

وزارة التعليم والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE



ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

S U J E T

UN SYSTEME EXPERT POUR LA LECTURE DE SPECTROGRAMMES

Proposé par :

Etudié par :

Dirigé par :

M^r BOUSSEKSOU B.

M. BEN ABDESSELAM

M^r B. BOUSSEKSOU

F. H. MANSOURI

PROMOTION : JUIN 88

DEDICACES

Je dédie ce modeste travail:

A la mémoire de mon grand-père,
A ma grand-mère,
A mes adorables parents,
A mes frères et soeurs,
A ma tante,
A mes cousins et cousines,
A LYNDA,
A tous mes amis,
A ma nièce,
A mon collaborateur de cette année.

F.H.MANSOURI

Je dédie ce modeste travail

A mes parents, pour leur soutien et pour tout
ce qu'il m'ont donné durant mes études,
A mes frères et soeurs,
A tous mes amis (es),
A mon binôme.

B. Mustapha

SOMMAIRE

	Page
-INTRODUCTION	1
PARTIE 1	
I-RECONNAISSANCE DE LA PAROLE	2
1- Introduction	2
2- Problèmes de la reconnaissance automatique de la parole.....	2
3- Traitement	4
3-1- Analyse du signal	
3-2- Comparaison	
4- Méthode de reconnaissance.....	6
5- Reconnaissance	6
5-1- Reconnaissance globale	
5-2- Reconnaissance analytique	
6- Intérêt de l'intelligence artificielle pour la parole.....	9
7- Conclusion	9
II- LECTURE DES SPECTROGRAMMES	
1- Introduction	10
2- Définition	10
3- Lecture et intérêt des spectrogrammes.....	10
3-1- Intérêt	
3-2- Lecture	
4- Spectrogrammes analogiques et numériques.....	11
5- Conclusion	12
III- DECODAGE ACOUSTICO_PHONETIQUE	
1- Introduction	14
1-1- Les phones	
1-2- Les phonèmes	
1-3- Les diphonèmes	
2- Stratégie pour la segmentation.....	15
2-1- Segmentation en reconnaissance automatique	
3- Evénements vocaliques évidents.....	15
3-1- Macro-classe VOC	
4- Evénements consonantiques évidents.....	16
4-1- Macro-classe FRIC	
4-2- Macro-classe OCC	
4-3- Macro-classe CONS	
5- Conclusion	17
PARTIE 2	
IV- INTELLIGENCE ARTIFICIELLE ET SYSTEMES EXPERTS	
1-Introduction	19
2-Représentation des connaissances.....	20
2-1- Définition	
2-2- Notion de logique	
-Logique des propositions	
-Calcul des prédicats	

2-3- Règles de production	
2-4- Réseaux sémantiques	
2-5- Systèmes de production	
3- Systèmes experts	24
3-1- Principe	
3-2- Cycle de fonctionnement	
3-3- Modes d'invocation des règles	
3-4- Modes d'inférences	
3-5- Techniques de recherche	
4- Langage	29
5- Présentation du système	31
5-1 Principes généraux	
V-CONCEPTION DU SYSTEME	
1- Introduction	33
2- Représentation des connaissances interprétés	33
2-1- Représentation des règles dans la base de règles	
2-2- Représentation d'un fait dans la base de faits	
3- Cycle de fonctionnement du moteur	36
3-1- Phase sélection	
3-2- Phase résolution de conflits et filtrage	
3-3- Phase exécution	
3-4- Phase question	
3-5- Phase d'arrêt du moteur	
3-6- Phase impression	
4- Module de dialogue	43
VI- VOCABULAIRE DE DESCRIPTION	
1- Introduction	44
2- Description du spectrogramme et base de faits	44
2-1- Aspect global	
2-2- Aspect segmental	
3- Typologie des règles	48
3-1- Importance du contexte	
3-2- Règles d'identification	
4- Conclusion	52
VII- FONCTIONNEMENT DU SYSTEME	
1- Introduction	54
2- Cycle de fonctionnement	54
3- Utilisation des classes phonétiques	55
4- Question à l'utilisateur	55
5- Traitement de l'incertain	56
5-1- La logique floue	
5-2- La méthode des coefficients de vraisemblance	
6- Module de dialogue	58
7- Solution adoptée	59
VIII- CONCLUSION ET PERSPECTIVES	
	59
ORGANIGRAMMES	60

BIBLIOGRAPHIE

ANNEXE

- Spectrogramme
- Description
- Ensembles de règles
- Code utilisé

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Introduction

Dans le cadre de la communication parlée entre un homme et une machine, diverses sources de connaissances peuvent être utilisées tant au niveau acoustique, phonétique, lexical, que le niveau syntaxique ou pragmatique. Chaque source de connaissances permet de fixer des contraintes lors de l'étape de reconnaissance du signal de parole et ainsi faciliter la tâche. Il a été néanmoins établi que pour qu'un système de reconnaissance de la parole continue soit exploitable, la reconnaissance acoustique doit être très fiable. Cependant, malgré l'extrême variabilité du signal de parole, quelques experts phonéticiens sont capables de déchiffrer une représentation visuelle de la parole, le spectrogramme, avec de meilleures performances que les systèmes actuels.

La maîtrise de cette expertise présente un intérêt capital pour le décodage acoustico-phonétique et l'usage de techniques d'intelligence artificielle semble approprié à la résolution de ce type de problèmes.

Nous présentons ici un système de reconnaissance permettant de modéliser le raisonnement d'un expert phonéticien.

Nous décrivons dans une première partie les problèmes de la reconnaissance automatique de la parole, l'intérêt que présente l'emploi des techniques d'intelligence artificielle. Certaines de ces réalisations:

- La lecture de spectrogrammes.
- Le décodage acoustico-phonétique.

La deuxième partie est consacrée à la formalisation des connaissances:

- Le langage de description du spectrogramme.
- La conception du système.
- Le fonctionnement du système.

Et nous terminons par une conclusion et des perspectives.

1ere PARTIE
Reconnaissance de la parole

I Reconnaissance de la parole

Problèmes de la reconnaissance automatique de la parole
Traitement
Méthodes de reconnaissance
Intérêt de l'intelligence artificielle pour la reconnaissance

1- Introduction

La reconnaissance de la parole reste jusqu'à nos jours un problème crucial dans la communication parlée. Si une phrase en général ne s'écrit que d'une seule manière, sa prononciation peut varier à l'infini.

Ainsi l'objet de ce chapitre est la définition des grands axes d'une stratégie générale mise en oeuvre dans un système de reconnaissance automatique de la parole.

2- Problèmes de la reconnaissance automatique de la parole*

Il peut paraître étonnant qu'après plus de trente années de recherche, le problème de la R.A.P ne soit pas encore résolu dans sa généralité. Mais actuellement le problème est plus souple.

Les difficultés fondamentales résident dans:

*La reconnaissance automatique de la parole est notée par la suite R.A.P

* La redondance des données:

La R.A.P traite des données en très grand nombres et souvent très redondantes, la quantité d'information à traiter, contenue dans un signal de parole est, en effet, très largement supérieure à celle transmise d'un point de vue phonétique. Il faut donc trouver une représentation opératoire de la parole permettant de réduire au maximum cette masse de données, tout en conservant l'information pertinente pour la tâche de reconnaissance.

* La variabilité:

On peut concevoir, en première approximation le signal de parole comme une suite de segments qui se concatènent dans le temps. Chaque segment peut être mis en correspondance avec une unité linguistique, le phonème par exemple, celui-ci est supposé présentant une articulation bien précise. A priori, les possibilités articulatoires de la production de sons d'un être humain sont infinies, pourtant seul un nombre limité de phonèmes est utilisé pour produire des sons dans une langue donnée. Si le nombre d'unités linguistiques dans une langue est fini, les réalisations acoustiques de ces mêmes unités sont extrêmement variables. Le problème de la variabilité du signal acoustique provient du fait que la production d'un son est le résultat de la coarticulation d'un ensemble d'articulateurs.

Pour qu'un son soit perçu comme un phonème donné, il n'est pas nécessaire que l'articulation de celui-ci atteigne son état stable, le simple mouvement vers cet état semble suffire. [LIE 77]. Chaque articulateur doit non seulement réaliser le son en cours (influencé par le mouvement précédent), mais également préparer le suivant. C'est l'aspect continu de la production de la parole qui est la source de l'influence contextuelle d'un phonème, dont l'ensemble des phonèmes est souvent regroupé sous le terme de coarticulation. La forme acoustique d'un phonème varie donc en fonction de son contexte fig(1-1). Mais la coarticulation ne constitue pas la seule source de variabilité.

- Variabilité acoustique:

Elle peut tenir au changement d'environnement (bruit ambiant), à la position de la bouche par rapport au micro, aux caractéristiques du micro.

- Variabilité intra_locuteur:

Elle peut tenir au changement physiologique de la voix (chuchotée, normale, criée, etc...), aux états émotionnels (frayeur...), au soin mis dans l'articulation.

- Variabilité inter_locuteur:

Elle peut tenir à des différences sociologiques: accents, dialecte,...., ou physiologiques: formes différentes des conduits vocaux et fosses nasales, longueurs différentes des cordes vocales.

* Difficultés liées à l'utilisation des niveaux supérieurs:

Nous entendons par niveau supérieur ou "haut niveau":[REC 87]

- Lexical.
- syntaxique.
- Sémantique.
- Pragmatique.

Ces niveaux dépendent du type d'application considérée: dictée vocale, commande vocale ou dialogue.

Pour la dictée vocale par exemple, l'accès au lexique demande un temps de calcul très important, dans la mesure où l'on traite de très grand vocabulaires. Ce problème déjà complexe est encore rendu plus difficile car il faut tenir compte des erreurs dans le treillis phonétique (omissions, insertions). Dans le cadre du dialogue homme-machine, le langage oral obéit à une syntaxe différente de celle du langage écrit. Les phrases peuvent également contenir des mots mal prononcés ainsi que des mots d'appui (sans sens). Or une grammaire englobant toutes les possibilités du langage paraît aujourd'hui difficilement réalisable. Dans la pratique, pour qu'un dialogue soit possible avec une machine, il faut se restreindre à une tâche bien précise pour laquelle une grammaire artificielle est élaborée.

3- Traitement

Face aux problèmes de la R.A.P, plusieurs approches ont été envisagées. Ces dernières concernent le choix d'une représentation du signal de parole. Il existe d'autre part différents traitements adaptés à une forme de variabilité: au niveau de l'analyse du signal, de la comparaison entre formes acoustiques, etc.....

3-1- Analyse du signal:

Certaines techniques de traitement du signal utilisent des modélisations de la perception et de la production de la parole.

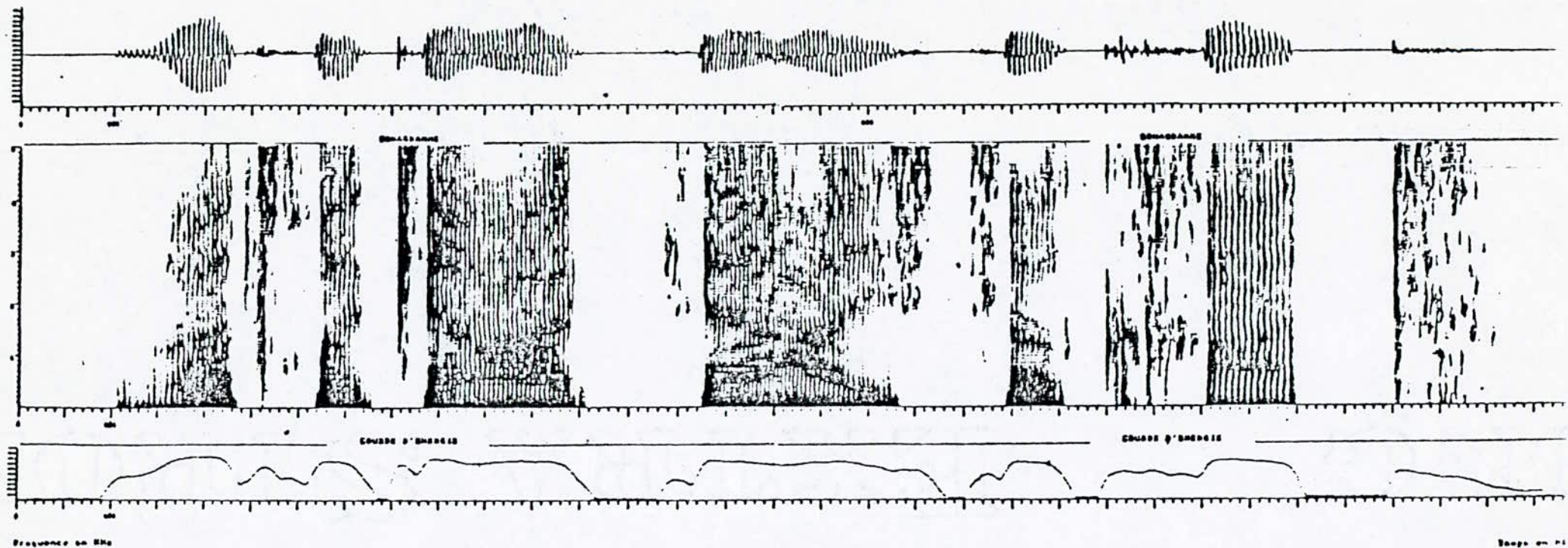
Nous citons parmi ces techniques:

- L'analyse cepstrale. [KUN 81].
- La transformée de fourier. [LIF 80].
- Prédiction linéaire. [HAS 86].
- Analyse spectrographique. [EME 77]....

Des échelles fréquentielles non linéaires en R.A.P sont couramment employées: Bark, Mel, imitant ainsi une partie du fonctionnement de l'oreille interne, qui effectue une analyse fréquentielle plus fine pour les basses fréquences que pour les hautes fréquences.

3-2- Comparaison:

Des techniques de programmation dynamique sont employées. Elles permettent en particulier, d'effectuer un alignement temporel entre deux formes à comparer. [KAUF 65].



v o t r e c a r a c t e r i s t o k r a t

Figure 1.1: Spectrogramme de: *vo*tre caractère aristocrate.

4- Méthodes de reconnaissance

Les méthodes de reconnaissance consistent à considérer un modèle de la parole, choisi essentiellement en fonction de ces caractéristiques soit par: des méthodes statistiques nécessaires à l'apprentissage de ces paramètres et des méthodes de recherche optimale pour effectuer la reconnaissance.

Soit à partir de connaissances acquises par l'expérience humaine ou de manière automatique.

5- Reconnaissance

On distingue dans le cadre de la R.A.P deux approches différentes:

5-1- approche globale

La reconnaissance globale procède par comparaison entre la forme acoustique à reconnaître et un ensemble de formes préalablement acquises, dites de références, indépendamment du contenu linguistique de cette forme.

Travailler sur des références acoustiques assure donc une indépendance de cette technique vis à vis de la langue utilisée.

La programmation dynamique fournit alors un algorithme de comparaison. La fig (1-3) illustre le fonctionnement d'un tel algorithme. [HAT 87],[SAK 78].

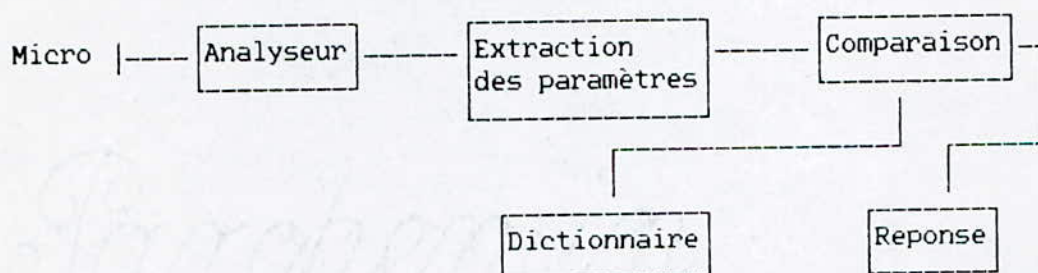


fig (1-2).Schéma général d'un système de reconnaissance de mots.

Cependant, l'usage de cette méthode impose des contraintes au niveau de la prononciation (mots isolés, enchainés...). Le vocabulaire susceptible d'être reconnu est figé lors de l'apprentissage des références et cette méthode ne nécessite pas une segmentation qui est un avantage pour celle ci, mais pour un grand vocabulaire, la phase d'apprentissage est très fastidieuse. De plus, les risques de confusion augmentent avec la taille du vocabulaire.

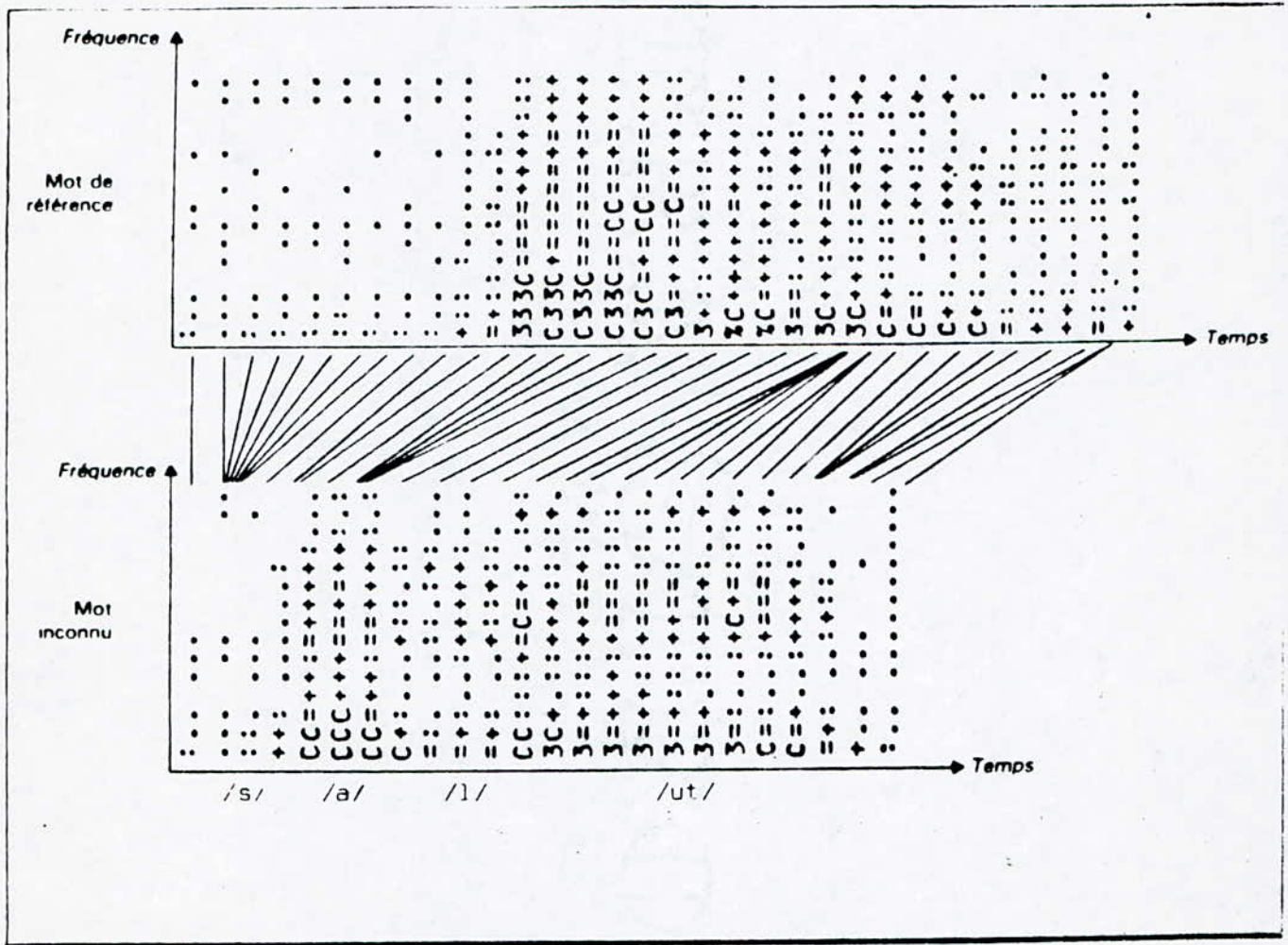


Figure 43— Comparaison par programmation dynamique. La représentation spectrographique des mots utilisée ici est fournie par un Vocoder à 15 canaux couvrant la plage de fréquences 200-5 000 Hz. Chaque colonne correspond à 20 ms de parole. L'intensité sonore dans chaque canal est codée de 0 à 7, soit 8 valeurs représentées graphiquement par 8 caractères depuis blanc (pour 0) jusqu'à "°" (pour 7).

5-2- Reconnaissance analytique

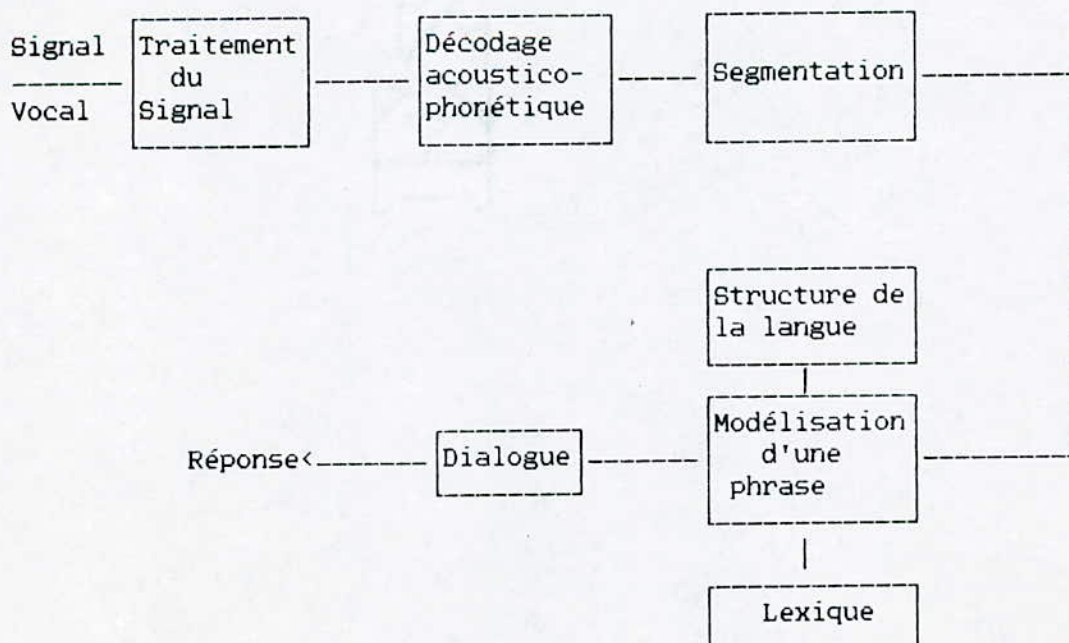
Dans cette approche, ce sont des unités linguistiques que l'on cherche à reconnaître dans le signal de parole.

L'avantage d'une telle méthode est que le vocabulaire à reconnaître n'est pas figé, vis à vis de l'application et présente donc moins de contraintes pour l'utilisateur. Cependant, l'utilisation des connaissances phonologiques et lexicales rend la méthode dépendante de la langue. De plus les problèmes posés par la variabilité du signal de parole [§ 2] limitent encore les performances d'une telle approche et, en pratique, pour faciliter la reconnaissance il est nécessaire d'introduire des contraintes de haut niveau

De façon générale, la R.A.P implique un ensemble de processus qui sont schématisés sur la fig (1-4)

Cette figure résume donc les problèmes à résoudre en R.A.P mais ne représente aucunement l'architecture réelle d'un système de reconnaissance. On constate la grande diversité des problèmes rencontrés:

- Traitement du signal pour la mise en forme du signal de parole et l'extraction des paramètres pertinents.
- Décodage acoustico-phonétique: (c_f III)
- Segmentation: (c_f III)
- Modélisation d'une phrase, en liaison avec les structures syntaxiques de la langue d'une part, et le lexique, d'autre part.
- Recherche d'une interprétation d'une phrase parmi toutes les solutions possibles et le dialogue homme-machine.



fig(1-4). Schéma de principe d'un système de reconnaissance.

6- Intérêt de l'intelligence artificielle pour la parole

Tous les systèmes de reconnaissance fondés sur une connaissance spécifique de la parole, sont confrontés aux problèmes de représentation et d'utilisation de cette connaissance. Les techniques d'I.A fournissent dans ce cadre un certain nombre de solutions.

L'espace de solutions, dans le cas de la R.A.P est l'interprétation partielle ou totale de la phrase étudiée, c'est à dire les mises en correspondance de portions de signal vocal et de portions de message: mots, syntagmes, groupes de mots, phrases.

On risque à tout moment une explosion combinatoire du nombre de solutions, d'où la nécessité de stratégies très élaborées pour contrôler la recherche et tenir compte du contexte. Devant les problèmes de cette taille et de cette complexité, il devient indispensable de chercher d'abord des solutions partielles dont la combinaison et l'extention permettent ensuite de déduire la solution globale.

Les structures de contrôle des systèmes mis en oeuvre doivent alors:

- Acquérir et analyser les données physiques.
- Fixer des sous-butts à atteindre, effectuer les inductions et les déductions plausibles, etc.

En effet, on ne dispose encore que de peu de connaissances dans le domaine de la parole. Il est donc nécessaire de créer des systèmes permettant de formaliser et de tester les connaissances actuellement disponibles afin de pouvoir les faire évoluer pour mieux dominer les problèmes de la parole.

A l'heure actuelle, ces systèmes sont plus des outils de recherche que des systèmes de reconnaissance. Il ne faut toute fois pas attendre de miracle de l'I.A, car un système fondé sur ces techniques n'a de valeur que dans la mesure où les connaissances qu'il manipule en ont.[REC 87]

VI- Conclusion

Nous avons situé nos objectifs et le cadre de notre contribution à la résolution d'un problème de la reconnaissance.

D'où nous avons opté pour un traitement dit de bas niveau "phonétique" qui est d'une très grande importance pour la reconnaissance et la compréhension de la parole.

Une étape aussi importante qui est l'analyse et la paramétrisation du signal vocal. Ainsi pour l'analyse nous avons retenu l'analyse spectrographique et notre travail, c'est surtout basé sur la lecture de spectrogrammes, d'où un décodage acoustico-phonétique.

II Lecture des spectrogrammes

- Définition
- Lecture et intérêt
- Spectrogrammes analogiques et numériques

1- Introduction

Malgré l'importance et l'adaptation des autres méthodes d'analyse du signal pour la parole, cette technique a été retenue pour nous servir comme support de travail et de base pour l'établissement de nos règles de production que nous n'avions pas pu obtenir par une autre méthode.

2- Définition

Un spectrogramme est défini comme une visualisation de la décomposition spectrale d'un signal sonore. Il représente en effet l'évaluation dans le temps dans un plan, ayant le temps en abscisses et les fréquences en ordonnées. Des niveaux de gris sont associés à des intervalles de valeurs d'énergies dans les différentes bandes de fréquence. Plus le niveau d'énergie est élevé plus l'image est noire.

3- Utilisation et intérêt de la lecture de spectrogrammes

3-1- Intérêt:

Le spectrogramme permet d'établir les relations existantes entre des événements acoustiques, les distinctions phonétiques, ainsi que les modes articulatoires.

On dispose ainsi d'un support visuel permettant d'établir les relations entre la production et la réalité acoustique de la parole. La reconnaissance de ces relations est indispensable à un travail de R.A.P. La manière dont fonctionne l'oreille pour décoder la parole nous est mal connue. En effet chaque être entendant se comporte comme un expert en décodage acoustico-phonétique. Mais cette expertise n'est pas formalisable.

La difficulté de travailler directement au niveau de la perception auditive justifie l'intérêt que l'on porte à la perception visuelle de la parole à partir d'un spectrogramme. [LON 86].

3-2- Lecture de spectrogrammes

La lecture de spectrogramme consiste à chercher la suite de phonèmes prononcés à partir de l'observation de l'image représentant la prononciation d'une phrase à priori inconnue.

Du fait qu'un être humain soit capable de lire un spectrogramme prouve qu'il existe dans la parole des informations invariants permettant de réaliser un décodage acoustico-phonétique.

La lecture de spectrogrammes nécessite des informations phonologiques, propres à l'application puisque les connaissances phonétiques seules ne sont pas suffisantes.

La lecture de spectrogrammes est fondée sur l'utilisation de tout un ensemble d'indices, savoir les utiliser et les contrôler peut donc être d'un intérêt capital pour la reconnaissance de la parole.

4- Spectrogrammes analogiques et numériques

- Spectrogrammes analogiques:

Les spectrogrammes analogiques présentent plusieurs inconvénients tels que:

- . Le manque de fiabilité dû au système mécanique.
- . La difficulté de visualiser directement des données stockées sous forme numérique.
- . La difficulté de traiter de façon numérique l'image correspondant au spectrogramme obtenu de façon analogique.

Ce dernier point constitue l'argument principal pour utiliser des spectrogrammes numériques. [LIE 77].

- Spectrogrammes numériques:

Pour effectuer des opérations d'analyse et de représentation du signal acoustique, il est tentant d'utiliser l'ordinateur, on peut attendre plus de rapidité, plus d'objectivité, et plus de souplesse.

Cependant, il faut vaincre plusieurs types de difficultés d'ordre théorique et pratique, concernant la transformation du signal en une suite de nombres, l'application de la transformée de Fourier d'un signal donné sous forme discrète, le temps de calcul et la capacité mémoire requis pour mener à bien ces traitements.

Il faut cependant remarquer que l'apparition des spectrogrammes numériques peut résoudre plusieurs difficultés. Dans la mesure, où à l'heure actuelle les personnes étudiant le signal utilisent des supports numériques.

L'utilisation des spectrogrammes numériques résout un problème qui apparaît dans un spectrogramme analogique qui est le traitement d'images des spectes. [LIE 77].

5- Conclusion

Lors de l'interprétation d'un spectrogramme on peut s'intéresser à l'évolution de la fréquence fondamentale (niveau prosodique) ce qui permet grandement d'améliorer la lecture. Mais les niveaux syntaxiques et sémantiques sont peu utilisés puisque les performances étant identiques pour des phrases dénuées de sens ou syntaxiquement incorrectes.

Par conséquent, on peut obtenir un taux élevé de reconnaissance acoustique sans l'intervention de niveaux supérieurs.

Nous nous sommes limités à une reconnaissance acoustico-phonétique en espérant que cette reconnaissance soit généralisable au niveau d'une machine.

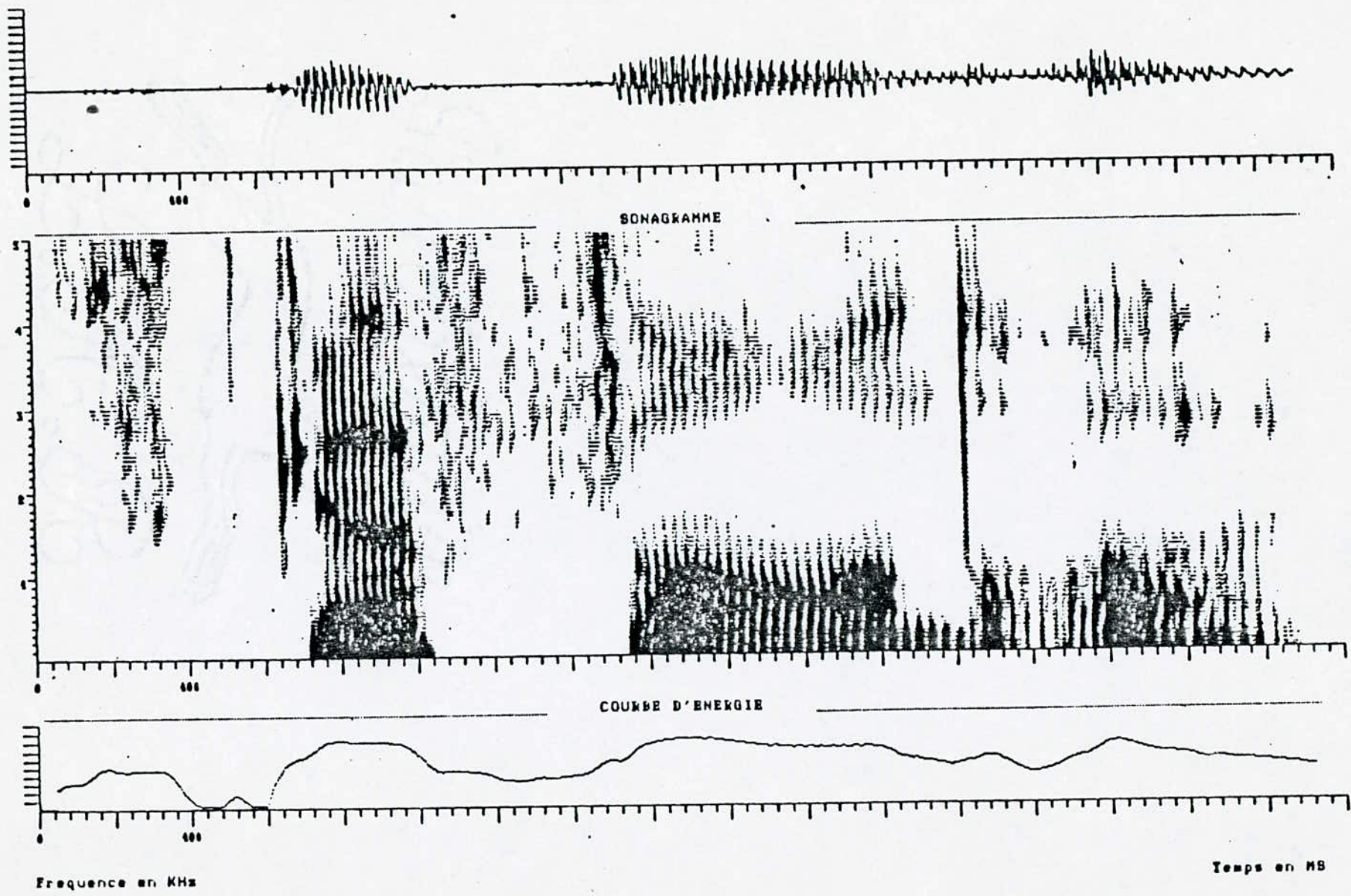


Figure 2.1 : Un exemple de spectrogramme numérique

III Décodage acoustico-phonétique

- Stratégie pour la segmentation
- Evénements vocaliques évidents
- Evénements consonantiques évidents

1- Introduction

Le décodage acoustico-phonétique est l'étape du traitement de la parole qui cherche à transformer le signal acoustique en une suite d'unités pseudo-phonétiques discrètes, c'est à dire une suite de symboles équivalents ou non aux phonèmes.

Le rôle primordial de ce module est de segmenter le continuum sonore en utilisant des indices de séparation présents dans le signal.

La difficulté de la segmentation réside déjà dans le choix de l'unité de décision (ou forme de base) et celle-ci conditionne fortement la stratégie de reconnaissance associée au système. Ces segments peuvent être des sons élémentaires constitutifs des phonèmes, des pseudo-phonèmes, des diphonèmes ou des syllabes.

1-1- Les phones.

Ce sont des unités infra-phonémiques de nature purement acoustiques. Ces événements sont ensuite assemblés en unités de plus en plus large dans différents niveaux linguistiques du système de reconnaissance.

1-2- Les phonèmes.

Unité élémentaire des sons du langage (schématiquement voyelles ou consonnes). Il est reconnu que tout auditeur humain est capable d'identifier dans la chaîne parlée des segments discrets de la taille du phonème. Si celui-ci apparaît comme l'unité idéale, ses frontières ne sont pas toujours clairement définies dans le signal et la segmentation en phonèmes semble une opération de haut niveau chez l'auditeur. [HAT 87-1].

1-3- Diphones et Syllabes

Segment de parole qui s'étend de la zone stable d'un phonème à la zone stable du phonème suivant en son centre la transition entre ces deux sons. La segmentation en diphones et en syllabes à l'avantage de produire des unités longues qui intègrent les phénomènes de coarticulation. On évite ainsi une description explicite des

déformations contextuelles des phonèmes. Mais sa localisation ainsi que son évolution sont complexes à réaliser.

Ces différentes techniques utilisent des algorithmes de segmentation qui consiste généralement à déterminer les portions stables et instables du spectre du signal d'énergie. D'autres méthodes ont été employées:

- Analyse temporelle de la densité de passage par zéro.
- Quantification vectorielle, etc. [BON 86].

Le principe que nous avons retenu pour la segmentation s'appuie sur des règles qui sont directement inspirées de ceux décrites par M.Bulot.

2- Stratégie pour la segmentation

Les unités que nous souhaitons segmenter puis identifier sont les phonèmes ou des portions de phonèmes. On pourrait privilégier le niveau de la syllabe dont les limites sont plus nettement marquées, mais le problème de la description globale de ce type de segments est délicat en raison de la richesse des informations qu'il contient.

2-1- Segmentation en reconnaissance automatique

La phase de segmentation consiste à effectuer une identification partielle et locale des événements acoustico-phonétiques en leur affectant les étiquettes suivantes:

- voc pour les noyaux vocaliques,
- fric pour les segments constrictifs,
- occ pour les segments interrompus,
- cons pour les événements consonantiques qui n'ont pu être classés constrictifs ou interrompus.

Le choix des macro-classes utilisées dans cette première description du signal vocal est uniquement guidé par la nature des formes qui sont employées pour la localisation des événements et qui sont caractéristiques des sons analysés. Une description détaillée de ces macro-classes est donnée dans les paragraphes suivants.

3- Evénements vocaliques évidents.

Les divers événements vocaliques sont désignés par la coïncidence de plusieurs types de collines apparaissant sur des paramètres standard ou provisoires. Il s'agit généralement de schémas très marqués sur des attributs tels que l'énergie totale, l'énergie basse fréquence, les différences d'énergie entre certaines zone de spectre, Ces unités, sont toutes désignées par la même étiquette pseudo-phonétique voc, elle correspond à la partie vocalique des syllabes.

3-1- Macro-classe voc

Les noyaux vocaliques évidents sont caractérisés avant tout par une belle colline d'énergie (énergie totale (Ero), spectrale (Esp), basse fréquence (Ebf)) et doivent se maintenir pendant au moins 60ms. fig(3-1)

4- Evénements consonantiques évidents.

Comme pour les événements vocaliques évidents, ces unités sont décrites par des règles qui n'utilisent pas des phénomènes contextuels déjà identifiés. Les événements consonantiques détectés dans cette phase du décodage se projettent sur trois macro-classes pseudo-phonétiques:

- La macro-class fric est affectée aux consonnes fricatives,
- Les silences et la plupart des consonnes interrompues sont classés occ,
- Les autres événements consonantiques bien marqués qui n'ont pu être rattachés aux deux macro-classes précédentes sont étiquetés avec cons.

4-1- Macro-classe fric

Les événements constrictifs sont repérés au moyen de collines importantes sur la densité de passage par zéro ou des vallées sur des paramètres tels que la différence entre l'énergie dans les basses fréquences et celle des hautes fréquences. (fig 3-2).

Ces formes sont maintenues la plupart du temps plus de 100ms, et coïncident fréquemment avec une vallée de l'énergie basse. Il est possible que dans un contexte non voisé à gauche ou à droite, cette vallée n'apparaisse pas, l'énergie basse doit alors présenter un plat relatif et sa valeur moyenne ne doit guère être éloignée du seuil de bruit.

4-2- Macro-classe occ

Les segment recherchés à ce niveau sont ceux que l'on peut rapprocher d'une fermeture partielle ou totale du conduit vocal. Les événements interrompus sont définis par une importante vallée de l'énergie suivit éventuellement d'une explosion que l'on détecte à partir de collines particulières pour des paramètres tels que l'énergie totale, la densité des passages par zéro, etc. (fig 3-3).

L'interruption est confirmée en vérifiant que l'énergie atteint des valeurs proches du seuil du bruit ou encore en observant une brusque augmentation de l'énergie lorsqu'on se déplace vers la voyelle adjacente.

4-3- Macro-classe cons

Les événements consonantique évidents désignent des phénomènes acoustico-phonétiques qui n'ont pu être assignés à l'une des deux classes précédentes. Ils sont désignés par la coïncidence de vallées pour les divers paramètres qui mesurent l'énergie du spectre dans des zones privilégiées.

5- Conclusion

L'étude des indices qu'on propose pour la segmentation du signal de parole en unités pseudo-phonétiques, ainsi que l'étiquetage des segments nécessitent une chaîne d'acquisition et de traitement, dans le but de réaliser des algorithmes de segmentation.

cependant, ces indices ne seront utilisés que pour délimiter les frontières des phonèmes pour la lecture des spectrogrammes.

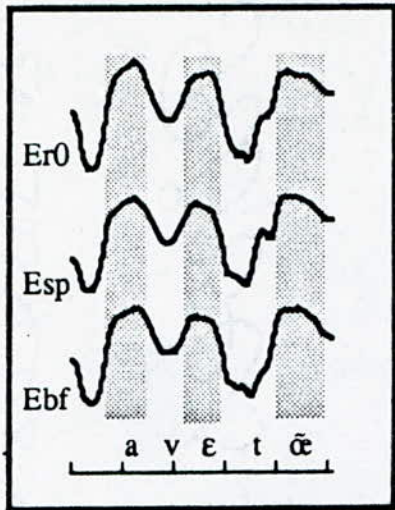


Figure 3.1: Coïncidence de collines d'énergies sur des voyelles

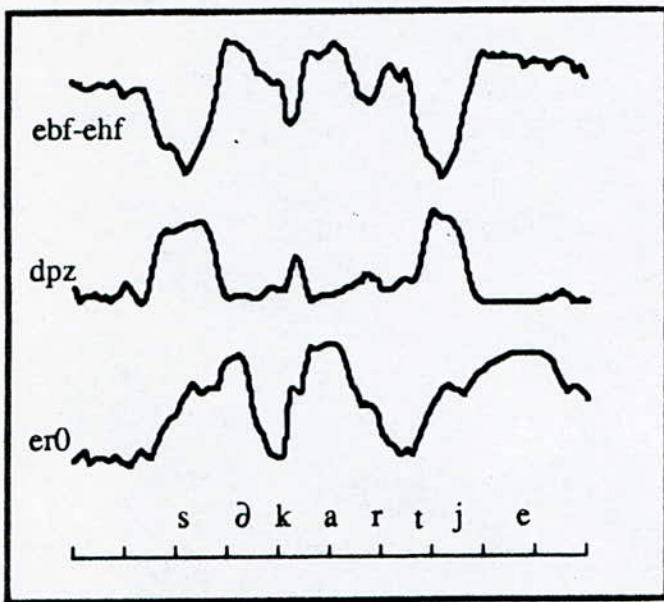


Figure 3.2: coïncidence de collines et de vallées pour des événements consonantiques

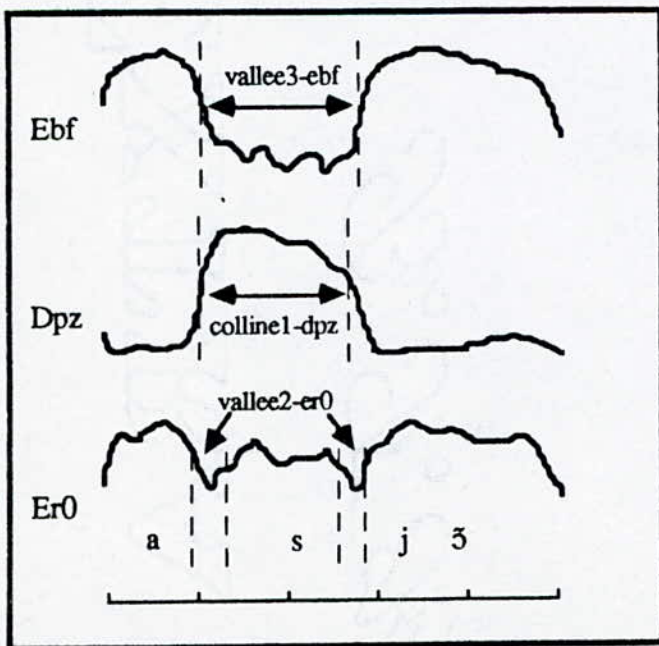


Figure 3.3: Séquences de formes sur une consonne fricative

2 eme PARTIE
Intelligence artificielle

IV Intelligence artificielle et systèmes experts

- **Représentation des connaissances**
- **Systèmes experts**
- **langage**

1- Introduction

Définir la notion d'intelligence est un vaste problème. Nous nous contentons de l'idée, fondée sur l'observation, du comportement intelligent.

C'est ce que proposent d'étudier les nouvelles disciplines telles que: la CYBERNETIQUE et l'INTELLIGENCE ARTIFICIELLE.

L'intelligence artificielle a pour but l'étude des activités réputées intelligentes de l'être humain pour lesquelles aucune méthode standard n'est à priori connue.

Ainsi si l'informatique traditionnelle est la science du traitement de l'information qui manipule surtout des nombres et des caractères (calcul numérique, gestion), l'I.A nouvelle de cette science, s'intéresse à des cas où ce traitement ne peut être ramené à une méthode algorithmique simple 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 .

L'I.A présente un intérêt fondamental, que l'on peut illustrer par ses utilisations potentielles:

- L'amélioration de la communication homme machine.
- Résolution de problèmes nouveaux.
- La construction progressive.

L'orientation actuelle de l'I.A consiste à réaliser des systèmes susceptibles de résoudre des problèmes dans un domaine restreint en se fondant sur les connaissances propres à ce domaine. L'I.A apparait ainsi comme le domaine privilégié de la formalisation et de la manipulation des connaissances.

Il est bien connu que pour résoudre un problème il faut bien le poser. Les problèmes en I.A n'échappent pas à cette règle, et leur résolution nécessite que les données qu'ils ont à traiter soient représentées selon des structures facilitant leur manipulation, dans ce qui suit seront exposés les modes de représentation les plus couramment rencontrés dans les systèmes actuels.

2- Représentation des connaissances

Pour établir un diagnostic ou recherche d'une panne de moteur, médecins ou mécaniciens ne font pas seulement appel à leur capacité de raisonnement mais aussi d'une manière plus au moins consciente à une foule de "connaissances".

Donc pour espérer réaliser des tâches intelligentes, il faut disposer d'une grande quantité de connaissances relevant du domaine traité. Ce qui suppose que l'on sache les représenter dans la mémoire de l'ordinateur. [GALL 85]

2-1 Définition

Représenter les connaissances dans un ordinateur consiste à trouver une correspondance entre le monde extérieur et un système symbolique permettant de raisonner. [BON 84]

IL existe plusieurs représentations de connaissance dont les plus importantes: les règles de production, les réseaux sémantiques etc....

Cependant, avant de les présenter, il faut préciser que ces différents formalismes se déduisent des formules de la logique des propositions (logique d'ordre 0) et celle de la logique des prédicats (logique d'ordre 1).

2-2- Notion de logique

- Logique des propositions

Elle est définie par une syntaxe et des règles d'inférence.

* On définit dans la syntaxe:

- Les connecteurs: ET, OU, NON, IMPLIQUE, EQUIVALENT.
- Les parenthèses: (et).
- Des variables propositionnelles représentant des propositions.

Définition: une proposition est une assertion affectée de l'une des valeurs "VRAI" ou "FAUX".

- Des formules bien formées (FBF) définies récursivement comme suit

Exemple1:

Si on pose A="la mer est calme"
 B="la mer est agitée"
 A ou B est une FBF exprimant "la mer est calme ou agitée"

* Règles d'inférence:

Le calcul des propositions repose sur la règle dite du MODUS PONENS qui veut que:

Si P implique Q et P est vrai alors Q est vrai, qui s'exprime formellement par: $(P \text{ et } (P \text{ implique } Q)) \text{ implique } Q$.

Exemple2:

soit la FBF: l'animal vole alors l'animal est un oiseau.
et soit la proposition vraie: l'animal vole.
par MODUS PONENS: l'animal est un oiseau.

- Le calcul des prédicats

Le calcul des propositions est insuffisant pour exprimer de nombreuses assertions.

Exemple3:

tout ami de X est ami de Y
Z est un ami de Y

Le MODUS PONENS ne suffit pas pour déduire que Z est un ami de Y

Ainsi le calcul des prédicats est une extension du calcul propositionnel, défini aussi par une syntaxe et des règles d'inférence.

* Au niveau de la syntaxe, l'alphabet de base est composée de:

- Connecteurs des calculs des propositions.
- Un ensemble de variables: X,Y,Z.....
- Un ensemble de constantes.
- Un ensemble de prédicats à m arguments.
- Un ensemble de fonctions à n arguments.

définition: Un prédicat est une fonction prenant un ou plusieurs arguments et retournant les valeurs "VRAI" ou "FAUX"

* Introduction des quantificateurs

les quantificateurs universels (: quelque soit) et existentiel (: il existe) permettent de préciser la portée de toute assertion.

Exemple4:

"tout Homme est mortel" est représenté par:
 $(X) (\text{homme}(X)) \text{ implique mortel } (X)$

Remarque: Les quantificateurs sont ajoutés à l'alphabet de base.

* Règles d'inférence:

Elles permettent de créer de nouvelles assertions à partir d'assertions déjà existantes, les deux principales sont le MODUS PONENS et la spécialisation universelle.

Cette dernière consiste à remplacer une variable quantifiée par une constante: $(\forall X) (P(X))$ implique $P(A)$
Ce processus est une substitution.

Exemple5:

Si on prend l'exemple 3 symbolisé par:

- 1- $(\forall X) (\text{ami}(\text{ali}, X), \text{implique } \text{ami}(\text{omar}, X)).$
- 2- $\text{ami}(\text{ali}, \text{ahmed}),$ "ahmed est ami de ali".

par substitution on obtient:

$\text{ami}(\text{ali}, \text{ahmed})$ implique $\text{ami}(\text{omar}, \text{ahmed}).$

par MODUS PONENS on déduit:

$\text{ami}(\text{omar}, \text{ahmed}),$ "ahmed est ami de omar".

2-3- Les règles de production

Les connaissances sont fragmentées sous forme de couples: condition/action, plus précisément, une règle de production est la spécification d'une action conditionnelle:

Si (condition) alors (action)

Les parties condition et action peuvent être respectivement un ensemble de conditions et un ensemble d'actions reliés par des conjonctions ou des disjonctions.

les règles de production sont interprétées de façon chaque fois que les conditions d'une règle sont vérifiées, les actions de la règle sont exécutées.

* Avantages:

- Modularité: La base de connaissances sera facile à étendre et à modifier car les règles sont totalement indépendantes les unes des autres.
- Caractère naturel: cette représentation est très naturelle à l'Homme, ceci facilitera:
 - La collecte des connaissances.
 - La compréhension du raisonnement.

* Inconvénients:

- L'utilisation des règles risque de conduire à des systèmes très lents lorsque le nombre de règles augmente.

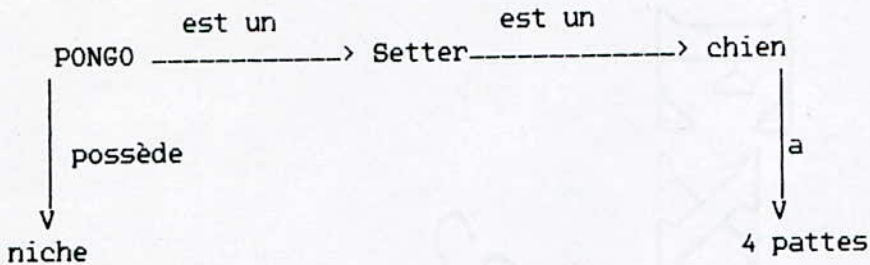
2-4- Réseaux sémantiques

Un réseau sémantique est un ensemble de noeuds reliés par des arcs:

- Les noeuds représentent des concepts du domaine.
- Les arcs, des relations entre concepts.

Exemple 6:

PONGO est un Setter



Il y a héritage des propriétés des concepts entre les plus élevés vers leurs fils, ainsi dans l'exemple précédent (chien vers setter). Il est possible d'avoir des structures plus complexes (plusieurs attributs).

Il existe différentes techniques d'interprétation des réseaux parmi lesquelles:

- Activation progressive du réseau: chaque noeud actif transmet le signal à ses voisins, ceux ci à leur tour transmettent ce signal d'activation à leurs voisins. S'il existe un noeud atteint par les deux signaux, c'est qu'il existe un chemin entre les deux noeuds de départ, la réponse est alors positive.
- Comparaison de fragments de réseau: la question posée s'est formulée sous la forme d'un fragment de réseau, ce fragment est ensuite comparé aux réseaux stockés dans la base de connaissances afin de savoir s'il existe un fragment identique.

* Avantages:

- Les réseaux sémantiques permettent une représentation aisée des connaissances relationnelles.
- Il est possible d'ajouter des règles de production à un réseau sémantique.

* Inconvénients:

- La taille des fragments de connaissances est trop faible.
- Difficulté pour un raisonnement à cause du premier inconvénient.
- Les problèmes de transmission de propriétés entre les noeuds.

2-5- Système de production

La plus part des applications de l'I.A utilisent une structure de systèmes de production. L'intérêt des systèmes de production est de bien séparer l'ensemble des connaissances de l'usage qui peut en être fait, ce qui permet d'affiner progressivement cet ensemble. Un système de production se caractérise par l'utilisation de règles de production.

Un système de productions se décompose en trois parties:

- Interface utilisateur:

C'est un élément indispensable, c'est elle qui va jouer le rôle de liaison entre le système et un utilisateur qui n'est pas censé connaître le mode de représentation des connaissances.

- Base de connaissances:

Les deux éléments bien connus de la base de connaissances sont la base de faits et la base de règles.

* Base de règles:

L'ensemble des règles de production qui représentent la formalisation des connaissances.

* Base de faits:

Elle est appelée aussi mémoire de travail, ou à court terme, qui contient la description du problème et les données déduites par le système.

- Moteur d'inférences:

Composant essentiel d'un système de productions, il représente la structure de contrôle du système, c'est à dire il puise de la base de connaissances (faits et règles) pour démontrer ou aboutir au but demandé, il est caractérisé par la façon dont il procède pour atteindre ce but étant donné les contraintes imposées par les conditions initiales.

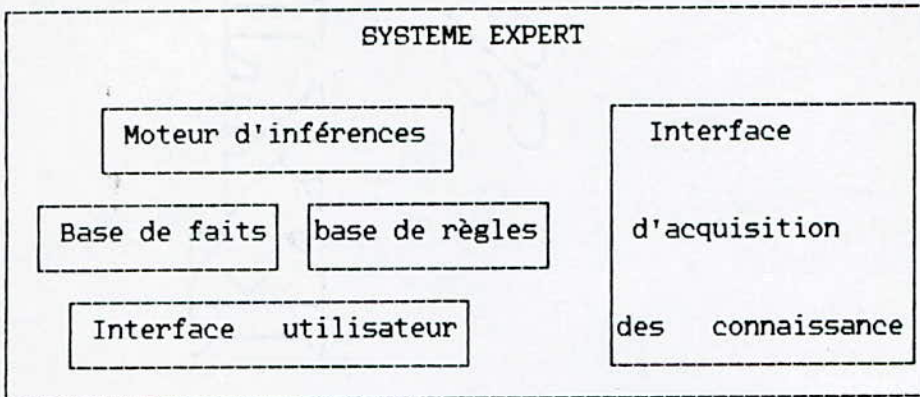
Les systèmes experts, cas particulier des systèmes de production sont choisis selon le domaine d'application.

3- Systèmes experts

3-1- Principe

Le principe de tels systèmes est d'exploiter au mieux une masse de connaissances acquises au près d'experts humains pour traiter avec compétence des problèmes. La séparation entre les connaissances d'une part, et les programmes qui les utilisent, d'autre part, est une caractéristique essentielle des systèmes experts.

A la traditionnelle structure d'un programme en informatique classique (algorithme + structure de données) on substitue ainsi une nouvelle structure composée.



- Interface utilisateur:

- Bases de connaissances:

Base de règles

Elle représente les connaissances spécifiques d'un domaine d'expertise indiquant quelles conséquences tirer, ou quelle action établir.

Base de faits

Elle contient les situations que l'on considère établies ou à établir.

-Moteur d'inférences:

Le moteur d'inférences constitue l'organe principal d'un système expert d'où nécessite une étude plus importante.

3-2- Cycle de fonctionnement

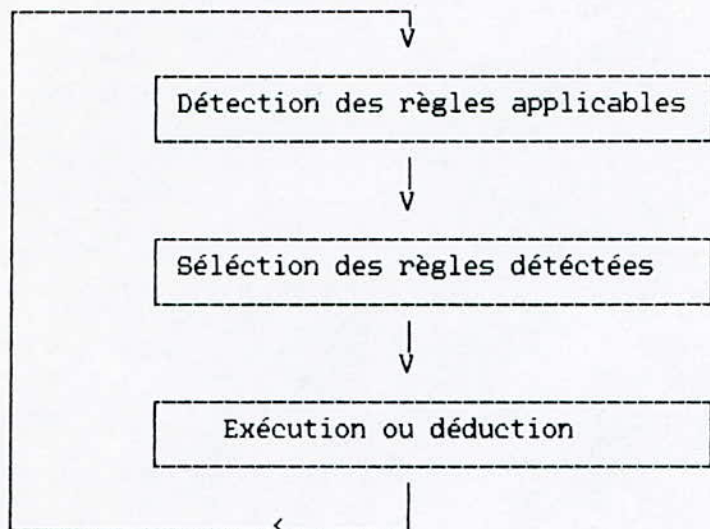
Le cycle du moteur d'inférences comprend trois étapes:

Détection: On effectue un filtrage règles, c'est à dire que l'on vérifie si les conditions ou prémisses sont vérifiées dans la mémoire de travail. L'ensemble des règles validées est ainsi constitué. Si le système utilise des variables, plusieurs instanciations d'une même règle peuvent être validées.

Sélection: Les règles qui doivent être déclenchées sont choisies parmi l'ensemble des règles validées. Différents critères peuvent être utilisés pour cela par exemple, la règle dont les prémisses sont les plus récentes, la règle la plus spécifique, la "mieux" instanciée, etc.. Cette étape est également appelée résolution de conflits.

Déduction: La ou les règles sélectionnées sont déclenchées (exécution d'actions ou modification de la base de faits). Ce cycle s'arrête lorsqu'aucune règle n'a été sélectionnée dans la première étape ou sur une intervention explicite, par exemple, en conclusion de règle.

Tous les systèmes de production suivent ce cycle mais réalisent chaque étape de façon spécifique en fonction du problème traité.



Il est possible que dans l'étape de détection il n'y aura aucune règle candidate: Dans ce cas l'un des choix précédent a dû être erroné et il faudra rétablir la situation précédente, c'est à dire que le moteur remet en cause des déclenchements antérieurs, on dit qu'il opère un retour arrière.

3-3- Modes d'invocation des règles

Le moteur d'inférences met en oeuvre des mécanismes déductifs et inductifs pour résoudre les problèmes décrits dans la base de faits.

Le choix d'un mécanisme par rapport à un autre dépend de l'application. D'où, il n'existe pas d'arguments généraux, à priori en faveur de l'une ou l'autre.

Chainage avant

Un moteur d'inférences fonctionne en chaînage avant lorsque les faits de la base de faits sur les quels portent les déclencheurs de règles, représentent des informations dont la valeur de vérité est déjà établie.

Les règles utilisées sont dites alors en avant, en exploitant de telles règles, le moteur d'inférences suit un raisonnement depuis les données (faits de la base de faits) vers le but (partie conclusion des règles).

La partie condition des règles est la partie déclencheur des règles, chaque fois qu'une règle est déclenchée, sa partie conclusion est rajoutée à la base de faits.

Exemple:

base de règles

- (1)- A,B---->C
- (2)- C,H---->L
- (3)- L,M,N--->O
- (4)- L,I---->P

évolution de la base de faits

A,B,H,I
règle(1)
A,B,H,I,C
règle(2)
A,B,H,I,C,L
règle(4)

base de faits initiale:

A,B,H,I,P(but à atteindre)

A,B---->C: signifie si A et B alors C.

Chainage arrière

Un moteur d'inférences fonctionne en chainage arrière lorsque certains faits de la base de faits sont considérés comme à établir ou à évaluer.

La règle déclenchée apporte donc un fait à établir dans la base de faits.

Le moteur s'arrête sur un succès lorsque la base de faits ne contient plus de faits à établir ou lorsque le fait à démontrer est maintenant établi.

Exemple

Base de règles précédente.

Base de faits initiale précédente.

évolution de la base de faits:

P ,A,B,H,I
règle(4)
L ,A,B,H,I
règle(2)
C ,A,B,H,I
règle(1)
A,B,H,I

Un fait souligné désigne un fait à établir ou à démontrer.

Il n'y a plus de faits à démontrer dans la base de fait donc réussite.

Chainage mixte

Dans les moteurs d'inférences fonctionnant en chainage mixte la base de faits comprend des faits considérés comme établis (données) et des faits à établir (buts).

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

Chaînage bidirectionnel

Un moteur d'inférences fonctionne en chaînage bidirectionnel lorsqu'il invoque tantôt des règles en avant tantôt des règles en arrière.

Avec le chaînage bidirectionnel on fait enchaîner des cycles en chaînage avant, et des cycles en chaînage arrière.

3-4- Modes d'inférence

Lorsque les connaissances sont représentées par des propositions de la logique d'ordre 0 (zéro), le moteur d'inférences exploite les connaissances en utilisant les règles d'inférence du calcul propositionnel. Dans ce cas, le moteur est d'ordre zéro.

Le moteur d'ordre 1 (un) manipule des connaissances pouvant être représentées par des prédicats comme "PROLOG".

Cela signifie que le moteur utilise les règles d'inférence du calcul des prédicats.

3-5- Techniques de recherche

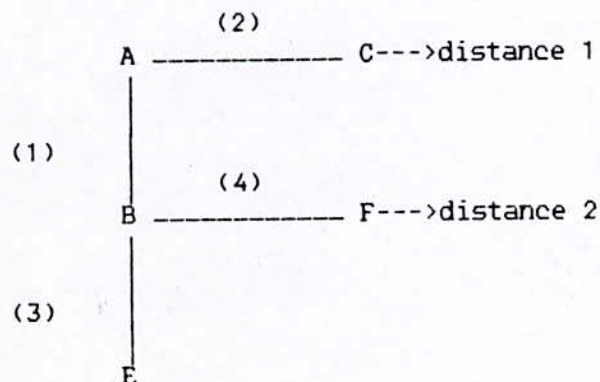
Le processus de résolution du moteur d'inférences se ramène à la construction d'un arbre de recherche dont l'un des éléments terminaux est la solution cherchée (conditions initiales en chaînage arrière, énoncé du problème en chaînage avant).

- La méthode largeur d'abord:

Le moteur d'inférences en largeur d'abord cherche à déclencher toutes les règles possibles d'un état de la base de faits avant d'adjoindre leurs faits conclusion dans la base de faits.

Tous les noeuds à distance "n" de la racine sont développés avant ceux de la distance "n+1".

Exemple:



- La méthode profondeur d'abord:

Dans ce type de moteur d'inférences, lorsque une règle est déclenchée les faits conclusion de la règle sont immédiatement rangés dans base de faits.

Le graphe est développé en profondeur, le noeud à développer est le dernier généré, la recherche se fait dans une seule direction en s'écartant de la racine (noeud de départ).

4- Langage

Le langage le plus utilisé en I.A depuis plus de 15 ans est LISP, cependant la conception de logiciels en LISP réquiert la spécification de toute une série de détails souvent inutiles au regard de l'application.

De plus les langages algorithmiques conventionnels (FORTRAN, BASIC, PASCAL...etc) sont de nature impérative: le programmeur doit indiquer pas à pas les étapes de résolution du problème posé.

Depuis quelques années, le langage "PROLOG" est utilisé pour construire des systèmes experts, en fait PROLOG lui même, est déjà un moteur d'inférences, fonctionnant en chaînage arrière, en profondeur d'abord, ayant un régime par tentatives (retour arrière).

Le système que nous avons réalisé a été écrit en PROLOG. Ce langage nous permet de satisfaire un grand nombre de principes évoqués.

En effet, il peut être vu d'une part comme un langage de spécification dans la mesure où l'on peut spécifier le problème que l'on cherche à résoudre et laisser à l'interpréteur PROLOG la charge de sa résolution. D'autre part, un programme peut donc facilement répondre aux exigences de déclarativité.

Caractéristiques du langage

Le langage PROLOG est dérivé de la logique des prédicats. Le principe de l'interpréteur PROLOG est fondé sur une technique de preuve de théorèmes (la résolution) s'appliquent sur un ensemble de clauses de Horn.

Une clause de Horn est de la forme:

A -----> B1 et B2 etBn. n <=1

où A et Bi sont des littéraux (atomes ou prédicats suivis de ses arguments).

On distingue trois sortes de clauses:

A ----> B1,B2,....Bn. une règle
 A ----> une assertion ou un fait
 ----> B1,B2,....Bn. une suite de buts

Une clause PROLOG peut s'interpréter par: A est vrai si B1 et B2....Bn sont vrais.

Structure d'un programme PROLOG

Un programme PROLOG se décompose:

- D'un ensemble de faits et de règles (clauses de Horn).
- De questions:
 - * Un fait est une affirmation PROLOG contenant un seul prédicat.
 - * Une règle est une affirmation PROLOG établissant une relation entre plusieurs prédicats.
 - * Question suite de termes ou buts.

Donc PROLOG offre un cadre logique au programmeur pour:

- Ecrire les connaissances à l'aide de prédicats.
- Poser le problème à résoudre.
- Laisser le contrôle à l'interpréteur pour la résolution du problème.

Cependant, en utilisant PROLOG, comme moteur d'inférences deux obstacles apparaissent très rapidement

- . Le formalisme des connaissances (expression des règles) en PROLOG (clause de Horn) n'est pas général.
- . La stratégie est figée, elle s'apparente à celle d'un moteur d'inférences fonctionnant en chaînage arrière.

Le fonctionnement du système est ainsi entièrement déterminé par l'ordre des règles et également celui des faits. Si cette méthode peut permettre de formaliser et tester quelque règle, elle n'est pas adoptée à la conception d'un système plus important qui exige une gestion propre à la base de règles, l'inconvénient majeur de l'utilisation directe de PROLOG est la sensibilité à l'ordre des clauses.

Néanmoins, l'utilisation de PROLOG en tant que langage symbolique de programmation des systèmes experts est très avantageuse.

En programmant un nouveau moteur en PROLOG on dispose au départ d'un certain nombre de mécanismes précieux pour réaliser, enchaîner des cycles de base.

- La notion d'espace de travail (data base), assure une grande souplesse pour la manipulation des fichiers.
- Il supprime les boucles, étiquettes, au profit d'une définition du problème qui contient en soit sa propre recherche de solution.
- La notion de liste est la principale structure de données utilisée.
- La possibilité d'utiliser la récursivité, une notion importante pour les applications de l'I.A qui attaquent des problèmes dont on ne connaît pas d'algorithmes.

5- Présentation du système

Le système que nous avons construit a pour objet de représenter et de permettre de formaliser l'expertise en lecture de spectrogrammes. Il faut en particulier formaliser les indices utilisés et la façon dont ils sont combinés pour effectuer l'identification d'un spectrogramme. Le contenu acoustique d'un spectrogramme étant à la fois fortement dépendant du contexte et redondant, il est nécessaire de pouvoir, lors de l'identification, émettre différentes hypothèses et vérifier leur consistance.

Un système de production semble une bonne structure de base pour réaliser ce travail car les connaissances doivent être collectées de façon incrémentale et utilisées indépendamment de la structure de contrôle. Pour construire ce système, nous avons développé un moteur d'inférences.

En outre l'utilisation d'un outil existant déjà nous aurait contraint à respecter toutes les caractéristiques du système.

D'où le choix d'un logiciel est donc difficile à effectuer avant d'avoir une vue très précise des connaissances à traiter.

Nous présentons maintenant quelques principes qui nous ont guidés pour la réalisation du système.

5-1- Principes généraux

- Modularité:

Afin de pouvoir assembler et tester une base de connaissances en lecture de spectrogrammes de façon incrémentale, il est nécessaire de préserver les caractéristiques d'un système de production:

- Séparation entre contrôle et connaissance.
- Représentation déclarative de cette connaissance.

- Modélisation des stratégies:

Pour que la modélisation de la stratégie utilisée par l'expert, lors du décodage d'un spectrogramme, soit aisée, le contrôle au niveau du moteur d'inférences, c'est à dire la gestion du choix et l'ordre de déclenchement des règles, doit pouvoir se faire le plus souplement possible.

- Système général:

Le système étant essentiellement évolutif (les connaissances qu'il contient étant amenées à être modifiées à tous les niveaux), nous avons conservé l'aspect le plus général possible pour le système, en général en ce qui concerne la richesse d'expression des connaissances (on utilise *des propositions*, les connecteurs qui semblent nécessaire).

* La version de PROLOG utilisée est "TURBO-PROLOG" version 1.1

- Aspect interface:

Si le système doit rester général dans son fonctionnement il doit cependant, être pourvu d'un ensemble utilitaires pour faciliter son développement et son utilisation.

- Efficacité:

Certaines précautions ont été néanmoins prises afin d'éviter les problèmes d'explosion combinatoire.

- éviter de tester l'ensemble des règles à chaque cycle.
- supprimer les règles dont l'opération de filtrage a échoué.

- Déclarativité:

C'est une conséquence du caractère évolutif du système. Les parties susceptibles de changer doivent être les plus déclaratives possibles:

- contrôle au niveau du moteur d'inférences.
- le filtrage.
- la représentation interne des faits.
- l'interface.

V Conception du système

- . Représentation des connaissances interprétées
- . Cycle de fonctionnement du moteur
 - . Phase de sélection
 - . Phase résolution de conflits et filtrage
 - . Phase exécution
 - . Phase question
 - . Phase impression
- . Module de dialogue

1- Introduction

Le principe général de "raisonnement" choisi pour le moteur d'inférences est le chaînage avant, principe qui facilite la mise au point et la révision progressive de la base de connaissances, la plus adéquate pour satisfaire le principe ci-dessus et celui des règles de production.

2- Représentation des connaissances interprétées

2-1- Représentation des règles dans la base de règles

Une règle peut prendre différentes formes:

1- Forme simple:

SI condition ALORS conclusion.

2- Les conditions sont reliées par des conjonctions:

SI cond1 et cond2.....et condn ALORS conclusion.

3- Les conditions sont reliées par des disjonctions:

SI cond1 ou cond2.....ou condn ALORS conclusion.

Une règle de cette forme sera écrite sous la forme de plusieurs règles:

SI cond1 ALORS conclusion.

SI cond2 ALORS conclusion.

.

SI condn ALORS conclusion.

4- Les conclusions sont reliées par des "et":

SI cond ALORS conc1 et conc2.....concn.

5- Les conclusions sont reliées par des "ou":

SI cond ALORS conc1 ou conc2.....concn.

Une règle de cette forme sera écrite sous la forme de plusieurs règles:

SI cond ALORS conc1.

SI cond et non conc1 ALORS conc2.

.

.

.

SI cond et non conc1.....et non conc(n-1) ALORS concn.

- Représentation des règles:

Elles sont représentées par un prédicat à trois arguments:

règle(I,L1,L2) déclarée en data base.

où:

I: est une liste à deux éléments tels que:

- Le premier représente le numéro de la règle.
- Le deuxième élément représente le nombre de conditions de la règle.

L1: est la liste de conditions.

L2: est la liste de conclusions.

Exemple: règle: SI la transition du segment est dentale
ET et le segment à droite est une voyelle
ALORS le phonème est un T + 67

- Représentation interne d'une règle:

Règle([1,2],[["la","transition","du segment","est","dentale"],
["vrai"]],[["le","segment","à","droite","est","une voyelle"],
["vrai"]],[["le","phonème","est","une occlusive sonore
+20"],["vrai"]]).

2-2- Représentation d'un fait dans la base de faits:

Un fait est une affirmation entrée par l'utilisateur ou déduite par le moteur d'inférences, il est représenté dans base de faits par le couple:

Fait (assertion, valeur logique).

assertion: est une proposition écrite dans un langage proche du langage naturel.

valeur logique: peut prendre l'une des valeurs "vrai" ou "faux".

Exemple:

Fait: Transition du segment est dentale.

Fait: Il n'y a pas de bas_voisement.

- Représentation interne d'un fait:

Un fait est représenté par un prédicat à quatre arguments:

Fait1(I,H,K,L) déclaré en data base.

où:

I:variable qui nous indique si le fait est exploité ou non.

Si I=1 alors le fait n'est pas encore exploité.

Si I=0 alors le fait a été exploité (on a pu déclencher une règle à partir de ce fait).

Donc lorsqu'on déclenche une règle pour un fait donné la variable I est remise à zéro.

H: variable qui sert à différencier les faits de chaque segment.

K: variable qui sert à différencier les faits initiaux (faits entrés par l'utilisateur), et les faits déduits par le moteur d'inférences.

Si K=1 alors c'est un fait initial.

Si K=0 alors c'est un fait final.

L: est une liste à deux éléments, le premier désigne l'assertion, le second sa valeur logique.

Exemple:

Si l'utilisateur entre le fait "la zone de concentration h-f est comprise entre "4500 et 4600", sa représentation interne sera:

fait1(1,1,1,["la","zone de concentration h-f",
"est","comprise <4500,4600>"],["vrai"])).

Si c'est un fait déduit par le moteur d'inférences sa représentation sera la suivante:

```
fait1(1,1,2,["la","zone de concentration h-f",
"est","comprise <4500,4600>"],["vrai"])).
```

Dans les deux cas lorsque le fait est exploité on peut déclencher une règle à partir de ce fait. Sa représentation sera:

```
fait1(0,1,1,["la","zone","de","concentration","h-f",
"est","comprise","<4500,4600>"],["vrai"])).
```

```
fait1(0,1,2,["l","aspect","du","bruit","est","marque
"],["vrai"])).
```

Si l'utilisateur entre le fait l'aspect du bruit n'est pas marqué, il sera représenté de la manière suivante:

```
fait1(0,1,1,["l","aspect","du","bruit","est","marqué"]
,["faux"])).
```

3- Cycle de fonctionnement du moteur

Le moteur d'inférences réalisé est un moteur d'ordre zéro, invoque les règles en chaînage avant et recherche la solution en utilisant la stratégie en profondeur d'abord avec retour arrière.

Le schéma de fonctionnement est donné ci-dessous:

3-1- PHASE de sélection

Cette phase détermine à partir d'un fait de la base de faits courante et d'un état de la base de règles, un ensemble de règles susceptibles d'être appliquées. Donc la sélection permet de:

- Construire une liste contenant les numéros et le nombre de conditions des règles applicables pour un fait donné (les règles qui contiennent le fait dans leurs parties conditions) appelée "les règles".
- Supprimer les règles contenant la négation du fait considéré (car elles ne sont plus utilisées).

Après avoir récupéré la liste "les règles", deux cas peuvent se présenter:

- . Si la liste est vide alors deux éventualités se posent:
 - Si le fait considéré est un fait initial, alors il n'y a pas de règles applicables pour ce fait, supprimer le fait de la base de faits.
 - Si c'est une conclusion déduite, alors elle est insérée dans la base résultats.

Après exécution du traitement ci-dessus on examine s'ils existent des faits dans la base de faits qui n'ont pas encore été exploités:

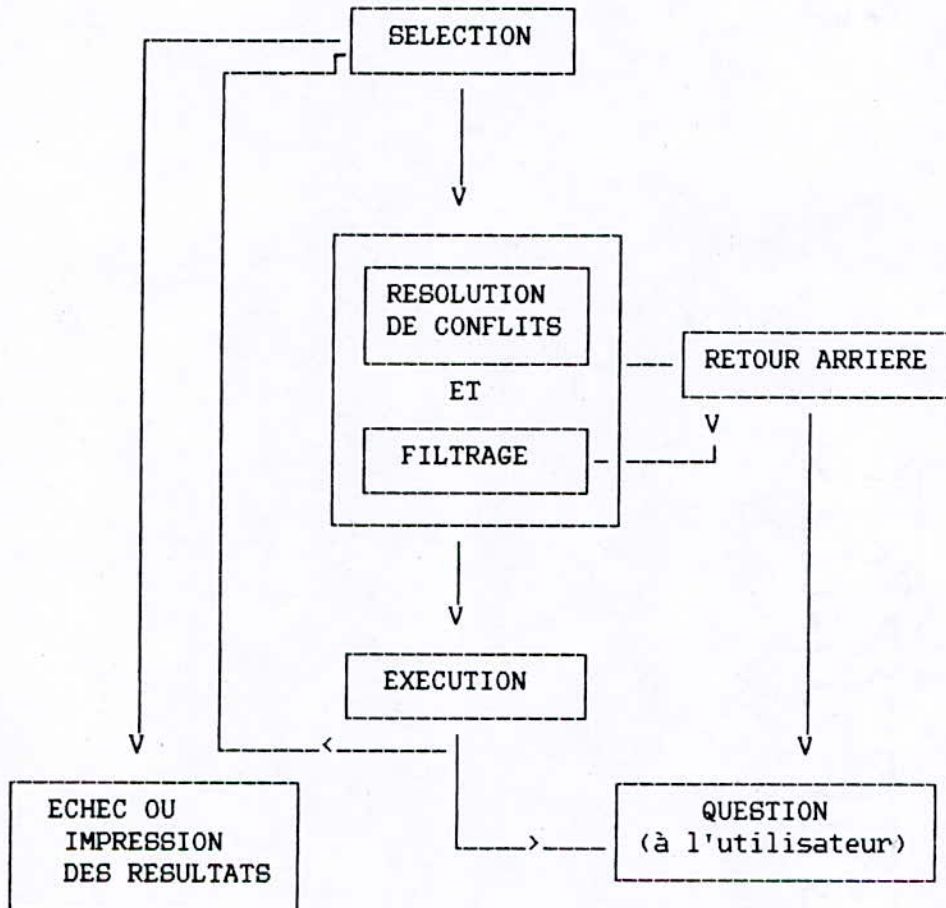
Alors

- prendre un fait.
- exécuter la procédure sélection pour ce fait.

Sinon

- exécuter la procédure impression.

. Si la liste "les règles" n'est pas vide le moteur enchaîne la phase de filtrage et résolution de conflits.



SCHEMA DE FONCTIONNEMENT DU SYSTEME
[VOIR ORGANIGRAMMES].

Exemple:

Base de règles:

- (1)- A et B ----> C
- (2)- A et E et F ----> D
- (3)- K et I ----> L
- (4)- non(A) et D ---->M

Base de faits initiale:

A, B.

L'exécution de la procédure sélection pour le fait "A" :

- 1)- Permet de récupérer la liste "les règles" qui contient les éléments suivant:

les règles=[[1,2],[2,3]].

- Les éléments de la liste "les règles" sont des listes dont:

- Le premier élément de la liste correspond au numéro de la règle.
- Le deuxième élément de la liste représente le nombre de conditions de la règle.

2)- La règle (3) est supprimée car elle contient la négation du fait "A".

On remarque que la liste récupérée n'est pas vide donc le moteur d'inférences enchaîne la phase de filtrage et résolution de conflits.

Le principe de sélection se présente sous forme de simples clauses PROLOG:

```
ordre(T):-fait1(1,2,T),recherche(T,L),confirme(T,L),!.
recherche(T,[Z|L4):-règle(I,X,Y),test(X,T,I,X,Y),asserta
                (règle(I,X,Y)),retract(règle(I,X,Y)).
test([T|_],T,I,X,Y).
confirme([],T):-fait(1,2,T),asserz(fragile(T)).
```

3-2- Phase résolution de conflits et filtrage

Lors de cette étape le moteur d'inférences essaie de déclencher une règle parmi celle qui sont récupérées dans la liste "les règles" (construites à la phase sélection).

La technique utilisée consiste à trier les règles de la liste "les règles" suivant l'ordre décroissant du nombre de conditions (celle qui a le plus de conditions est en tête de liste).

Ce critère permet d'exploiter au maximum la base de faits: C'est l'étape résolution de conflits.

Ensuite le filtrage consiste à prendre la première règle de la liste "les règles-triées", et comparer son déclencheur (les conditions) par rapport à la base de faits courante.

Si les conditions sont vérifiées par les faits

Alors

- le moteur passe à la phase exécution.

Sinon

- le moteur prend la nouvelle règle pointée par tête de liste "les règles triées" et refait le filtrage.

A la fin de cette phase, il est possible qu'aucune règle de "les règles" n'a pu être activée, par conséquent le moteur opère un retour arrière.

Exemple:

Pour l'exemple précédent, la listes "les règles" pour le fait "A" est la suivante:

Les règles =[[1,2],[2,3]].

- Trier "les règles" suivant l'ordre décroissant du nombre de conditions, on obtient la liste "les règles-triées".
Les règles-triées=[[2,3],[1,2].
- Prendre la tête de liste (la règle(2)), les conditions de cette règle ne sont pas vérifiées dans la base de faits (car les faits "E" et "F" n'existent pas dans la base de faits), alors prendre la règle suivante (règle(1)), on remarque que les conditions de cette règle sont vérifiées ("A" et "B" existent dans la base de fait), donc le moteur entame la phase exécution.

3-3- Phase d'exécution (organigramme 4)

Après avoir déterminé la règle à déclencher lors de la phase précédente, le moteur active la règle.

Lorsqu'une règle est déclenchée:

- 1)- La partie conclusion est insérée dans la base de fait, elle est considérée comme un nouveau fait.
- 2)- Déclenchement de la phase sélection pour ce nouveau fait.

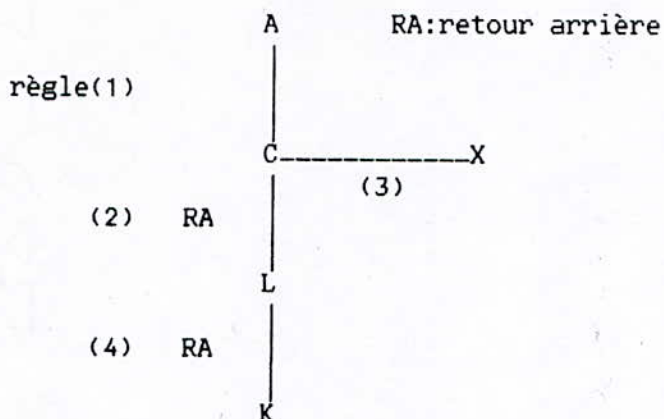
La succession des phases d'exécution génère un arbre d'exécution en profondeur.

Le nouveau fait établi est chaîné au fait qui a permis sa génération suite à l'exécution d'une règle, ce chaînage permettra par la suite au moteur d'inférences d'effectuer éventuellement des retours arrière, de faciliter l'explication du raisonnement entrepris.

Exemple:

Après l'exécution de la phase résolution de conflits et filtrage pour l'exemple précédent, la règle (1) est déclenchée (A et B ---->C) la conclusion "C" est insérée dans la base de faits, et la procédure sélection est déclenchée pour le fait C.

Le chaînage consiste à faire le lien entre le fait générateur (le fait qui a déclenché la règle) et le fait conclusion de la règle, dans notre exemple le chaînage est établi entre le fait "A" et le fait "C".



Dans cet exemple, l'exécution a échoué sur le fait K, car la règle (5) n'a pas pu être appliquée.

Etant donné que K est le dernier fait établi, on remonte au fait générateur L pour lequel la liste des règles applicables est vide par conséquent on remonte au fait générateur C, dont la deuxième règle applicable est activée (règle(3)), on aboutit alors à la conclusion X.

Dans certain cas l'exécution des retours arriere aboutit jusqu'à la racine de l'arbre, cela signifie que moteur a généré toutes les feuilles possibles de l'arbre et n'a pas trouvé de solution.

Par conséquent s'il existe des faits dans la base de faits qui n'ont pas encore été exploités:

ALORS

- Prendre un fait.
- exécuter la procédure sélection pour ce fait.

SINON

- exécuter la procédure question.

3-4- Phase question

Le moteur d'inférences est doté de la stratégie de questions suivante:

- 1)- Ne poser la première question à l'utilisateur qu'après avoir vérifié qu'aucune solution ou que le fait initialement cherché ne peut être déduit d'aucune manière.
- 2)- Les questions sont posées sur les conditions des règles qui n'ont pas pu être déclenchées au niveau du noeud le plus profond de l'arbre d'exécution.
- 3)- Parmi les règles précédentes, celles qui a le moins de conditions à vérifiées.

Le premier et troisième critère évitent de poser beaucoup de questions à l'utilisateur, tandis que le deuxième permet de continuer par le chemin le plus avancé dans la recherche de la solution.

Ainsi pour suivre la stratégie citée le moteur:

- 1) Prend tous les faits insérés dans la "BASE DEMANDEES"
(juste avant le retour arrière).

Chacun de ces faits correspond à un noeud échec de l'arbre donc caractérisé par une profondeur (nombre de règles déclenchées successivement).

- 2) Récupère la profondeur de chaque noeud à l'aide du chaînage établi lors de la phase exécution.
 - Construit une liste de ces faits avec leurs profondeurs notée "liste-profondeur".
 - Trie la liste suivant l'ordre décroissant de profondeur.
- 3) Prend le premier élément de la liste.
 - Récupère la liste "les règles" pour ce fait.
 - Trie la liste suivant l'ordre croissant du nombre de conditions des règles et obtient la liste "les règles-triées".
- 4) Exécute la procédure "question-utilisateur" schématisée par l'organigramme 6.

Les reponses de l'utilisateur peuvent être de trois types:

- a) Une reponse affirmative concernant un fait, alors le moteur déclenche la règle si toutes les conditions sont vérifiées.
- b) Une reponse négative, alors le moteur supprime toutes les règles concernant le fait.

Si la liste "les règles-triées1" n'est pas vide :

Alors:

- prendre une autre règle de "les règles-triées1"

Sinon:

- prendre l'élément suivant de la liste "liste-profondeur".

- c) Une reponse interrogative (pourquoi), dans ce cas le moteur affiche la règle concernée par la question.

Exemple

Base de règles

- (1)- A et B---->C
- (2)- C et H---->L
- (3)- L et I---->K
- (4)- K et O---->Z
- (5)- C et I---->X
- (6)- X et N---->M

Base de faits initiale

A, B, H, I.

Comme dans l'exemple précédent, l'exécution a échoué sur le fait "K" car la règle (4) n'a pas pu être appliquée.

En faisant des retours arrières, on génère le fait "X" par la règle (5), et l'exécution échoue une seconde fois sur le fait "X" car la règle (6) n'a pas pu être activée.

"K" et "X" sont les derniers faits établis pour lesquels aucune règle applicable n'a pu être activée.

L'exécution de la procédure question permet de:

1) Récupérer la liste: "liste-profondeur".

"Liste-profondeur"=[[X,2],[K,3]].

Les éléments de la liste "liste-profondeur" sont des listes dont:

- Le premier élément de la liste correspond au fait.

- Le deuxième élément de liste représente la profondeur du fait dans l'arbre d'exécution.

2) Trier la liste "liste-profondeur" suivant l'ordre décroissant de profondeur.

"liste-profondeur"=[[K,3],[X,2]].

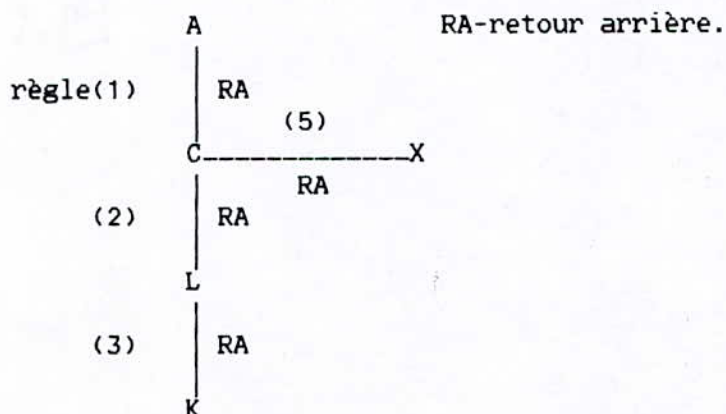
3) Récupérer la liste "les règles" pour le fait le plus profond, dans notre exemple cela correspond au fait "K".

- Les règles=[[4,2]].

- Poser à l'utilisateur la question suivante:
est ce que le fait "0" existe: ?

Si l'utilisateur fournit une réponse affirmative alors la règle: (K et 0---->Z) est activée et la procédure sélection est déclenchée pour le fait "Z".

. Arbre d'exécution



3-5- Conditions d'arrêt du moteur

Le moteur peut s'arrêter dans des phases différentes, c'est à dire suivant le cas :

- 1)- Dans la phase exécution si la solution cherchée est atteinte.
- 2)- Dans la phase sélection si la base de fait est vide.
- 3)- Dans la phase question, si le moteur ne peut déduire aucune conclusion des réponses de l'utilisateur.

Après l'arrêt du moteur la phase impression est déclenchée.

3-6- Phase impression

Cette étape permet l'affichage des solutions éventuelles, en effet si la "BASE RESULTAT" est vide alors le moteur n'a pu rien déduire à partir des faits initiaux dans le cas contraire, il affiche toute les solutions.

4- Module de dialogue

Une part importante dans la conception d'un S.E* concerne les communications avec l'environnement. D'où l'utilité d'un module de dialogue qui a pour but de communiquer avec l'opérateur dans un langage à la fois très précis et proche du langage naturel.

Ce module se compose d'un ensemble d'outils de programmation destinés à faciliter la tâche à l'expert, parmi lesquels une fonction d'explication, qui est capable de fournir la trace d'un raisonnement et expliquer le comportement du système.

L'intérêt d'un tel module est son utilisation par un Profane* et la facilité qu'il lui offre pour faire entrer ses propres faits.

* S.E est l'abréviation de système expert.

* Profane: est utilisé pour désigné une personne ne connaissant pas PROLOG.

VI Vocabulaire de description

- . Description du spectrogramme et base de faits
- . Typologie des règles

1- Introduction

Dans notre système, la base de faits initiale est obtenue par description symbolique de l'image. On cherche ainsi à se mettre dans des conditions expérimentales qui tendent à prouver que la perception visuelle d'un spectrogramme peut conduire à un décodage phonétique. Il faut être certain des données d'entrée du système pour pouvoir effectuer un raisonnement fiable sur ces données.

Cependant, si une description symbolique supprime le risque d'erreurs de détection, dans les données, elle introduit une part de subjectivité dans la description de l'image. Le langage doit donc être bien défini pour éliminer ce problème. Il faut pourtant laisser à l'utilisateur le plus de liberté possible dans la description d'un spectrogramme, ainsi, pour un même spectrogramme plusieurs descriptions sont envisageables et acceptables. Les connaissances, en particulier les règles doivent donc pouvoir rendre compte de ces diverses représentations d'une même situation.

2- Description du spectrogramme et base de faits

Lorsqu'un spectrogramme inconnu est présenté à un expert, celui-ci effectue une segmentation de l'image suivant l'axe temporel. Les segments correspondent, à quelques exceptions près, à un découpage phonémique. L'expert propose ensuite pour chaque segment un ou plusieurs phonèmes.

Durant la tâche de décodage du spectrogramme, l'expert [LON 86] ne relève dans l'image que les motifs qui lui semblent pertinents. Cependant, les éléments de description qu'il fournit comportent déjà une part d'interprétation. Par exemple, le terme qu'il utilise pour décrire un formant (formant nasal ou formant léger) d'une consonne nasale est différent de celui qu'il emploie pour le formant d'une voyelle. Si l'on tenait compte de cette part d'interprétation, seul l'expert serait vraiment capable de décrire le spectrogramme au système.

Nous avons donc répertorié tous les éléments de description utilisés par l'expert, ce qui nous a permis de dégager un ensemble de primitives de description. L'image peut ainsi être décrite d'une manière moins subjective par un non expert.

La description de l'image comporte un "aspect global" et un "aspect segmental". Examinons les primitives de description dégagées en fonction de ces deux aspects*. [HAT 87_1], [HAT 87_2].

2-1- Aspect global

L'expert effectue une observation globale avant de décoder le spectrogramme.

. Longueur

La longueur moyenne d'un segment est évaluée en fonction de la durée de tous les segments dans l'image. Pour chaque segment, il juge alors sa durée par rapport à cette durée moyenne: *sup_moy*, *moy*, *inf_moy*, *très_petite*.

. Aspect

L'intensité de chaque segment est également jugée par rapport à l'ensemble des autres segments.

. Position

On précise également à ce niveau la position du segment dans le groupe rythmique c'est à dire *debut* ou *fin*. Par défaut, un segment est considéré comme étant à l'intérieur du groupe.

. *Bien_découpe*

On précise si le segment se découpe nettement de ses voisins de gauche et de droite.

2-2- Description segmentale

On distingue trois textures dans un spectrogramme: régulière, irrégulière et les zones blanches. Les textures du spectrogramme correspondent à des indices de description, formants, bruit, silence (voir figure 6-1). Chaque indice est caractérisé par un ensemble de paramètres.

L'intensité des indices est en outre décrite de manière relative à l'intérieur d'un même segment (exemple: le troisième formant est plus marqué que le second).

*les termes en caractères gras sont utilisés pour la description des spectrogrammes.

- Texture régulière

Formant

Pour chaque formant (formant-1,...,formant-4), sont indiqués

. Zone

Sa position en fréquence est donnée par sa fréquence centrale qui est déterminée au niveau de la partie stable du formant.

. Aspect

Son intensité: peu_marqué, marqué, tres_marqué.

. Partie_gauche, partie_droite

La direction et l'étendue de ses transitions à gauche et à droite: montant, descendant, descendant_rapide*

. Evolution

Pour les formants les plus élevés seulement (trois et quatre), l'évolution du formant est caractérisée sur tout le segment (vallée, colline).

Basse_fréquence

De plus, pour décrire le bas du spectre, on distingue trois types de formes dans les basses fréquences (entre 0 et 800 HZ):

. Bas_voisement

Energie régulière qui ne s'étend pas forcément sur toute la durée du segment et dont la zone fréquentielle ne dépasse pas 500 HZ.

. Haut_voisement

Energie régulière qui s'étend sur toute la durée du segment et couvre au moins la zone [0,500].

. Irrégularité_très-basse_fréquence (t-b-f)

Energie dans le bas du spectre qui ne présente aucun caractère régulier.

*Par convention les différents mouvements sont toujours décrits du point de vue du segment considéré afin de rendre symétriques les mouvements à gauche et à droite.

. Sabot_gauche, sabot_droit

Prolongement de voisement du segment précédent (ou suivant) dans le segment.

- Texture irrégulière

Bruit

Le bruit est d'abord décrit de manière globale: par son intensité: aspect, son étendue fréquentielle: zone on le décrit également par:

. Concentration_hf, concentration_bf

Les concentrations particulières d'énergie dans le segment bruité sont décrites par leur fréquence centrale (Zone) et leur intensité relative (aspect).

Remarque: Il n'est pas question ici de formant (comme c'est souvent le cas car ces formes ne présentent pas de texture régulière. Lorsqu'une texture est régulière dans un segment contenant du bruit, on parlera de formants.

. Limite basse

- Zone

Indication de la position fréquentielle de la limite basse du bruit.

- Mouvement

L'évolution, le mouvement de cette limite basse sont précisés. Tous les cas de figure sont possibles et peuvent être, décrits au système sous une forme imagée:

- *Vallée
- *Colline
- *Montant_droit
- *Montant_gauche
- *Descendant_droit
- *Descendant_gauche
- *Horizontale

- Aspect

Comme pour les formants, il s'agit de l'aspect relatif dans le segment.

Remarque: La limite supérieure n'est pas utilisée car, en général, le bruit se termine au dessus de 5KHZ, de plus cette information n'est pas pertinente dans la plus part des cas.

- Explosion

Les explosions sont décrites d'une manière analogue au bruit.

- . aspect
- . zone
- . durée

La durée de l'explosion est qualifiée de courte, moyenne, ou longue.

- Zone blanche

Silence

Le terme silence permet de décrire les parties sans énergie correspondant par exemple aux occlusives sourdes. Le silence inclut la zone de basse fréquence du spectre. Les éléments de description concernant les basses fréquences sont donc incompatibles avec silence.

3- Typologie des règles

3-1- Importance du contexte

Plusieurs règles sont nécessaires pour caractériser un phonème en fonction de sa réalisation acoustique, ceci en raison de la variabilité du signal acoustique. Il est souvent nécessaire de connaître les conditions d'application de ces règles, qui sont souvent des informations sur les segments voisins.

Ainsi, la prise en compte d'éléments contextuels est une caractéristique essentielle des connaissances utilisées. Elle peut intervenir à deux niveaux: au niveau des indices et au niveau de connaissances phonotactiques. Ces deux niveaux peuvent d'ailleurs apparaître au sein d'une même règle.

- Indices contextuels

L'expert recherche des indices acoustiques dans les segments voisins du segment analysé. Cette observation indispensable pour effectuer l'identification d'un segment. Elle peut porter sur un contexte très large, jusqu'à trois segments plus loin dans certains cas.

- Contraintes phonologiques et connaissances phonotactiques

Dès qu'un segment est identifié d'une manière certaine, on peut parfois réduire l'espace de recherche pour le segment adjacent en tenant compte des contraintes de type phonologique. Ainsi la suite \F\ \S\ \F\ n'est pas possible à l'intérieur d'un groupe rythmique.

Outre les connaissances purement acoustiques, on peut distinguer les connaissances phonotactiques utilisées par l'expert.

Par phonotactique nous entendons les connaissances utilisant aussi bien la position d'un segment à l'intérieur d'un groupe rythmique, ou un contexte plus général sur l'influence de la nature acoustique d'un segment.

Ce type d'information est utilisé de manière secondaire au cours du raisonnement, en particulier lorsqu'apparaît un ensemble d'indices acoustiques non typique car ne correspondant pas directement à l'information phonétique.

La prise en compte du contexte nécessite l'introduction de variables dans les règles, notamment pour représenter le numéro du segment. Des fonctions spéciales permettent d'instancier les contextes voisins (adjacent, à_droite, à_gauche).

3-2- Les règles d'identification

La majorité des connaissances utilisées par l'expert lors de l'identification d'un spectrogramme ont été formalisées sous forme de règles de production.

- Forme prototypes

Les règles du système ont pour la plupart une forme de type:

```
SI (ensembles de primitives de description )
ET (contexte particulier)
ALORS (ensemble de phonèmes ou classes phonétiques).
```

Elles sont pondérées par des coefficients numériques de vraisemblance sur les conclusions (chap...), et s'appuient sur des connaissances phonétiques, mais également contextuelles (phonologie, coarticulation).

- Connaissances générales et éléments de support à l'identification

Ces règles sont le plus souvent non-contextuelles, elles peuvent porter sur des connaissances générales.

Exemple:

```
SI formant-1 formant-2
ET formant-3 formant-4
ALORS voyelle +85
```

Si le segment comprend quatre formants il s'agit d'une voyelle.

Elles peuvent conclure sur des supports à l'identification, c'est à dire des éléments qui seuls, ne précisent pas vraiment le contenu phonétique du segment (ce ne sont ni une classe ni un phonème) mais permettent le déclenchement ultérieur des règles.

Exemple:

```
SI zone formant-1 < 450HZ
ALORS bas_f1 +70
```

Si la fréquence centrale du premier formant est inférieure à 450HZ, le segment est considéré comme bas_f1.

- Règles contextuelles

Dans de nombreux cas, la présence ou la position d'un indice acoustique est dépendante de l'identification d'un segment adjacent. Généralement on se sert, pour l'identification d'une consonne, de la connaissance de la voyelle.

Exemple:

```
SI bruit
ET 4500 <zone(concentration_hf(bruit))< 4600
ET adjacent s'
ET s' est un \I\
ALORS le phonème est un \Z\ + 20, un \S\ +20
```

S'il existe une zone de concentration du bruit comprise entre 4500 et 4600HZ et que le phonème \I\ est une solution très vraisemblable d'un segment adjacent alors il s'agit peut être des phonèmes \S\ ou \Z\.

Mais le contexte de la consonne peut également servir à identifier la voyelle.

Exemple:

```
SI bruit
ET non(position(finale))
ET limite_basse(bruit,descendant_droit)
ET a_droite (s+1)
ET voyelle (s+1)
ALORS voyelle arrière (s+1) + 60, le phonème(s+1) est un \W\ +20
```

Si le segment contient du bruit et que sa position n'est pas finale et que la limite basse du bruit est descendante dans la partie droite de ce segment, le segment à droite du segment considéré est probablement une voyelle arrière et peut être un \W\.

- Contexte large

Un contexte relativement large (jusqu'à trois segments) est nécessaire dans certains cas.

Exemple:

```
Si voyelle écartée
ET non (position(finale))
ET a_droite (s+1), transition (s+1) dentale
ET partie_gauche (formant-2, horizontale)
ET partie_droite (formant-2, horizontale)
ET partie_gauche (formant-3, horizontale)
ET partie_droite (formant-3, horizontale)
ET partie_gauche (formant-4, horizontale)
ET partie_droite (formant-4, horizontale)
ET a_gauche (s-1)
ALORS transition(s-1) dentale +80
```

Si le segment considéré est une voyelle écartée et que sa position n'est pas finale et si la transition du segment suivant est sûrement dentale et que les trois premiers formants ont des mouvements horizontaux à gauche et à droite alors la transition à gauche de ce segment est très probablement de type dentale.

- Règles d'exclusion

Si de nombreuses connaissances génèrent des hypothèses, certaines connaissances permettent de les éliminer.

Exemple:

SI zone formant-2 > 1500 HZ
ALORS le phonème est un \W\ - 80

Si le second formant est supérieur à 1500 HZ, il ne s'agit vraisemblablement pas du phonème \W\

- Règles phonotactiques

Des règles diminuent la possibilité d'avoir un certain phonème dans un contexte donné.

Exemple:

SI phonème \N\
ET position (finale)
ALORS le phonème \N\ - 20

S'il est possible que le segment soit le phonème \N\ et que le segment soit en fin de groupe rythmique alors, ce n'est vraisemblablement pas un \N\.

Ces informations peuvent être utilisées de manière suggestive.

Exemple:

SI vocalique
ET a_gauche (s-1)
ET voyelle (s-1)
ET a_droite (s+1)
ET voyelle (s+1)
ALORS liquide +40, nasale +40

Si le segment est vocalique et que les deux segments adjacents sont des voyelles alors le segment est probablement une liquide ou une nasale.

Les connaissances phonotactiques sont utilisées pour réduire le plus souvent l'espace des solutions.

Exemple:

Si bruit
ET aspect(bruit, très_marqué)
ET a_gauche (s-1)
ET voyelle(s-1)

ET a_droite(s+1)
 ET voyelle (s+1)
 ALORS le phonème \R\ -40

Si le segment contient du bruit d'aspect très marqué et que les segments adjacents sont des voyelles, alors le segment n'est probablement pas un \R\.

4- Conclusion

La description du spectrogramme au système permet de constituer la base de faits initiale et également comment représenter les différentes connaissances. Ces connaissances doivent être formalisées afin d'être exploitables. En effet, le vocabulaire décrit est doté d'un caractère progressif, ce qui justifie le choix de l'utilisation d'un système qui permet la mise au point incrémentale des connaissances.

Conscients du fait qu'il est impossible de traiter tous les problèmes de la lecture de spectrogrammes en temps limité, nous avons décidé de choisir un domaine dans lequel nous avons le plus d'informations fiables. Les options multilocuteurs, changement de conditions acoustiques d'enregistrements, etc., ont été rejetées.

Nous avons donc axé notre étude sur un corpus de phonèmes restreint:

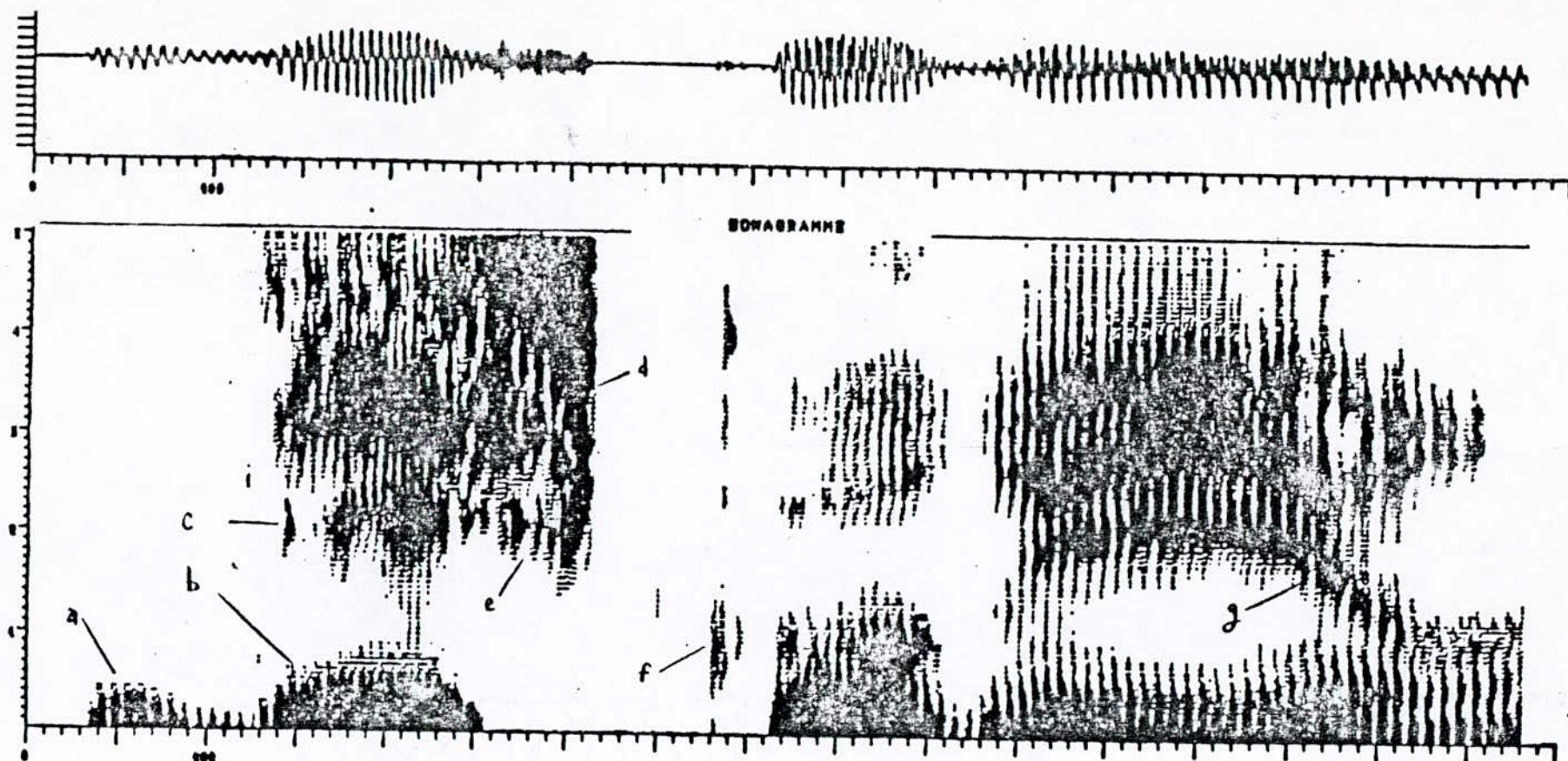
12 phonèmes représentant des oppositions de classes phonétiques:

Sourde/sonore, dentale/vélaire.....

L'ensemble des phonèmes étudiés comprend:

- Les occlusives sourdes: \T\, \K\.
- Une occlusive sonore: \D\.
- Les fricatives sourdes: \S\, \F\.
- Une consonne nasale: \N\.
- Une consonne liquide: \R\.
- Quatre voyelles orales: \A\, \I\, \W\, \E\.
- Une voyelle nasale: *\.

Figure 6.1 : Vocabulaire de description d'un spectrogramme



- | | |
|----------------------------|-----------------------------|
| a : bas-voisement | d : bruit |
| b : formant-1, zone 300 | e : limite-basse du bruit, |
| c : formant-2, zone 2100 | zone 2200, descendant-droit |
| g : formant-2, | f : explosion |
| partie-droite(descendante) | |

SPECTROGRAMME DU MOT: discourir /DISKWRIR/

VII Fonctionnement du système

- . Cycle de fonctionnement
- . Utilisation des classes phonétiques .
- . Question à l'utilisateur
- . Traitement de l'incertain
- . Module de dialogue .
- . Solution adoptée

1- Introduction

Le moteur d'inférences, organe de "raisonnement" du système expert permet, à partir d'une base de faits et d'un ensemble de règles, de résoudre les problèmes d'identification de spectrogrammes posés au système, nous décrivons dans ce chapitre les principales caractéristiques de son fonctionnement.

L'objet du système n'est pas d'imiter le comportement de l'expert à chaque étape de son "raisonnement" mais plutôt, de faire emmagasiner au système les stratégies de "raisonnement" de celui-ci, indispensables à l'identification d'un spectrogramme, pour les formaliser en méthodes utilisables par le moteur d'inférences.

2- Cycle de fonctionnement

La résolution de ce problème d'identification appelle un mode de fonctionnement à partir des faits initiaux décrivant le problème.

Au cours de la tâche d'identification, différentes hypothèses peuvent ainsi être librement testées, car il ne s'agit pas uniquement de vérifier un résultat présumé dans une base de connaissances mais plutôt d'affiner progressivement le résultat. Ainsi le moteur d'inférences réalisé s'adapte à ce mode de fonctionnement qui est déductif.

En matière de lecture de spectrogrammes, le raisonnement n'aboutira pas forcément à un stade terminal, c'est à dire l'obtention d'un phonème comme solution à un segment. Parfois même, seule la classe phonétique du phonème recherché sera identifiée.

Le cycle du moteur consiste donc à choisir (instancier) une règle, tester la liste de ses conditions (existence d'un fait), vérifier

si les conclusions n'ont pas déjà été démontrées (en particulier on ne déclenche qu'une fois une règle sur un segment), déclencher la règle, c'est à dire ajouter les faits ou déclencher les actions contenues dans la liste des conclusions .

3- Utilisation des classes phonétiques

L'expert raisonne de manière progressive à partir de traits acoustiques ou phonétiques. Comme il peut généralement préciser si le segment est, par exemple occlusif ou fricatif. Parallèlement, il détermine le point d'articulation en grande partie à l'aide du mouvement des formants et peut qualifier le segment comme dental, palatal, ou la labial. Ainsi pour un segment vocalique, les traits compacts et écartés sont utilisés pour qualifier les voyelles.

Le moteur d'inférences utilise ces informations à deux niveaux:

- Dans la partie conditions des règles:

Exemple:

Si on a conclu précédemment sur \T\, on peut utiliser le fait occlusif, plus général, sans que une règle ne l'ait conclu.

D'une façon générale, si un fait n'existe pas dans la base de fait, on vérifie s'il n'existe pas dans l'arborescence décrite par la représentation des classes phonétiques, un fils qui puisse permettre d'inférer son existence.

- Dans la partie conclusion des règles

Chaque fois que l'on déduit une classe, cette classe phonétique sera affectée d'un poids, fonction de la pondération de la règle.

On évite ainsi de représenter des règles qui ne contiennent pas d'information supplémentaires, comme:

Si occlusive_sourde.

Alors le phonème est un \P\ (poids), le phonème est un \T\ (poids), le phonème est un \K\ (poids).

L'une des caractéristiques du système est l'utilisation des coefficients de vraisemblance. Ce point est traité de manière spécifique car il concerne le traitement de l'incertain.

4- Questions à l'utilisateur

Le système "raisonne" à partir des faits de description de spectrogrammes qui lui sont fournis initialement. Toute fois, on ne peut supposer que cette description soit exhaustive, le système permet de compléter la base de faits en cours de raisonnement en posant des questions à l'utilisateur.

Une analyse plus fine peut ainsi être demandée de façon conditionnelle afin d'obtenir une information plus précise que celle obtenue à la suite d'une première analyse.

Notre objet n'est pas de posséder des informations exhaustives sur le spectrogramme, mais de disposer d'un sous ensemble d'informations au "raisonnement" du système. L'interaction avec l'utilisation permet de compléter la base mais le système ne doit poser que des questions pertinentes. Or si en chaînage arrière les questions sont guidées par la recherche du but à résoudre, en chaînage avant, en revanche le domaine d'investigation n'est pas limité et pour éviter une surcharge de questions. Un certain nombre de principes ont été adoptés:

- Aucune question ne peut être demandée sur un fait démontrable (qui n'appartient à une conclusion).

- Une question trop générale n'est pas pertinente.

5- Traitement de l'incertain:

Les connaissances utilisées pour l'identification d'un spectrogramme sont généralement formulées par des conclusions pondérées par des quantificateurs comme peut-être, probablement, certainement... Cette formulation se retrouve fréquemment lorsque l'on travaille à partir d'expertise humaine. Le système devra prendre en compte, au cours du "raisonnement" des inférences incertaines. D'où il s'agit là d'un problème difficile à modéliser.

De nombreuses méthodes ont été proposées pour le traitement de l'incertain parmi elles la méthode des coefficients de vraisemblance et la logique floue.

5-1 La logique floue:

Elle peut être définie comme une extension de calcul de prédicats au quel s'ajoute une notion d'incertitude selon cette méthode, comme avec les coefficients de vraisemblance, l'incertitude est représentée de façon numérique. A chaque proposition est associée une distribution: un ensemble flou. La possibilité d'une proposition est déterminée à l'aide de cette distribution.

Exemple:

Proposition: X est un nombre grand
 Distribution: X [0,10] 0.1
 X [10,100] 0.2
 X > 1000 0.7

L'inconvénient majeur de cette méthode est qu'elle ne permet pas d'effectuer un renforcement d'hypothèses dans la mesure où les seuls opérateurs retenus sont de types "MIN, MAX". En particulier, si deux règles ont conclu sur une même information seul le MAX des possibilités obtenues pour ces deux règles sera conservé. On obtiendra ainsi le même résultat si la règle comportant la possibilité maximale s'applique seule.

Pour cela nous avons retenu la méthode des coefficients de vraisemblance.

5-2- La méthode des coefficients de vraisemblance:

Elle permet notamment de gérer la confiance en un fait, lorsque plusieurs sources de connaissance contribuent à son existence.

Notre système s'appuie sur cette méthode. Nous avons pris comme modèle celui de MYCIN (système expert en diagnostic).

- Rappel sur MYCIN :

A chaque fait est associé un coefficient (de vraisemblance, certitude etc.) prenant sa valeur dans $[-1,1]$. Chaque conclusion dans une règle est pondérée par un coefficient (poids), qui détermine la valeur que prendrait la conclusion si toutes les conditions "Ai" étaient vrais de manière "certaine" (coefficient de vraisemblance égal 1).

Soit "B" le fait déduit à partir des "Ai", ce fait doit être affecté d'un coefficient de vraisemblance des conditions "Ai" de la règle R1. De plus l'existence de plusieurs règles (R1, R2,...et Rn) concluant au même fait "B" doit conduire à un renforcement du coefficient de ce dernier.

La combinaison des coefficients de vraisemblance est le principal problème posé par l'utilisation d'un S.E.

- Calcul proposé par MYCIN:

La solution adoptée affecte à "B", après l'application de la règle R1, un coefficient obtenu par la formule:

$$CV1(B) = CV1 * \min [CV (Ai)] .$$

De même après l'utilisation de la règle R2 on obtient:

$$CV2(B) = CV2 * [CV (Ci)].$$

Soient R1 et R2 concluant sur "B" avec les coefficients CV1 et CV2.

MYCIN calcule les coefficients définitifs de "B" en combinant comme suit:

$$CV(B) = CV1(B) + CV2(B) - [CV1(B) * CV2(B)] \quad \text{si } CV1 > 0 \text{ et } CV2 > 0$$

$$CV(B) = -[-CV1(B) - CV2(B) - CV1(B) * CV2(B)] \quad \text{si } CV1 < 0 \text{ et } CV2 < 0$$

$$CV(B) = \frac{CV1(B) + CV2(B)}{-\min[|CV1(B)| * |CV2(B)|] + 1}$$

D'une façon générale, l'introduction des coefficients de vraisemblance crée une relation d'interdépendance entre les règles. Et ceci va à l'encontre du désir de bien séparer les connaissances.

6- Module de dialogue

Le spectrogramme est décrit au système au moyen d'une interface l'utilisateur. Cette interface n'assure pas la cohérence avec la base de faits initiale.

La description du spectrogramme est ainsi accessible à un utilisateur ne connaissant pas PROLOG, la représentation interne des faits étant générée automatiquement par l'interface.

La description est guidée par le vocabulaire de description, il serait inutile de chercher à décrire le spectrogramme en d'autres termes que ceux employés dans les règles.

Si l'utilisateur entre un fait incohérent avec les termes employés dans les règles cette incohérence est signalée.

6- Solution adoptée

Dans notre système, la cohérence de la base de faits est assurée de la manière suivante:

- Ordre de déclenchement des des règles

Une règle dont le déclenchement nécessite une information sur la nature du segment voisin, n'est naturellement utilisée qu'après l'identification de ce segment. Le déclenchement des règles n'est pas ordonné, par exemple le déclenchement de R2 se fait avant R1.

- Evaluation

L'évaluation de la pondération d'une règle s'exprime plutôt à l'aide d'adverbes. Dans la pratique, le poids exact d'une règle est peu significatif mais il indique l'importance relative d'une règle par rapport à toutes les règles qui traitent du même problème. Si l'on utilise les lois de MYCIN, il est extrêmement difficile de déterminer ces pondérations. Dans la conception de notre système pour limiter le problème posé par le choix des coefficients, nous avons attribué des coefficients par rapport à l'importance de la partie condition des règles et pour un même problème le coefficient est renforcé à fur et à mesure.

VIII. Conclusion et Perspectives

Le système que nous venons de décrire permet de tester différentes bases de connaissances dans le cadre de la lecture de spectrogrammes.

La première base de connaissances a été évaluée, montrant qu'il est possible d'obtenir des résultats comparables à ceux de l'expert sur des tâches bien précises. Cette base, constituée à l'heure actuelle d'une centaine de règles, doit encore être améliorée avant de procéder à son évolution.

L'extension de notre système est possible, en ce qui concerne les bases de connaissances grâce à certaines caractéristiques.

Le langage de représentation des connaissances a été fixé dans le cadre de la lecture de spectrogrammes, ce qui permettra de faire évaluer les différentes bases de connaissances, à l'aide d'interfaces spéciales: éditeur de règles ou interface en langage naturel.

La formalisation des connaissances en lecture de spectrogrammes, sous la forme d'un système expert, connaît une application immédiate d'un point de vue pédagogique. Il peut servir de support pour l'enseignement de la lecture de spectrogrammes.

Il reste à présent surtout à valider l'expertise dans le cadre de la reconnaissance continue, et pour cela il faut mettre au point une connexion entre les connaissances formalisées et une extraction automatique des indices acoustiques utilisés pour décrire le spectrogramme.

Ce travail pourra se poursuivre dans différentes directions:

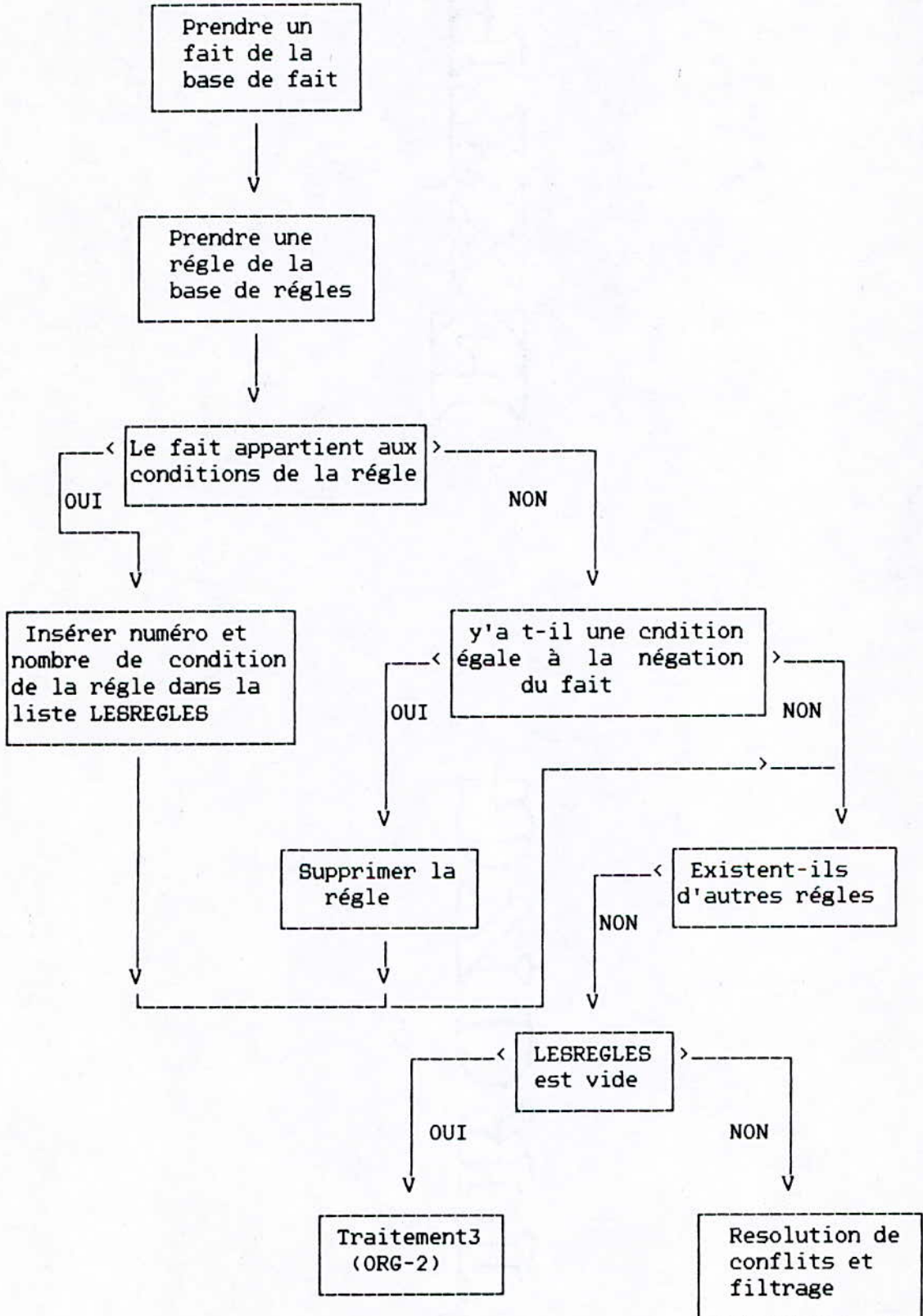
- Une approche "traitement d'images" dont l'objet consiste à intégrer les mécanismes propres à la perception visuelle du spectrogramme par l'expert.
- Une approche fondée sur la recherche d'indices acoustiques robustes.
- une approche pour développer des logiciels pour l'obtention des spectrogrammes numériques.

Enfin, une autre perspective serait l'utilisation des niveaux supérieurs.

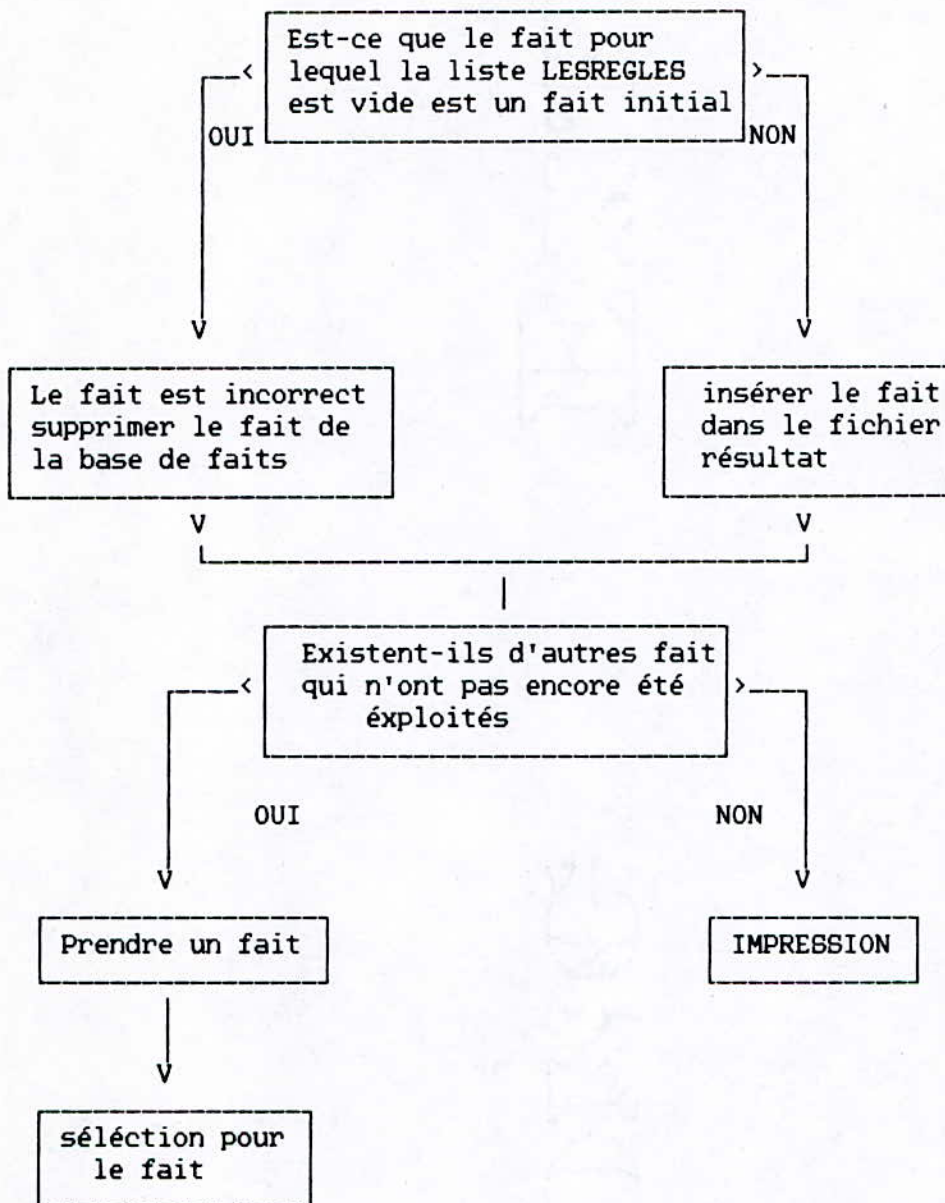
ORGANIGRAMMES

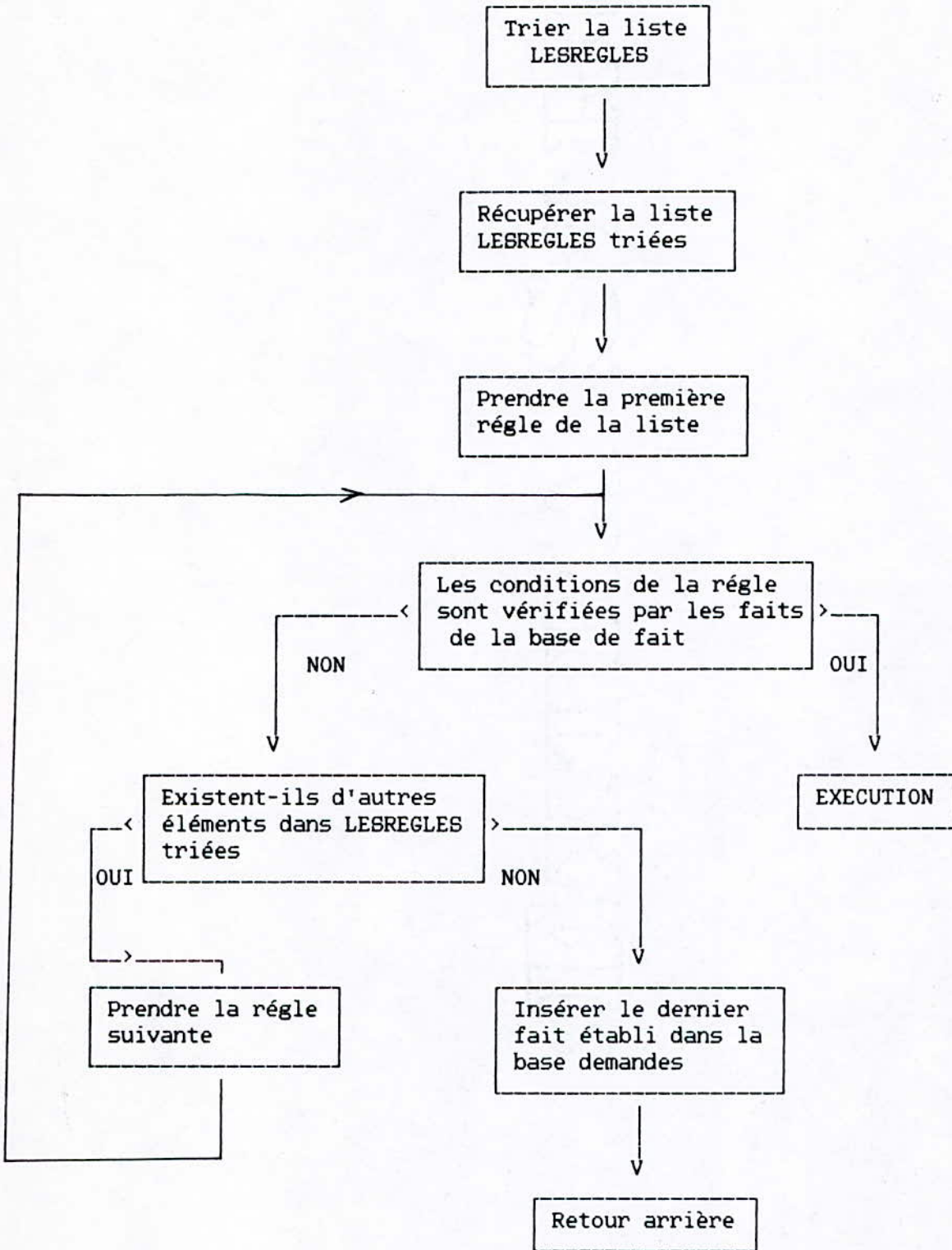
- 1- Phase selection
- 2- Phase traitement 3
- 3- Phase resolution de conflits
- 4- Phase execution
- 5- Phase retour arriere
- 6- Phase question a l'utilisateur

ORGANIGRAMMES

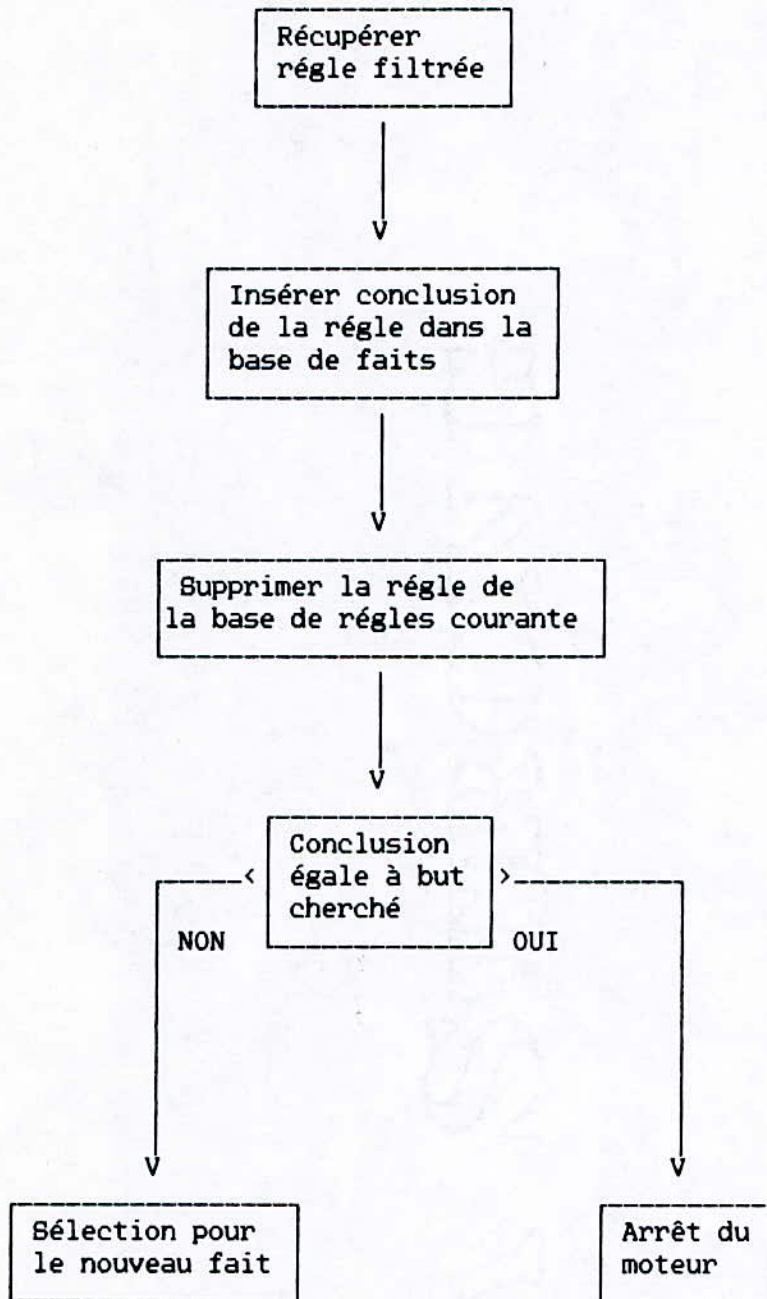


ORGANIGRAMME 1: La phase sélection.

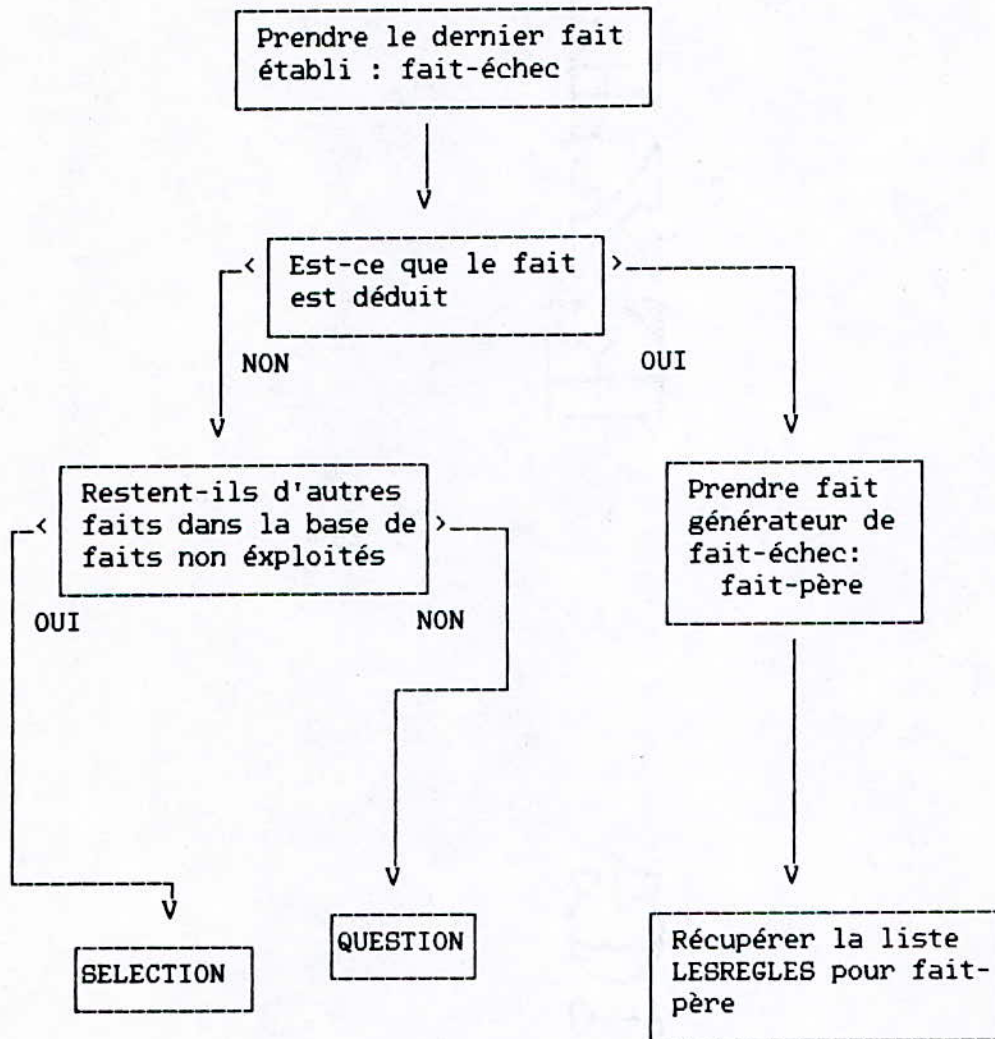




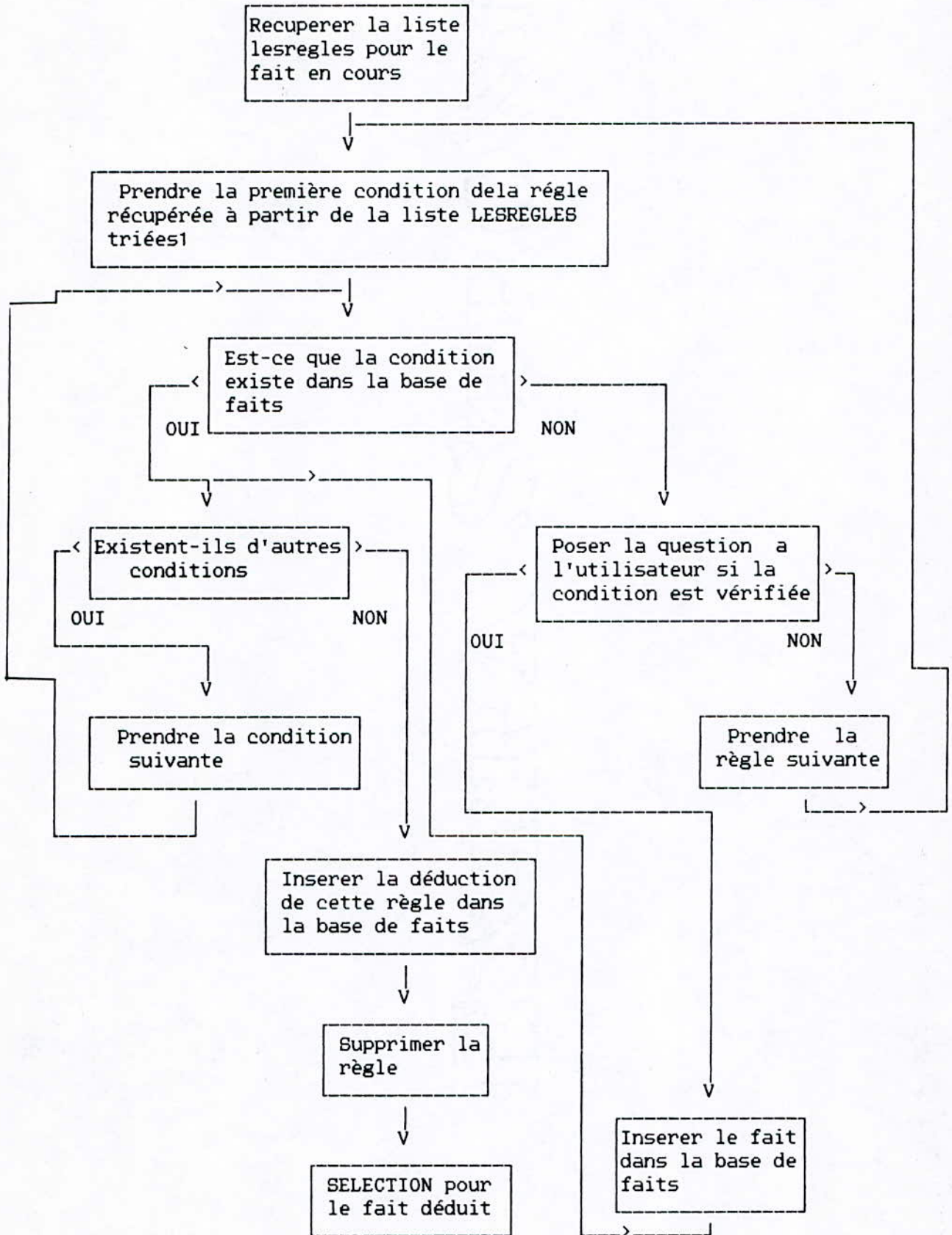
ORGANIGRAMME 3: La phase résolution de conflits et filtrage



ORGANIGRAMME 4:La phase exécution



ORGANIGRAMME 5:La phase retour arrière



ORGANIGRAMME6:La procédure de "question_utilisateur"

Bibliographie

Abréviations utilisées

- IEEE: Institut for Electrical and Electronic Engineers.
 ICASSP: International Conference on Acoustic, Speech and Signal Processing.
 ASI: Advanced Studies Institute.
 SNED: Société Nationale d'Édition et de Distribution.
 INRIA: Institut National de Recherche en Informatique et Automatique

- [BON 86] A.BONNEAU G.MERCIER M.ROSSI,
 Décodage acoustico_phonétique à l'aide d'un système
 Serac_Iroise,
 15ème journée d'étude sur la parole "AIX_EN_PROVENCE" mai 86.
-
- [BON 85] A.BONNEAU M.ROSSI G.MERCIER,
 Hierarchical recognition of french vowels by expert system
 Iroise_Serac,
 Séminaire Franco_Suédois fev 85
-
- [BON 84] A.BONNET,
 L'intelligence artificielle,
 Promesses et réalités 1984.
-
- [BOR 85] A.BORLAND,
 Turbo_prolog,
 Manuel d'utilisation 1985.
-
- [BUL 86] R.BULOT H.MELONI,
 Un système de traitement des connaissances pour le décodage
 acoustico_phonétique,
 12ème ICA, Montréal, août 86
-
- [BUL 87] R.BULOT,
 Techniques d'intelligence artificielle pour la reconnaissance de
 la parole,
 Thèse de Doctorat (spécialité informatique) Université
 Aix_Marseille II
-
- [CAR 84] N.CARBONELL D.FOHR J.P.HATON F.LONCHAMP,
 An expert system for the acoustic reading of french
 spectrograms,
 IEEE ICASSP N°3 1984.
-
- [COM 81] C.COMBESURE,
 20 listes de dix phrases phonétiquement équilibrées,
 Revue d'acoustique N° 56 1981

- [CLO 85] W.CLOCKSIN,
Programmer en prolog,
Edition Eyrolles 1985.
-
- [DEM 84] R.DEMORI M.GILLOUX J.VAISSIERE G.MERCIER,
Integration of acoustic, phonetic, prosodic and lexical
knowledge in an expert system for speech understanding,
IEEE 1984
-
- [EME 77] E.EMERIT,
Cours de phonétique acoustique,
Edition SNED, 1977.
-
- [FAR 85] F.FARREMY M.GHALLAB,
Elements d'intelligence artificielle,
Edition Hermès 1985
-
- [GAL 85] H.GALLAIRE,
Représentation des connaissances,
Revue la recherche N°170 Octobre 1985
-
- [GAN 85] J.G.GANASCIA,
La conception des systèmes experts,
Revue la recherche N°170 Octobre 1985
-
- [GON 85] M.GONDRAN,
Introduction aux systèmes experts,
Edition Eyrolles 1985
-
- [GRA 86] M.GRAVERO R.PIERACCINI,
Definition and evaluation of phonetic units for speech
recognition by HIDDEN MARKOV models,
ICASSP page 2235-2238 1986.
-
- [HAS 86] N.HASSEINE M.OTMANI
Etude et réalisation d'un synthétiseur,
Projet de fin d'études E.N.P.A Janvier 1986.
-
- [HAT 87_1] J.P.HATON,
Knowledge-based approaches in acoustic-phonetic decoding
of speech,
NATO ASI Speech processing, Bad Windsheim, RFA, juillet 87.
-
- [HAT 87_2] J.P.HATON N.CARBONELL D.FOHR,
Aphodex, an acoustic-phonetic decoding expert system,
Int J pattern recognition and artificial intelligence
N° 1, 2 1987
-
- [HAT 85] J.P.HATON,
A frame language for the control of phonetic decoding in
continuous speech recognition,
IEEE 85

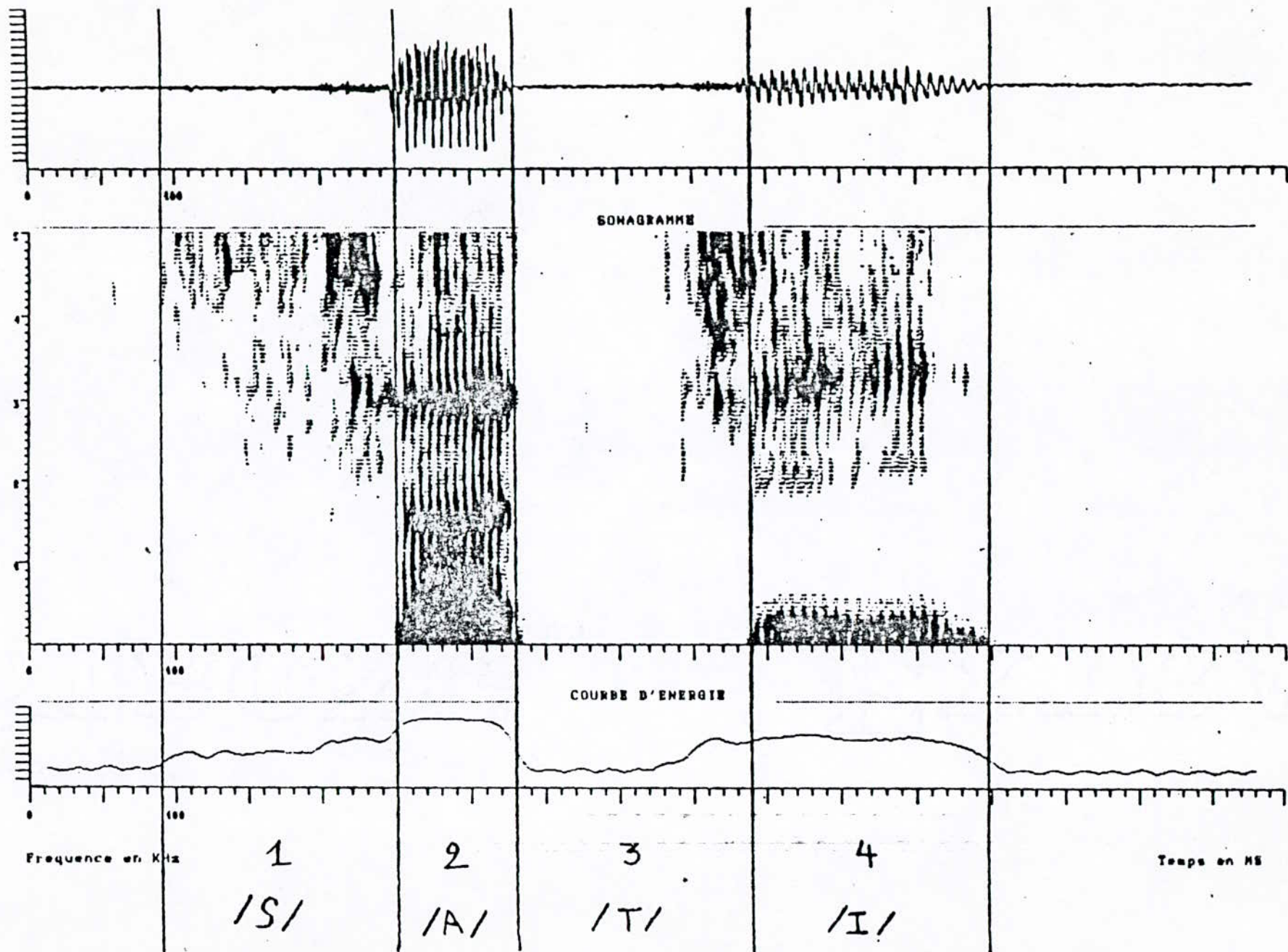
- [HAT 87_3] J.P.HATON,
Principe et technique de la reconnaissance de la parole,
Séminaire Toulouse 1987.
-
- [JAM 85] M.JAMES,
Introduction à l'intelligence artificielle sur micro ordinateur,
Edition Eyrolles 1985
-
- [JOH 83] J.JOHANNSEN S.ROSS,
A speech spectrogram expert,
ICASS (83) IEEE
-
- [KAU 65] KAUFMANN,
Programmation dynamique,
Edition Dunod 1965.
-
- [KUN 81] M.KUNT,
Traitement numérique des signaux,
Edition Dunod 1981.
-
- [LIF 80] LIFERMANN,
Méthodes rapides de transformation du signal,
Edition Masson 1980.
-
- [LIE 77] LIENARD,
Processus de la communication parlée,
Edition Masson 1977.
-
- [MEL 82] MELONI,
Etude et réalisation d'un système de reconnaissance automatique
de la parole,
Thèse de doctorat en informatique Université de Marseille 1982
-
- [MEM 83] D.MEMMI M.ESKENAZI J.MARIANI,
Un système expert pour la lecture des sonagrammes,
Speech communication N°2-3 1983.
-
- [MER 84] G.MERCIER M.GILLOUX C.TARRIDEC J.VAISSIERE,
From KEAL to SERAC: A new rule-based expert system for speech
recognition,
Nato ASI Bonas JULY 1984.
-
- [MIC 87_1] REVUE MICRO-SYSTEMES,
Turbo prolog,
Janvier, février, Mars 1987.
-
- [MIC 87_2] REVUE MICRO-SYSTEMES,
Réseaux sémantiques,
Mai 1987.
-
- [MIC 87_3] REVUE MICRO-SYSTEMES,
Traitement de l'incertain,
Juin 1987.

- [MIC 87_4] REVUE MICRO-SYSTEMES,
Systèmes experts,
Septembre, Octobre, Novembre, Décembre 1987.
-
- [REC 87] RECUEIL INRIA
Recherche et développement dans l'industrie de la langue,
INRIA Décembre 1987.
-
- [SAK 78] H.SAKOE CHIBA,
Dynamic Programming Algorithm optimization for spoken word
recognition,
IEEE Février 1978.
-
- [SEK 86] B.SEKHERI A.SAHED,
Etude et réalisation d'un micro système expert d'aide au
diagnostic des pannes de T.V,
Thèse de projet de fin d'études E.N.P.A Juin 1986.
-
- [STE 86] P.E.STERN M.ESKENAZI D.MEMMI,
An expert system for speech spectrogram reading,
ICASSP TOKYO 1986.
-
- [ZUE 85] V.W.ZUE,
Speech spectrogram reading,
M.I.T Special summer course 1985.
-

ANNEXE

- 1- Spectrogramme
- 2- Description du spectrogramme
- 3- Ensembles de regles
- 4- Code utilise

SPECTROGRAMME DU MOT: /SATI/



DESCRIPTION DU SPECTROGRAMME

----- Description du segment 1 -----

```
fait1(1,1,1,[[ "il", "ya", "du", "bruit" ], ["vrai"]])  
fait1(1,1,1,[[ "aspect", "peu", "marque" ], ["vrai"]])  
fait1(1,1,1,[[ "aspect", "tres-marque" ], ["vrai"]])  
fait1(1,1,1,[[ "il", "ya", "du", "bas_voisement" ]])
```

----- Description du segment 2 -----

```
fait1(1,2,1,[[ "nombre", "de", "formants", "est", "sup-eg 4" ], ["vrai"]])  
fait1(1,2,1,[[ "le", "formant-1", "est", "sup-eg 700" ], ["vrai"]])  
fait1(1,2,1,[[ "le", "formant-2", "est", "sup-eg 2100" ], ["vrai"]])
```

----- Description du segment 3 -----

```
fait1(1,3,1,[[ "silence" ], ["vrai"]])  
fait1(1,3,1,[[ "explosion" ], ["vrai"]])  
fait1(1,3,1,[[ "zone d'explosion", "est", "comprise <4000,5000>" ], ["vrai"]])  
fait1(1,3,1,[[ "transition", "du segment", "est", "dentale" ], ["vrai"]])
```

----- Description du segment 4 -----

```
fait1(1,4,1,[[ "la", "zone", "du formant-1", "est", "egal 300" ], ["vrai"]])  
fait1(1,4,1,[[ "la", "zone", "du formant-2", "est", "egal 2100" ], ["vrai"]])  
fait1(1,4,1,[[ "ecart-sup <300,2100,1500>" ], ["vrai"]])  
fait1(1,4,1,[[ "la", "zone", "du formant-1", "est", "inf-eg 300" ], ["vrai"]])
```



```

regle([14,4],[["il","ya","du","bruit"],["vrai"]],[["l","aspect","est",
"marque"],["vrai"]],[["l","aspect","est","tres_marque"],["vrai"]],[["il",
"ya","du","bas_voisement"],["faux"]],[["le","phoneme","est","un \S\ + 16
"],["vrai"]])
-----
regle([17,4],[["le","phoneme","est","un \S\ + 16"],["vrai"]],[["limite",
"basse","2800"],["vrai"]],[["sup_eg","","2500"],["vrai"]],[["rien",
"dessous","2000"],["vrai"]],[["le","phoneme","est","un \S\ + 40"],
["vrai"]])
-----
regle([18,2],[["il","ya","du","bruit"],["vrai"]],[["rien","dessous",
"1000"],["vrai"]],[["le","phoneme","est","un \Z\ - 26"],["vrai"]])
-----
regle([19,3],[["le","phoneme","est","un \S\ + 40"],["vrai"]],[["la",
"zone","de","concentration","h_f","est","","4600"],["vrai"]],[["
comprise","entre","<4600,4500,5000>"],["vrai"]],[["le","phoneme","est",
"un \S\ + 79"],["vrai"]])
-----
regle([20,3],[["il","ya","du","bruit"],["vrai"]],[["la","zone","de",
"concentration","h_f","est","","4600"],["vrai"]],[["comprise","entre",
"<4600,4500,5000>"],["vrai"]],[["le","phoneme","est","un \Z\ + 32"],
["vrai"]])
-----
regle([21,3],[["le","phoneme","est","un \S\ + 79"],["vrai"]],[["la",
"zone","de","concentration","b_f","est","","3100"],["vrai"]],[["comprise
","entre","<3100,3050,3400>"],["vrai"]],[["le","phoneme","est","un \S\
+ 90"],["vrai"]])
-----
regle([47,2],[["le","phoneme","est","un \S\ + 90"],["vrai"]],[["bien",
"decoupe","partout"],["vrai"]],[["fricative","sourde"],["vrai"]],[["le",
"phoneme","est","un \S\ + 91"],["vrai"]])
-----
regle([43,1],[["silence"],["vrai"]],[["le","phoneme","est","une occlu-
sive sourde + 60"],["vrai"]])
-----
regle([44,1],[["le","phoneme","est","une occlusive sourde + 60"],["vrai"
]],["le","phoneme","est","un \T\ + 20"],["vrai"]])
-----
regle([45,2],[["le","phoneme","est","un \T\ + 20"],["vrai"]],[["explosion
"],["vrai"]],[["le","phoneme","est","un \T\ + 33"],["vrai"]])
-----
regle([46,3],[["le","phoneme","est","un \T\ + 33"],["vrai"]],[["de",""
longueur","sup_moy"],["vrai"]],[["comprise","entre","<sup-moy,moy,sup-
-moy>"],["vrai"]],[["le","phoneme","est","un \T\ + 44"],["vrai"]])
-----
regle([47,3],[["le","phoneme","est","un \T\ + 44"],["vrai"]],[["explosion
"],["vrai"]],[["la","zone","d'","explosion","est","comprise","entre",""
<4000,5000>"],["vrai"]],[["le","phoneme","est","un \T\ + 67"],["vrai"]])
-----
regle([48,3],[["le","phoneme","est","un \T\ + 67"],["vrai"]],[["la",""
transition","du","segment","est","dentale"],["vrai"]],[["c'","est","une",
"occlusive","sourde","certain","ou","une","occlusive","sonore","certain"]
,["vrai"]],[["le","phoneme","est","un \T\ + 84"],["vrai"]])

```

```
regle([49,2],[[["la","zone","du","formant_1","est","egal","à","300"],["vrai"]],[["la","zone","du","formant_2","est","egal","à","2100"],["vrai"]]],[[["le","phoneme","est","une voyelle ecartee + 70"],["vrai"]]])
```

```
regle([50,3],[[["la","zone","du","formant_1","est","egal","à","300"],["vrai"]],[["la","zone","du","formant_2","est","egal","à","2100"],["vrai"]]],[[["l'","ecart_sup_moy","est","compris","entre","<300,2100,1500>"],["vrai"]]],[[["le","phoneme","est","un \I\ + 35"],["vrai"]]])
```

```
regle([51,2],[[["le","phoneme","est","un \I\ + 35"],["vrai"]],[["la","zone","du","formant_1","est","inf_eg","à","300"],["vrai"]]],[[["le","phoneme","est","un \I\ + 55"],["vrai"]]])
```

```
regle([52,4],[[["le","phoneme","est","un \I\ + 55"],["vrai"]],[["la","zone","du","formant_2","est","sup_eg","à","1900"],["vrai"]],[["le","segment","est","une","vocalique"],["vrai"]]],[[["le","phoneme","est","un \I\ + 78"],["vrai"]]])
```

```
regle([53,3],[[["le","phoneme","est","un \I\ + 78"],["vrai"]],[["bas_f_1"],["vrai"]],[["le","segment","est","une","voyelle","ecartee","certain"],["vrai"]]],[[["le","phoneme","est","un \I\ + 92"],["vrai"]]])
```

```
regle([22,1],[[["le","nombre","de","formants","est","sup_eg","à","4"],["vrai"]]],[[["le","phoneme","est","une voyelle + 85"],["vrai"]]])
```

```
regle([23,2],[[["le","phoneme","est","une voyelle + 85"],["vrai"]],[["le","formant_1","est","sup_eg","à","700"],["vrai"]]],[[["haut f1 + 70"],["vrai"]]])
```

```
regle([25,2],[[["haut f1 + 70"],["vrai"]],[["la","zone","du","formant_1","est","sup-eg","à","450"],["vrai"]]],[[["haut f1 + 85"],["vrai"]]])
```

```
regle([48,2],[[["bien","decoupe","partout"],["vrai"]],[["le","segment","est","une","fricative","sourde"],["vrai"]]],[[["le","phoneme","est","un \F\ + 25"],["vrai"]]])
```

```
regle([27,4],[[["haut f1 + 85"],["vrai"]],[["la","zone","du","formant_1","est","sup-eg","550"],["vrai"]],[["l'","aspect","est","tres_marque"],["vrai"]],[["la","longueur","est","inf_moy"],["vrai"]]],[[["le","phoneme","est","un \A\ + 30"],["vrai"]]])
```

```
regle([40,5],[[["le","phoneme","est","un \A\ + 30"],["vrai"]],[["la","zone","du","formant_1","est","comprise","entre","<700,850,800>"],["vrai"]],[["la","zone","du","formant_2","est","comprise","entre","<1500,1250,1650>"],["vrai"]],[["la","zone","du","formant_3","est","comprise","entre","<3000,2800,3000>"],["vrai"]],[["la","zone","du","formant_4","est","comprise","entre","<3800,3750,3850>"],["vrai"]]],[[["le","phoneme","est","un \A\ + 72"],["vrai"]]])
```

```
regle([42,4],[[["le","phoneme","est","un \A\ + 72"],["vrai"]],[["la","zone","du","formant_1","est","comprise","entre","<700,700,800>"],["vrai"]],[["la","zone","du","formant_2","est","comprise","entre","<1500,1250,1650>"],["vrai"]],[["la","zone","du","formant_3","est","comprise","entre","<3000,2200,3000>"],["vrai"]]],[[["le","phoneme","est","un \A\ + 75"],["vrai"]]])
```

```
regle([55,2],[[["le","phoneme","est","une","voyelle_compacte"],["vrai"]],  
[["la","partie","gauche","du","formant_2","est","montante"],["vrai"]],  
[["le","phoneme","est","une labiale - 100"]]])
```

```
regle([56,3],[[["le","phoneme","est","une","voyelle_compacte"],["vrai"]],  
[["la","partie","droite","du","formant_2","est","montante"],["vrai"]],  
[["la","position","du","segment","est","finale"],["faux"]],["le",  
"phoneme","est","une labiale - 100"]]])
```

```
regle([57,3],[[["le","phoneme","est","une","voyelle_compacte"],["vrai"]],  
[["la","partie","gauche","du","formant_4","est","montante"],["vrai"]],  
[["la","position","du","segment","est","finale"],["faux"]],["le",  
"phoneme","est","une dental + 50"]]])
```

```
regle([58,3],[[["le","phoneme","est","une","voyelle_compacte"],["vrai"]],  
[["la","partie","droite","du","formant_4","est","montante"],["vrai"]],  
[["la","position","du","segment","est","finale"],["faux"]],["le",  
"phoneme","est","une dental + 50"]]])
```

```
regle([59,3],[[["le","phoneme","est","une","voyelle_compacte"],["vrai"]],  
[["la","partie","gauche","du","formant_3","est","montante"],["vrai"]],  
[["la","position","du","segment","est","finale"],["faux"]],["le",  
"phoneme","est","une dental + 50"]]])
```

```
regle([60,3],[[["le","phoneme","est","une","voyelle_compacte"],["vrai"]],  
[["la","partie","droite","du","formant_3","est","montante"],["vrai"]],  
[["la","position","du","segment","est","finale"],["faux"]],["le",  
"phoneme","est","une dental + 50"]]])
```

```
regle([61,3],[[["le","phoneme","est","une","voyelle_ecartee"],["vrai"]],  
[["la","partie","droite","du","formant_2","est","horizontale"],["vrai"]],  
[["la","position","du","segment","est","finale"],["faux"]],["le",  
"phoneme","est","une dental + 35"]]])
```

```
regle([62,4],[[["le","phoneme","est","une","voyelle_basse_ecartee"],  
["vrai"]],["la","partie","droite","du","formant_2","est","montante"],  
["vrai"]],["la","partie","droite","du","formant_3","est","descendante"],  
["vrai"]],["la","position","du","segment","est","finale"],["faux"]],  
[["le","phoneme","est","une palatale + 70"]]])
```

```
regle([63,4],[[["le","phoneme","est","une","voyelle_basse_ecartee"],  
["vrai"]],["la","partie","gauche","du","formant_2","est","montante"],  
["vrai"]],["la","partie","gauche","du","formant_3","est","descendante"],  
["vrai"]],["la","position","du","segment","est","finale"],["faux"]],  
[["le","phoneme","est","une palatale + 80"]]])
```

```
regle([64,4],[[["le","phoneme","est","une","voyelle_ecartee"],["vrai"]],  
[["la","partie","gauche","du","formant_2","est","descendante_rapide"],  
["vrai"]],["la","partie","gauche","du","formant_3","est","descendante"],  
["vrai"]],["la","position","du","segment","est","finale"],["faux"]],  
[["le","phoneme","est","une labiale + 80"]]])
```

Les résultats obtenus pour l'exemple du spectrogramme SATI

S + 91

A + 75

T + 84

I + 92

Phonème	:	Exemple	:	Codage
/i/	:	lit	:	I
/e/	:	clé	:)
/ɛ/	:	gél	:	(
/y/	:	suc	:	U
/œ/	:	leur	:	œ
/ɛ̃/	:	petit	:	E
/ɔ̃/	:	fau	:	°
/a/ et /ɑ/	:	patte et pâte	:	A
/ɔ/	:	sol	:	+
/o/	:	saule	:	o
/u/	:	fou	:	v
/ɛ̃/ et /œ̃/	:	brin et brun	:	<
/ɑ̃/	:	chant	:	#
/ɔ̃/	:	bon	:	/
/y/	:	lui	:	7
/u/	:	oui	:	-
/j/	:	yole	:	Y
/s/	:	got	:	s
/z/	:	zèbre	:	z
/ʃ/	:	chat	:	X
/ʒ/	:	jour	:	j
/f/	:	fou	:	F
/v/	:	vin	:	V
/n/	:	nom	:	N
/m/	:	mont	:	M
/ɲ/	:	digne	:	Z
/l/	:	long	:	L
/r/	:	Rond	:	R
/p/	:	Pont	:	P
/b/	:	bon	:	B
/t/	:	ton	:	T
/d/	:	don	:	D
/k/	:	cou	:	K
/g/	:	gond	:	G
silence	:		:	§