les règles étant indépendantes entre elles; permettent d'ajouter ou de supprimer aisément des règles durant la prise de connaissance.
- Les règles permettent d'exprimer les connaissances granulaires (règles et métaconnaissances (métarègles) d'une manière uniforme.

Enfin, tout type de règles de productions peut se ramener à une forme unique de modélisation:

SI <condition (s)> ALORS <conclusion>

<condition (s)> ::= <condition 1> ET <condition 2> ... <condition N>
/ <condition>.

EXEMPLE :

SI <condition 1> ET <condition 2> ALORS <conclusion 1> OU <conclusion 2>

est équivalent aux deux règles suivantes:

SI condition 1 ET condition 2 ALORS conclusion 1

SI condition 2 ET condition 2 ALORS conclusion 2

Exemple de règle:

SI la voyelle est vocalique

SI le premier formant se situe entre 150 et 350 HZ

SI le 2 ème formant se situe entre 500 et 700 HZ

SI les 2 premiers formants sont séparés d'au moins 350 HZ ALORS C'est " U "

4- Validation :

Une première étape consiste à valider par l'expert les connaissances modélisées sous forme de règles de productions.

Une seconde étape de validation sera faite lorsque le premier prototype du système expert sera réalisé.

Cette validation permettra à l'expert de confirmer la modélisation de ses stratégies de raisonnement.

Le système Expert d'aide à la reconnaissance de la parole SEDARP permet les deux applications suivantes

-Consultation de SEDARP par un utilisateur.

-Modification de la base de connaissance par l'expert.

II-2 STRUCTURES DE DONNEES UTILISEES EN TURBO.PROLOG

Turbo prolog permet différentes possibilités, en outre la programmation modulaire des éléments suivants:

- Interface utilisateur/SEDARP (" IUTILIS.P80")
- Interface expert/SEDMDO - (" IEXPERT.PRO ").
- Moteur d'inférences . (" MOTEUR.PRO ").

II-2.a Dictionnaire

Mémoire de travail à long terme, la structure de données associée au dictionnaire est un fichier datadase (" DIC.DBA ") .

Ce fichier regroupe l'ensemble des sujets, verbes et compléments utilisés dans le domaine de la reconnaissance de la parole.

*Représentation d'un sujet:

Un sujet est représenté par le prédicat sujet déclaré en database comme suit:

sujet (symbol)

EXEMPLE: sujet (la voyelle) .

*Représentation d'un verbe:

un verbe est représenté par un prédicat verbe déclaré en database comme suit:

verbe (symbol)

EXEMPLE : verbe (est)

*Représentation d'un complément:

Un complement est représenté par le prédicat complément déclaré en database comme suit
complement (symbol)

EXEMPLE: complement (nasale).

II-2. b Base de faits:

Mémoire de travail à court terme, elle permet de regrouper dans une structure de fichier database (" BF.DBA "), l'ensemble des faits:

-Donnés initialement par l'utilisateur.

-Établie par le moteur d'inférences.

*Représentation d'un fait:

Un fait est représenté par un prédicat à deux arguments:

fait (L1,L2) déclaré en database.

-L1:liste exprimant une assertion.

-L2:liste exprimant la valeur logique de l'assertion.

EXEMPLES: fait([voyelle,est nasale],[vrai]).

fait([consonne ,explosive],[faux]).

II-2.c Base de connaissance:

Mémoire de travail à long terme, la structure de données associée à la base de connaissance est un fichier database (" BC.DBA ").

Ce fichier regroupe l'ensemble des métarègles et règles rescencées au cours de la saisie de connaissances.

Une représentation interne adéquate des règles et métarègles est très importante. En effet, la figure de la page suivante illustre les critères de choix d'une représentation interne adéquate. (RAIR, 81).

*Représentation d'une règle:

Une règle est représenté par un prédicat à trois arguments déclaré en database :
règle (arg1, arg2, arg3).

-Arg1: liste à deux éléments entiers désignant respectivement le numéro de la règle, et son nombre de premisses.

-Arg2: liste de premisses séparées par des virgules (exprimant la conjonction "et").

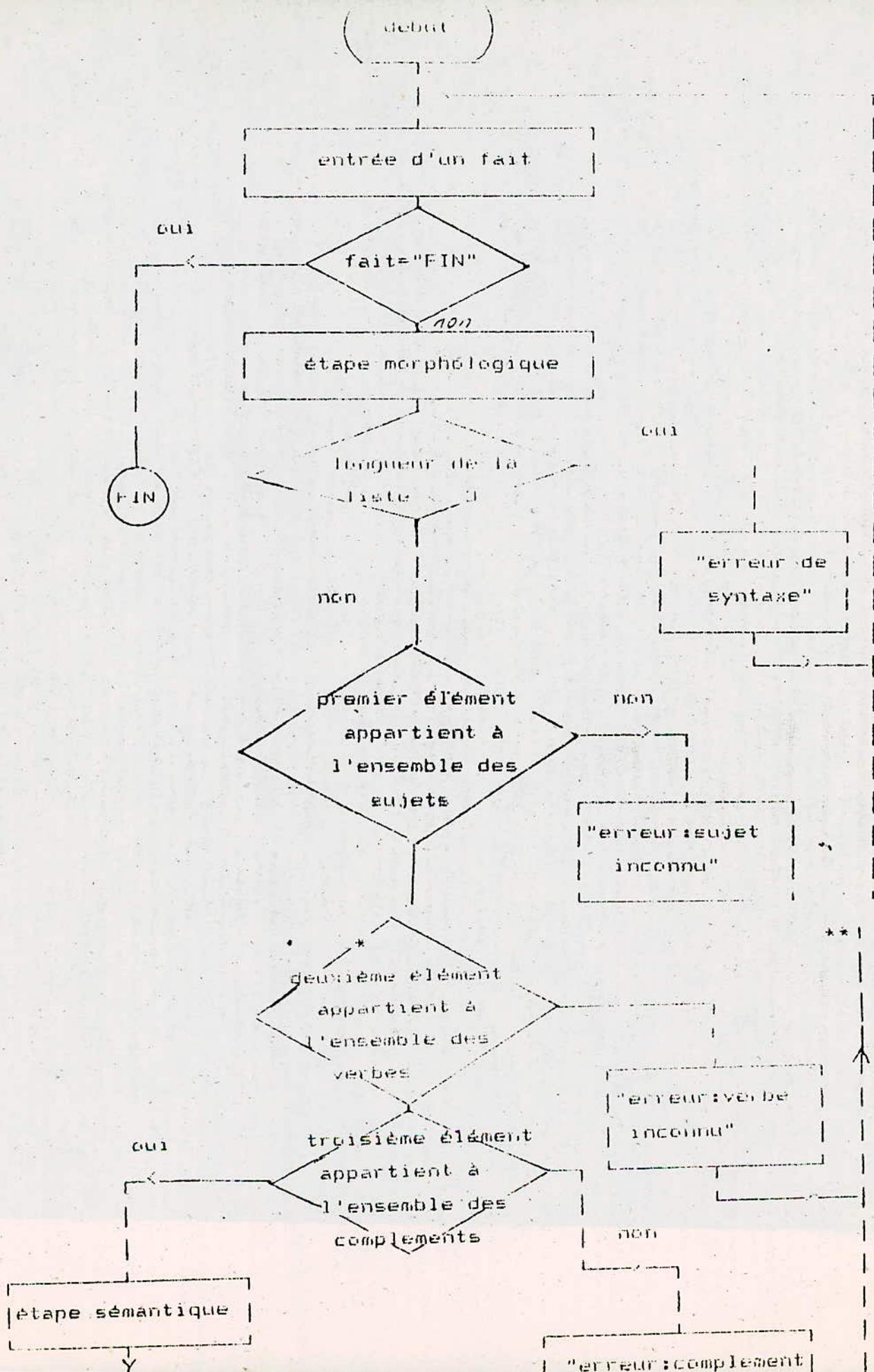
-Arg3: liste exprimant la conclusion de la règle.

EXEMPLE: La règle suivante (règle numéro 6)

SI le voyant-pret n'est pas allumé
et le voyant-papier n'est pas allumé
et le voyant-pret n'est pas allumé
et le changement-de-ligne n'est pas effectué
ALORS la carte-de-contrôle est défectueuse.

possède la représentation suivante:

règle ([6,4], [[voyant-pret, allumé], [faux]], [voyant-papier, allumé], [faux]],
[[voyant-pret, allumé], [faux]], [changement-de-ligne, effectué], [[carte-de-
contrôle, est, défectueuse], [vrai]]).



Ce qui donne la réponse suivante du moteur d'inférence:

La règle courante est la suivante:

Si

Le voyant-tension est allumé

et

Le voyant-pret n'est pas allumé

et

Le voyant-papier n'est pas allumé

et

Le changement-de-ligne n'est pas effectuée

Alors la carte-de-contrôle est défectueuse.

III-4 INTERABEE EXPERT/SEDMDO:

Cette interface permet à l'expert d'enrichir la base de connaissance (BC) avec de nouvelles règles ou bien, de modifier des règles existantes déjà dans BC.

Les deux seules transactions permises à un expert sont:

- L'ajout de règle (s).
- La suppression de règle(s).

Ainsi, la modification de règles, se traduit par la suppression de son ancienne version, suivi de l'ajout de sa nouvelle version.

II- 5) MOTEUR D'INFÉRENCES :

5.1- Objets traités par le moteur d'inférences :

Le moteur d'inférences SEDARP est d'ordre 0 et traite de quatre sortes d'objets :

a- Les paramètres:

Ce sont les fait donnés initialement par l'utilisateur ou déduit durant la consultation .

Dans le domaine de la reconnaissance de la parole, un paramètre est vrai ($X=1$) ou faux ($X=-1$) avec certitude .Ceci exclut la possibilité d'exprimer une information avec un facteur de certitude .

b- Les règles :

Elles sont exploitées par le moteur d'inférences en chainage arrière.L'ensemble des règles récoltées auprès de l'expert peuvent être exploitées en chainage arrière ou avant ;notre choix du type de chainage est donc fondé sur le fait que le

II- 5) MOTEUR D'INFÉRENCES :

5.1- Objets traités par le moteur d'inférences :

Le moteur d'inférences SEDARP est d'ordre 0 et traite de quatre sortes d'objets :

a- Les paramètres:

Ce sont les fait donnés initialement par l'utilisateur ou déduit durant la consultation .

Dans le domaine de la reconnaissance de la parole, un paramètre est vrai ($X=1$) ou faux ($X=-1$) avec certitude .Ceci exclut la possibilité d'exprimer une information avec un facteur de certitude .

b- Les règles :

Elles sont exploitées par le moteur d'inférences en chainage arrière.L'ensemble des règles récoltées auprès de l'expert peuvent être exploitées en chainage arrière ou avant ;notre choix du type de chainage est donc fondé sur le fait que le

chainage arrière est compatible avec le TURBO-PROLOG .

5-2) CONCEPTION DU MOTEUR D'INFÉRENCES :

Comme il a été précisé au niveau de la conception globale de SEDARP ,le moteur d'inférences est constitué de cinq modules principaux :

- a- Le module de sélection ou restriction
- b- Le module de filtrage ou "pattern matching"
- c- Le module de résolution de conflits
- d- Le module de déclenchement d'une règle
- e- Le module de mise à jours des mémoires à court termes .

Les modules (b,c,d,e) constituent un cycle du moteur d'inférences .Une consultation de SEDARP correspond à plusieurs cycles successifs .

L'organnigramme suivant illustre le fonctionnement générale du moteur d'inférences.

a-Sélection :

Dans cette première étape, le moteur d'inférences utilise exclusivement la métabase (ensemble des métarègles).

Les métarègles permettent de reconnaître la base de connaissances appropriée à la situation courante .

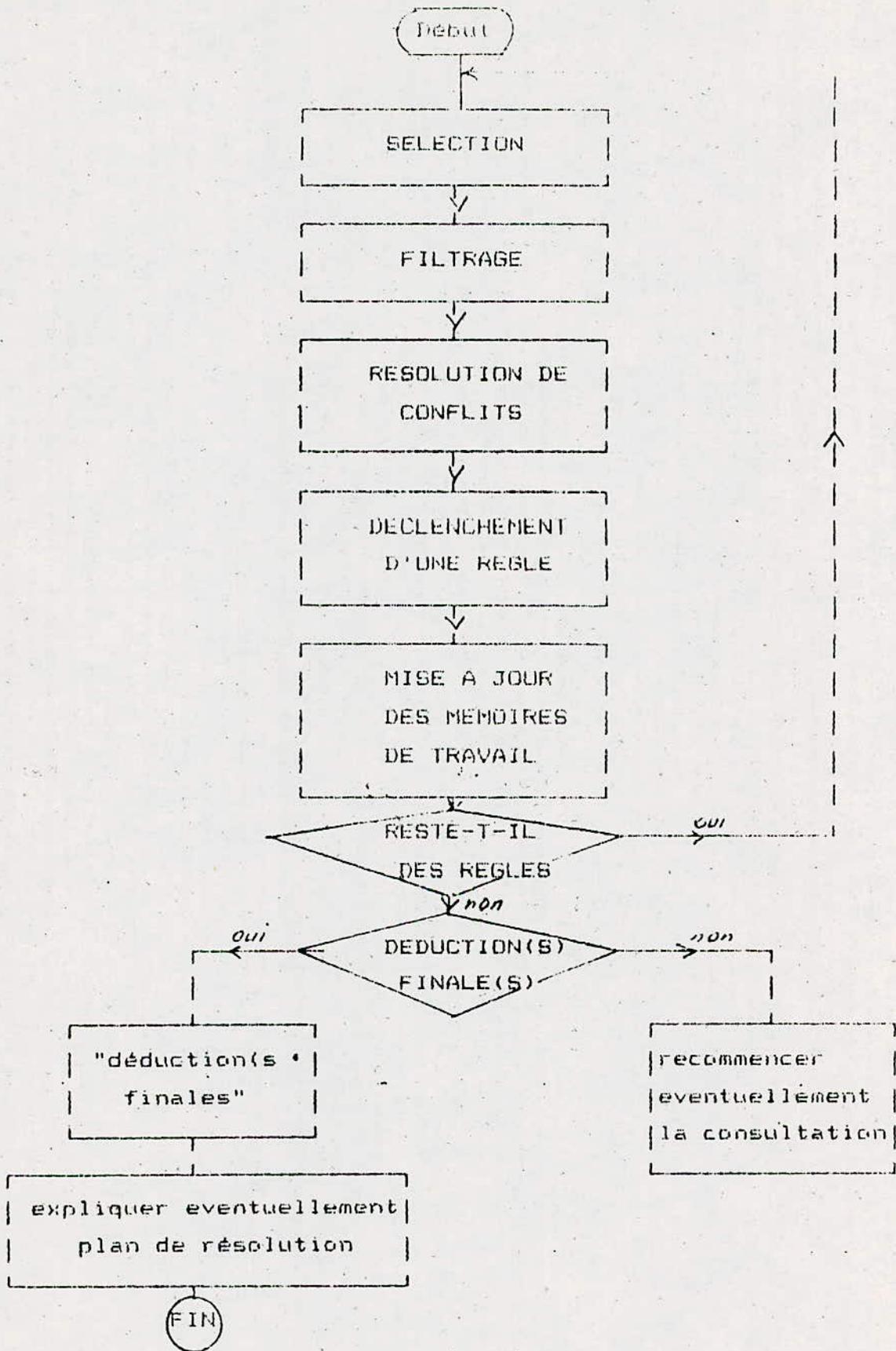


FIGURE 12. Schéma général du moteur d'inférences

En deuxième étape, le moteur d'inférences sélectionne le groupe de règles appropriées au cas courant : autrement dit, de localiser le phonème à un niveau particulier, et donc de ne garder en mémoire que le groupe de règles adéquat.

L'étape de sélection est schématisée par le second organigramme.

Remarque:

A la fin de l'étape de sélection, le moteur d'inférence demande à l'utilisateur s'il a observé (ou entendu) des symptômes particuliers sur le spectrographe. Si l'utilisateur répond par " oui " , le moteur fait appel à l'interface utilisateur d'entrée. Une fois que l'utilisateur a terminé l'entrée de ces faits, il introduit " fin " pour indiquer au moteur d'inférences qu'il n 'a plus d'informations à fournir.

b- Filtrage :

Quelque soit le type de raisonnements choisis, la

construction d'un moteur d'inférence comporte deux étapes particulièrement délicates:

- L'étape de filtrage et le choix de la règle à déclencher.

Le filtrage des règles candidates à chaque cycle nécessite

la confrontation de chacune d'elles avec l'ensemble des faits de la base de faits (BF).

Cette operation étant coûteuse en temps machine, différents filtres peuvent être constuits, qui éliminent avant le filtrage proprement dit, les règles inutiles.

Une règle est jugée inutile et donc supprimée de la base de règles (BR), si l'une de ses prémisses est en contradiction avec un fait de BF.

Enfin, pour chaque fait appartenant à BF, il faut décrementer le nombre de prémisses de toute règle de BR qui mentinne ce fait.

L'organigramme suivant reprend avec plus de détails les étapes ci-dessus.

c- Résolution de conflits et choix d'une règle :

Ce problème concerne la " résolution de conflits " entre toutes les règles résultantes de l'étape de filtrage.

La stratégie de choix est évidemment vitale, non seulement pour les performances, mais également et surtout pour la capacité du système à " comprendre " ce qu'il fait.

Les métarègles utilisées représentent, la stratégie adoptée par l'expert pour reconnaître un phonème .

L'expert distingue quatre niveaux hiérarchiques:

- Voyelles
- Consonnes
- Semi-voyelles

En observant les symptômes spectrales il peut localiser la nature du phonème.

À chaque niveau, est associée une métarègle indiquant quels symptômes localisent le phonème.

*Représentation interne d'une métarègle:

Une métarègle est représentée par le predicat metal (NM,LS,LR) déclaré en database ou :

- NM est le numéro de la métarègle.
- LS est la liste des symptômes particuliers à l'élément associé à cette métarègle.
- LR est la liste des règles à considérer.

II-2.d Trace:

Mémoire de travail à court terme, elle permet de regrouper dans une structure de fichier database ("TRACE.DBA "), le plan de résolution du moteur d'inférences.

À chaque déclenchement d'une règle par le moteur d'inférences, celle-ci est sauvegardée dans "TRACE.DBA " .

Ainsi, le moteur peut expliquer éventuellement (à la demande de l'utilisateur) son plan de résolution.

II.3 INTERFACE UTILISATEUR/SEDARP

L'interface utilisateur SEDARP ("IUTILIS.PRO") permet d'accepter les données de fournir des résultats, et des explications dans un langage quasi-naturel.

La compréhension du langage naturel avec toutes ses formes de syntaxes grammaticales étant un problème très complexe, nous avons dû restreindre la syntaxe selon la notation BNF suivante:

(fait) ::= (assertion-affirmative) / (assertion-négative).

(assertion-affirmative): := (sujet) (verbe) (complement).

(assertion-négative): := (sujet) non (complement).

(sujet): := voyelle

(verbe): := est /

(complement): := vocalique /

Remarque: les sujets composés sont séparés par un trait d'union.

Exemple de faits corrects:

-voyelle est nasale: fait exprimant une assertion de valeur logique égale à vrai.

-consonne non explosive: fait exprimant une assertion de valeur logique égale à

faut .

II-3.a Interface d'entrée:

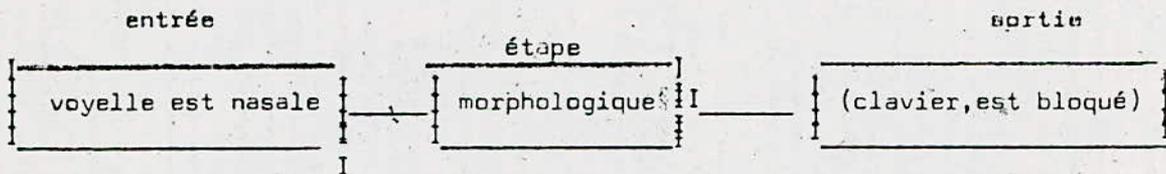
A chaque fait entrée par l'utilisateur, l'interface d'entrée exécute les différentes étapes suivantes:

1. Etape morphologique

Durant cette étape, sont identifiés les mots constituant le fait introduit par l'utilisateur.

L'exécution de cette étape consiste à transformer le fait entré sous forme de chaîne de caractères en une liste de mots.

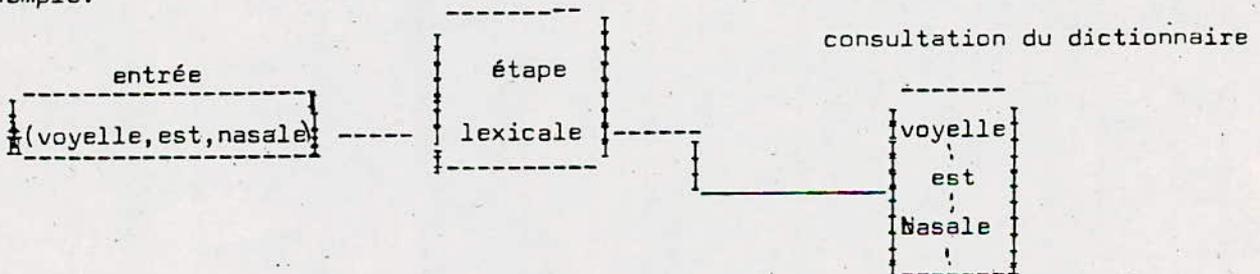
Exemple:



2- Etape lexicale:

Durant cette étape les mots résultants de l'étape morphologique sont recherchés dans le dictionnaire afin d'être identifiés comme valides ou non.

Exemple:



Dans ~~cette~~ le cas où un mot n'est pas retrouvée dans le dictionnaire, un message d'erreur spécifiant que le mot utilisé est un mot inconnu et de mande à l'utilisateur de réintroduire le fait en utilisant un autre mot. (retour à (1)).

3-Etape syntaxique:

Dans cette étape a lieu la vérification de la syntaxe grammaticale du fait entré.

Il faut vérifier d'une part que la longueur de la liste est égale à trois, ce qui exprime que l'utilisateur a bien utilisé l'une des deux formes de syntaxes.

D'autre part, (lorsque la longueur est correcte), il faut vérifier que chaque position d'un mot respecte la syntaxe:

- Le premier mot de la liste est un sujet.
- Le deuxième mot de la liste est verbe ou une négation.
- Le troisième mot de la liste est un complément.

Une erreur à cette étape, renvoie un message d'erreur spécifique (syntaxe non respectée); l'utilisateur doit réintroduire son fait en respectant la syntaxe. (retour à 1).

4-Etape sémantique:

Cette étape se charge de traduire la structure (liste simple) de l'étape syntaxique en une structure (liste de liste) exprimant le sens du fait entré.

Le sens est exprimé dans le langage de la logique des propositions, qui consiste à attribuer à chaque assertion une valeur logique égale à vrais ou faux.

En deuxième étape, le moteur d'inférences sélectionne le groupe de règles appropriées au cas courant : autrement dit, de localiser le phonème à un niveau particulier, et donc de ne garder en mémoire que le groupe de règles adéquat.

L'étape de sélection est schématisée par le second organigramme.

Remarque:

A la fin de l'étape de sélection, le moteur d'inférence demande à l'utilisateur s'il a observé (ou entendu) des symptômes particuliers sur le spectrographe. Si l'utilisateur répond par " oui " , le moteur fait appel à l'interface utilisateur d'entrée. Une fois que l'utilisateur a terminé l'entrée de ces faits, il introduit " fin " pour indiquer au moteur d'inférences qu'il n 'a plus d'informations à fournir.

b- Filtrage :

Quelque soit le type de raisonnements choisis, la

construction d'un moteur d'inférence comporte deux étapes particulièrement délicates:

- L'étape de filtrage et le choix de la règle à déclencher.

Le filtrage des règles candidates à chaque cycle nécessite

la confrontation de chacune d'elles avec l'ensemble des faits de la base de faits (BF).

Cette operation étant coûteuse en temps machine, différents filtres peuvent être constuits, qui éliminent avant le filtrage proprement dit, les règles inutiles.

Une règle est jugée inutile et donc supprimée de la base de règles (BR), si l'une de ses prémisses est en contradiction avec un fait de BF.

Enfin, pour chaque fait appartenant à BF, il faut décrementer le nombre de prémisses de toute règle de BR qui mentionne ce fait.

L'organigramme suivant reprend avec plus de détails les étapes ci-dessus.

c- Résolution de conflits et choix d'une règle :

Ce problème concerne la " résolution de conflits " entre toutes les règles résultantes de l'étape de filtrage.

La stratégie de choix est évidemment vitale, non seulement pour les performances, mais également et surtout pour la capacité du système à " comprendre " ce qu'il fait.

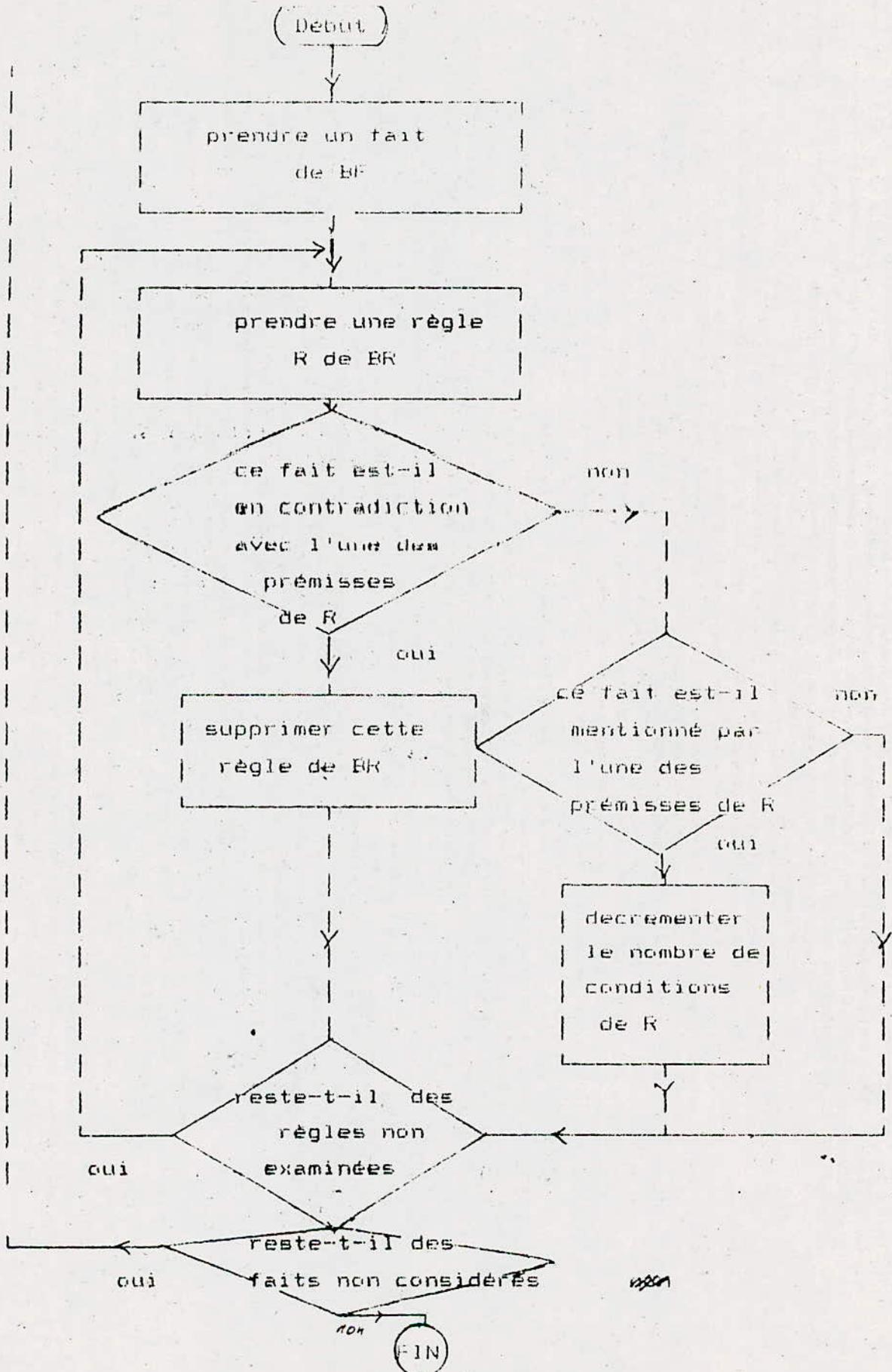


FIGURE 14. Etape de filtrage

La recherche exhaustive est écartée, car elle est coûteuse en temps machine et très peu performante.

Le contrôle par métarègles nécessite un niveau de détails encore plus fin dans la stratégie de raisonnement de l'expert. Or, à ce niveau de détails, l'expert agit presque par intuition, d'où la difficulté d'instaurer des métarègles.

C'est donc le choix par évaluation qui est adopté pour le moteur d'inférences.

Le critère adopté est celui de la précision. C'est à dire le choix de la règle qui a le plus de prémisses vérifiées dans la base de faits.

Ceci revient à choisir la règle avec le plus petit nombre de condition. L'organigramme n° 4 résume l'étape de résolution de conflits.

Remarque :

- Dans le cas d'égalité de nombres de conditions, la première

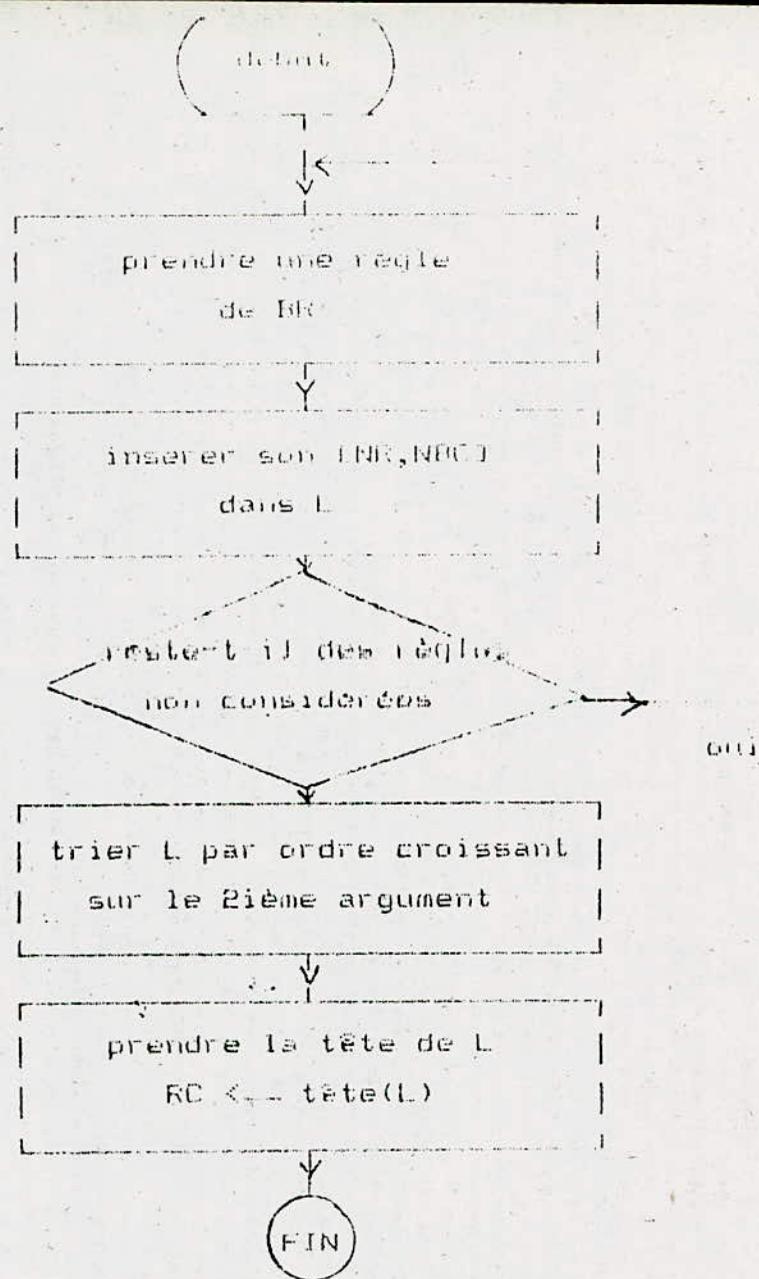


FIGURE 15. Etape de résolution de conflits

GLOSSAIRE :

- RC: règle courante
- NBC: nombre de conditions
- L: liste de [NR, NBC]

La recherche exhaustive est écartée, car elle est coûteuse en temps machine et très peu performante.

Le contrôle par métarègles nécessite un niveau de détails encore plus fin dans la stratégie de raisonnement de l'expert. Or, à ce niveau de détails, l'expert agit presque par intuition, d'où la difficulté d'instaurer des métarègles.

C'est donc le choix par évaluation qui est adopté pour le moteur d'inférences.

Le critère adopté est celui de la précision. C'est à dire le choix de la règle qui a le plus de prémisses vérifiées dans la base de faits.

Ceci revient à choisir la règle avec le plus petit nombre de condition. L'organigramme n° 4 résume l'étape de résolution de conflits.

Remarque :

- Dans le cas d'égalité de nombres de conditions, la première

règle est choisie .

d - Déclenchement d'une règle :

Pour déduire la conclusion d'une règle choisie (RC), il faut que l'ensemble de ses prémisses soient satisfaites en égard de la base de faits.

Si l'ensemble des prémisses de RC sont satisfaites, le degré de satisfaction de la partie prémisses de RC est alors totale, d'où la règle RC est immédiatement déclenchable.

Dans le cas contraire le moteur d'inférences tente de rendre total ce degré de satisfaction . Pour cela, il doit poser des questions à l'utilisateur.

Les questions concernent l'ensemble des prémisses de RC non encore établies.

Tout au long de la résolution du problème, le moteur d'inférences conserve une trace de toutes ces connaissances ceci dans le but d'être capable de savoir quelle question poser, et expliquer son raisonnement à l'utilisateur.

Enfin le moteur d'inférence de SEDARP procède à une exploration de l'arborescence en " profondeur d'abord ", et adopte une stratégie de contrôle irrévocable.

Une fois qu'il déclenche une règle , celle-ci est détruite de la base de règles (BR) courante, car elle ne sera plus exploitable . L'organigramme suivant traduit l'étape de déclenchement d'une règle.

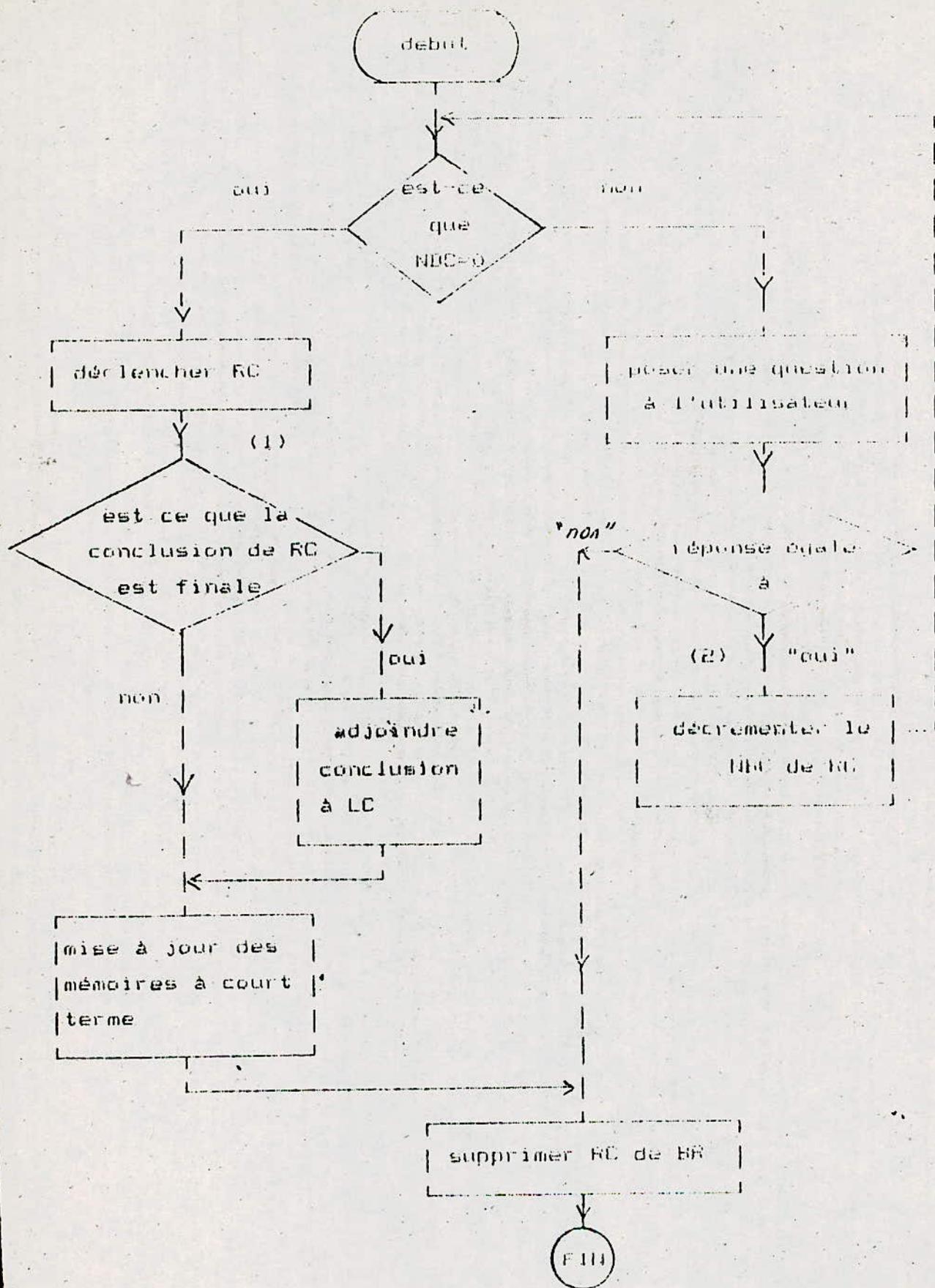


FIGURE 16. Etape déclenchement d'une règle

L'étude que j'ai pu faire sur les systèmes experts, m'a permis tout d'abord d'apprendre le langage TURBO-PROLOG, langage de l'IA par excellence, de maîtriser le domaine de la reconnaissance spectrale des phonèmes, d'avoir une idée globale sur la reconnaissance de la parole sous différents angles, et surtout d'avoir une bonne connaissance théorique sur les systèmes experts.

En ce qui concerne, la réalisation du SE pour la reconnaissance vocale, je ne peut pas dire que j'ai réussi à le réaliser, mais vu que le domaine a été restreint, j'ai pu réaliser un programme en TURBO-PROLOG, pour la reconnaissance de 30 phonèmes de la langue française.

- 1- SYSTEME EXPERT : Applications
J. B. NOEL Edition Masson 1989
- 2- INTELLIGENCE ARTIFICIELLE : Promesse et réalités
Alain BONNET Edition Eyrolles 1988
- 3- THEORIE DES SYSTEMES EXPERTS
B. WAYNE Edition Mc Graw Hill 1989.
- 4- SYNTHESE, RECONNAISSANCE DE LA PAROLE
MARC FERRETTI Edition Tests 1983
- 5- LES INDICES ACOUSTIQUES DE LA PAROLE
PIERRE DELATTRE EDITION MOUTON
- 6- VUES NOUVELLES SUR LE MONDE DES SONS
F. WINCKEL Edition Dunod
- 7- L'ACOUSTIQUE
S. S. STEVENS et F. WARSHOFSKI Edition Laffront Sciences
- 8- TURBO-PROLOG : Une introduction à l'intelligence
 "artificielle"
P. BIHAN Edition Eyrolles
- 9- OUTILS LOGIQUES POUR L'IA
J. P. DELAHAYE Edition Eyrolles
- 10- SEMANTIQUE LOGIQUE
J. P. DELAHAYE Edition Eyrolles
- 11- INTRODUCTION TO LOGIC PROGRAMMING
C. HOGGER Edition Masson
- 12- LOGIC FOR PROBLEM SOLVING
R. KOWALSKI Edition North Holland

