79/53

CENTRE DE RECHERCHE EN INFORMATIQUE DE NANCY

Sc N 79/47 A

chateau du montet
54500 vandœuvre les nancy

Jean François MARI

Contribution à l'analyse syntaxique et à la recherche lexicale en reconnaissance du discours continu.

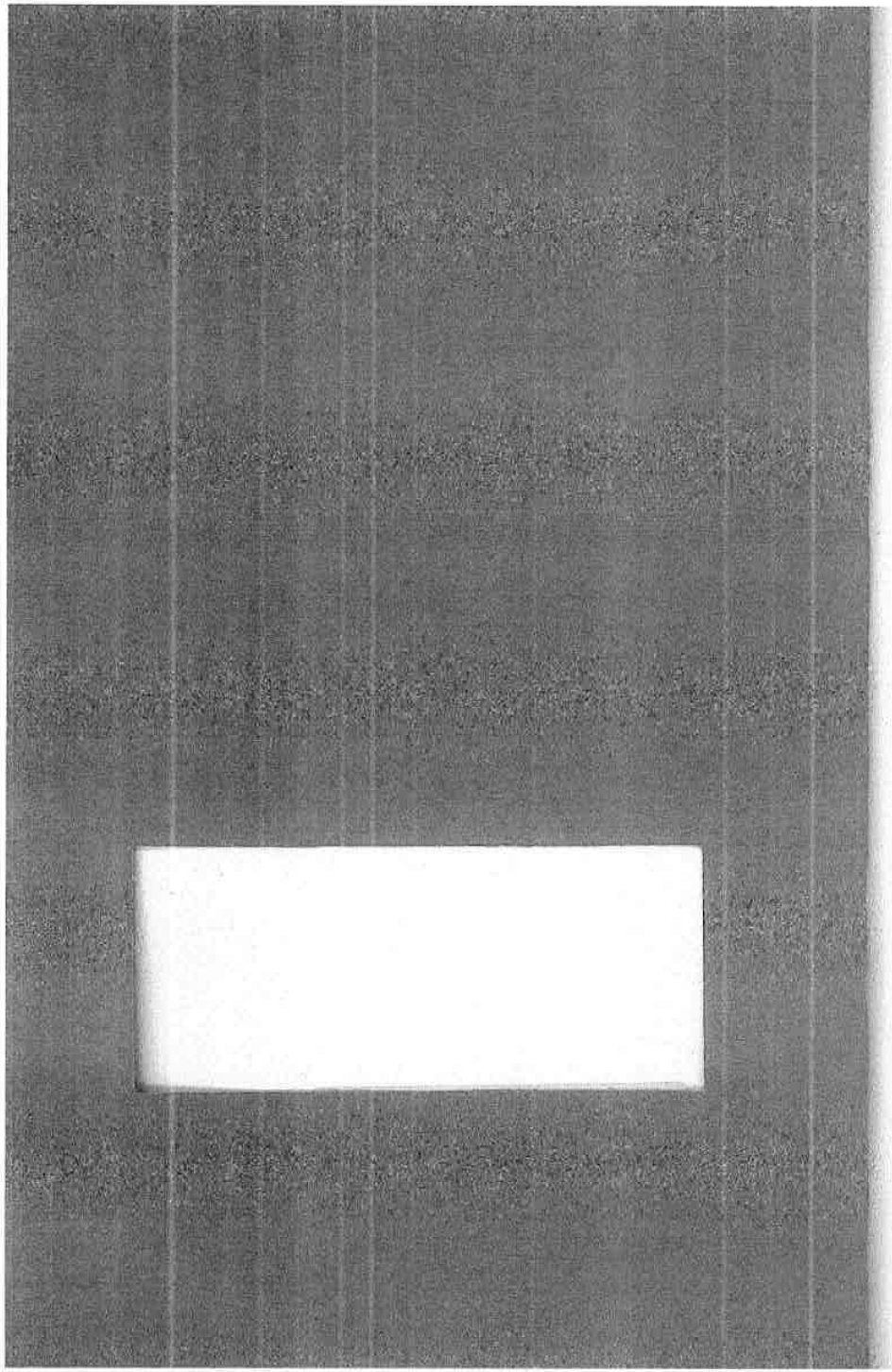
THESE

79-R-048

UNIVERSITE NANCY BIBLIOTHEQUE

BIBLIOTHEQUE SCIENCES
D 095 104515 4

univers
université de nancy 2
institut national polytechnique de lorraine



Jean François MARS

Contribution à l'analyse syntaxique et à la recherche lexicale
en reconnaissance du discours continu

THESE

UNIVERSITE DE NANCY
BIBLIOTHEQUE
79 048

Un versité de Nancy 1
U.E.R. de Mathématiques

Contribution à l'analyse syntaxique et à la recherche lexicale en reconnaissance du discours continu.

THESE

Présentée pour l'obtention du
doctorat de spécialité
de mathématiques appliquées (informatique)

par Jean François MARI
soutenue publiquement le 13 Juin 1979

JURY :	Président	C. PAIR
	Examinateurs	S. CASTAN
		M. GRIFFITHS
		J.P. HATON
		J.M. PIERREL

C'est avec plaisir que je rédige le dernier chapître de ma thèse... Celui consacré aux remerciements.

Je veux remercier en premier Jean-Paul HATON pour l'aide qu'il m'a apportée pendant mon travail par sa patience et sa compétence.

Je remercie beaucoup Monsieur C. PAIR pour s'être intéressé à ce travail et pour avoir accepté de présider mon jury.

Je suis reconnaissant à Monsieur M. GRIFFITHS pour les conseils qu'il m'a donnés et pour sa présence parmi nous.

Je remercie Monsieur CASTAN qui a spontanément accepté de prendre connaissance de mon mémoire et de participer au jury.

Que Jean-Marie PIERREL soit sûr de toute ma sympathie. Nous avons déjà travaillé souvent ensemble et c'est avec plaisir que je le vois siéger dans ce jury.

Je ne peux pas dans ces quelques lignes exprimer toute ma reconnaissance à tous ceux qui m'ont aidé dans les différents domaines d'une réalisation de thèse : travail de réflexion, frappe du manuscrit, tirage; j'espère qu'ils ne m'en voudront pas.

Le 4 juin 1979

Jean-François MARI

TABLE DES MATIERES

I	Notre approche de la reconnaissance automatique de la parole	1
II	Le niveau lexicographique	10
III	Le niveau syntaxique	44
IV	Application à un langage donné	104

CHAPITRE I

NOTRE APPROCHE DE LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

I. La reconnaissance de la parole : intérêt, champ d'application et axes de recherche

II. Le processus de communication parlée

III. Les sources d'informations et les traitements associés

III.1. Principes généraux

III.2. Les sources d'information indépendantes du contexte

III.3. Les sources d'information contextuelles

IV. Conclusions

NOTRE APPROCHE DE LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

Dans ce chapitre nous allons situer notre travail dans la recherche en reconnaissance -compréhension de la parole. Cela nous obligera, en premier lieu, à donner une description générale de ce domaine. Si certains aspects, certaines réalisations ne seront pas abordés, c'est que leur évocation n'aurait pas permis de mieux cerner notre activité ou mieux comprendre nos choix.

Pour avoir une vue plus complète sur l'ensemble de la recherche en reconnaissance de la parole, on se reportera à (1) ou (29).

I) La reconnaissance de la parole : Intérêt, champ d'application et axes de recherche

La reconnaissance de la parole s'est développée des 1948-1950 avec comme première motivation, la recherche d'un mode de transmission de la voix à faible bande. Actuellement, le but essentiel est la réalisation de systèmes de communication homme-machine sur support vocal, permettant une amélioration de certains postes de travail. En effet, l'usage de la voix comme moyen de communication présente de fait des avantages certains qui peuvent justifier les recherches pratiques dans ce domaine (1) :

- vitesse, simplicité et fiabilité d'une communication orale en comparaison avec d'autres modes : clavier, écriture, etc ... Mais cet argument peut être combattu car il ne tient pas compte de la faculté d'adaptation de l'homme. Il est probable qu'il n'y aura pas de système de compréhension automatique pour remplacer le service d'assistance à l'annuaire des PTT, chaque usager ayant pris l'habitude de présenter des demandes par écrit par l'intermédiaire d'un clavier que l'administration commence à distribuer.

.. ....

- simultanéité possible avec d'autres tâches permettant une communication en parallèle et préservant une liberté de mouvement du locuteur.

Ces avantages sont clairement mis en évidence dans les systèmes expérimentaux réalisés en FRANCE et à l'étranger, malgré leur caractère encore partiel.

Comme champs d'application on peut citer, entre autres :

- la commande d'une machine outil (2)
- le tri des paquets (postes, aéroports)
  Un employé prononce l'adresse du destinataire et le paquet est acheminé automatiquement vers le sac postal correspondant.
- la consultation d'un centre de renseignements.

Plusieurs de ces applications ont été ou sont étudiée dans notre laboratoire. Les algorithmes présentés dans les chapitres deux et trois de cette thèse sont évalués grâce à un langag de communication avec un standard téléphonique et nous les utiliserons dans le système MYRTILLE II capable de comprendre des demandes présentées en langue naturelle à un centre de renseignements météorologiques.

Il faut distinguer deux grandes classes de système de reconnaissance - compréhension de la parole :

- les systèmes de reconnaissance de mots isolés
- les systèmes de reconnaissance de la parole continue.

On peut encore considérer deux familles de systèmes de reconnaissance - compréhension de la parole continue, bien que la limite entre les deux soit plus floue :

a) les systèmes de reconnaissance de langages artificiels

b) les systèmes de compréhension de langages naturels.

.....

Le langage de communication avec un standard téléphonique qui sert dans ce travail de langage d'application fait partie de la première famille. Celle-ci regroupe en fait des langages formés de phrases en langue naturelle mais orientées vers une activité précise, si bien que le vocabulaire qui les constitue, leur syntaxe et leur sens sont, soit limités soit rigoureusement définis.

Les différences entre ces deux catégories de langages seront abordées plus précisément dans le chapitre III.

Le niveau de reconnaissance acoustico-phonétique d'un système est chargé de délivrer une transcription du message sous forme de treillis de phonèmes (voir figure IV.3) entachés d'erreur. Ce treillis sert de donnée à d'autres algorithmes du système et nous avons travaillé sur des treillis qui contenaient environ 40 % de phonèmes erronés afin de tenir compte des limites des systèmes acoustiques actuels.

Les travaux concernant le discours continu ont mis en évidence la différence existant entre reconnaissance et compréhension de la parole. La reconnaissance consiste à retrouver dans une phrase l'ensemble des éléments (mots, syllabes, phonèmes) la composant. En compréhension, au contraire, on s'attache à saisir le sens de la phrase, même si la reconnaissance est imparfaite, de façon à pouvoir exécuter l'action demandée. Cette seconde démarche s'apparente à celle du cerveau humain. Une bonne compréhension nécessite bien entendu une reconnaissance suffisamment correcte, mais la nature même du message parlé limite cette reconnaissance et rend nécessaire le recours à des processus de compréhension. Nous verrons au chapitre III comment l'analyseur syntaxique présenté met en oeuvre des informations sémantiques pour poursuivre une reconnaissance ayant échoué dans un premier temps.

Une contrainte importante en reconnaissance de la parole est la nécessité d'avoir une réponse rapide du système de compréhension afin de maintenir un "dialogue" entre l'homme et

.....

la machine. Cette contrainte fait qu'il faut envisager des solutions simples et efficaces, éventuellement sur des machines multi-processeurs.

La compréhension fait appel à des sources d'information très variées depuis l'acoustique jusqu'à la pragmatique, et la mise en oeuvre de ces sources d'information n'est pas résolue d'une façon générale. Pour notre part, nous nous sommes intéressés dans ce travail à la mise en oeuvre de la source d'information syntaxique en développant un algorithme d'analyse syntaxique adapté à la reconnaissance du discours continu. La mise en oeuvre de cette source d'information nous a obligés à envisager des interactions avec d'autres processus spécifiques des sources d'information lexicale et pragmatique. Ainsi, à partir d'un algorithme d'analyse syntaxique débutant n'importe où dans la phrase, nous avons développé un algorithme de localisation de mots, l'ensemble étant supervisé par une procédure spécifique de l'application.

Avant de décrire précisément ces trois procédures, il semble intéressant de décrire le processus de communication parlée qui nous permettra de cerner les différents niveaux de traitement d'un système de reconnaissance - compréhension dans leurs positions respectives.

II) Le processus de communication parlée

La communication parlée consiste en un échange d'informations bidirectionnel entre deux locuteurs.
Une idée est issue du niveau supérieur de l'intelligence et traduite aux niveaux inférieurs selon le schéma de la figure I.1 jusqu'à l'obtention du support qu'est le signal vocal, le processus inverse conduit à la compréhension.

.....

demande ⇓ parole

- niveau sémantique et pragmatique : lié aux connaissances a priori des locuteurs et au contexte de la conversation.
- niveau syntaxique : référence à la structure du langage
- niveau lexical : niveau des mots
- niveau acoustique : lié aux mécanismes de phonation et d'audition

Figure I.1
(d'après J.P. HATON)

De nombreuses sources d'information interviennent donc chez l'auditeur : le contexte de la conversation, (la connaissance consciente et inconsciente que chacun a de l'autre), le milieu dans lequel s'effectue la communication, l'information visuelle lorsque les personnes se voient. Si bien qu'un système de reconnaissance automatique de la parole, tout comme un auditeur moyen, devra mettre en oeuvre un ensemble de sources d'information hiérarchisées.

III) Les sources d'information et les traitements associés.

III.1. Principes généraux

Certains chercheurs ont longtemps cru que toute l'information nécessaire à la compréhension de la parole se trouvait dans le signal vocal ou dans les paramètres que l'on en extrait. Cette approche s'avère suffisante pour la reconnaissance d'un nombre limité de mots isolés mais elle est mise en défaut dès que l'on s'attaque aux phrases, comme en témoignèrent les échecs subis dans les années soixante dans le domaine des "machines à écrire phonétiques".

....

En réalité, nous venons de voir que le processus de communication orale doit être considéré comme un processus hiérarchique depuis la pensée du locuteur jusqu'à la pensée de l'auditeur. Chaque niveau de la hiérarchie amène une certaine redondance et contribue à la "robustesse" du message vocal : pensée (niveau sémantique) - phrase et mots (niveaux linguistiques) - encodage des mots - signal acoustique. En ce sens le traitement du signal acoustique et la connaissance du mode de production de ce signal sont des facteurs nécessaires et importants pour la compréhension finale, mais non suffisants.

Un système de compréhension doit faire appel aux diverses sources d'informations dont il dispose pour parvenir à la signification d'une phrase. Nous classerons schématiquement ces différentes sources - correspondant chacune à un niveau de traitement et souvent confondues avec lui - en deux classes :

- les sources indépendantes du contexte, c'est-à-dire du domaine d'application (acoustique, phonétique, phonologique, lexique et prosodique),
- les sources fonctions du domaine d'application (vocabulaire, syntaxe, sémantique et pragmatique).

Les niveaux de traitement de ce type mettent en oeuvre des techniques générales, réutilisables dans d'autres domaines. Ainsi, une version similaire de l'analyseur syntaxique décrit au chapitre III a été développée dans un système de reconnaissance de dessins et les techniques de localisation de mots sont aussi utilisées dans d'autres domaines de reconnaissance de formes comme nous le verrons dans le chapitre concernant le module de localisation de mots.

D'un point de vue de l'intelligence artificielle, ces connaissances spécifiques de la tâche à accomplir aident à limiter l'explosion lors de la recherche dans l'univers des solutions comme nous le mettons en évidence dans le chapitre IV.

...

Nous allons rapidement passer en revue ces sources d'information en précisant celles qui ont fait l'objet de notre étude dans ce travail et en donnant des références d'ouvrages pour celles que nous n'avons pas mises en oeuvre.

III.2. Les sources d'informations indépendantes du contexte.

a) niveau acoustico-phonétique

Lié à la représentation, au traitement, à la paramétrisation du signal vocal. Pour plus de détails on peut se reporter à (29, 14)

b) niveau phonologique et lexical

Lié à la représentation des mots du vocabulaire et à leurs altérations en contexte. Le chapitre II est une contribution apportée à ce domaine. Nous avons développé un algorithme de localisation de mots dans un treillis de phonèmes qui permet une comparaison élastique entre une forme de référence et une sous-chaîne quelconque extraite du treillis de phonèmes d'entrée. Nous avons voulu tenir compte dans cette comparaison des altérations phonologiques internes à un mot.

c) niveau prosodique

Nous n'avons pas abordé ce domaine, mais le module de localisation de mots développé peut être facilement modifiable pour prendre en compte des informations prosodiques (marqueurs de début de groupes prosodiques), comme c'est prévu dans le système MYRTILLE II (30).

III.3. Les sources d'information contextuelles

a) Vocabulaire

Si le niveau lexical met en oeuvre des techniques générales, le vocabulaire est bien entendu lié au domaine d'application. Le vocabulaire influe sur les performances du système de compréhension par sa taille et son degré d'ambiguïté.

.....

Le langage qui nous sert à tester les algorithmes d'analyse syntaxique et de localisation de mots comporte 30 mots ; il est défini au chapitre IV.

b) niveau syntaxique

Le point de départ de ce travail a été la réalisation d'un analyseur syntaxique de langages décrits par des grammaires hors contexte, capable de débuter l'analyse n'importe où dans le treillis. L'intérêt de cette stratégie est davantage développé dans le chapitre III. Il s'agissait de reprendre le système MYRTILLE I (3) articulé autour d'un analyseur descendant travaillant de la gauche vers la droite et d'évaluer ces performances avec cette nouvelle méthode d'analyse. En réalité celà nous a amené à réécrire la quasi totalité des algorithmes notamment l'algorithme de reconnaissance de mots qui devient un algorithme de localisation et de reconnaissance de mots. Le nouveau système garde les caractéristiques de l'ancien :

- le langage d'application est un paramètre du système
- structure modulaire pour en faciliter la maintenance
- le langage de programmation est le FORTRAN IV de base
- de plus, le système est orienté vers une architecture multi processeurs qui semble être une réponse à la contrainte de temps réel évoquée quelques lignes plus haut.

Il est apparu que la méthode d'analyse par ilôts de confiance développée dans le cadre des grammaires hors contexte était applicable aux grammaires de langue naturelle. En effet, un mode de représentation courant de la syntaxe est d'utiliser un réseau (réseau ATN (AS 4) où à noeuds procéduraux (12) et l'algorithme que nous avons développé peut être vu comme un algorithme de parcours de réseau récursif pouvant débuter en n'importe quel point du réseau.

c) niveau sémantique et pragmatique

Le type de langage qui nous a servi à valider notre approche possède une sémantique très liée à la syntaxe. Une ébauche de traitement sémantique est effectuée afin de permettre la

.....

resynchronisation du processus de reconnaissance après un échec de celle-ci.
Le niveau pragmatique est mis en oeuvre dans une procédure de l'analyseur. Dans notre application cette source d'informations nous permet de limiter la longueur des portions de treillis dans lesquelles s'effectue la reconnaissance de certains mots. Cet aspect est abordé au chapitre IV. Bien entendu cette procédure est spécifique de l'application considérée mais nous allons voir qu'elle s'écrit facilement à l'aide des procédures de reconnaissance (reconnaissance de mots où reconnaissance de phrases) développées dans les chapitres II et III.

IV) Conclusion

Notre travail se situe dans le cadre de la reconnaissance analytique du discours continu.

- Le chapitre II est une contribution à la recherche lexicale
- Le chapitre III est une contribution à l'analyse syntaxique en reconnaissance de la parole continue. Si l'analyseur décrit est orienté vers l'analyse des langages décrits par une grammaire hors contexte, il est néanmoins utilisable dans l'analyse des langues naturelles. Nous pensons l'utiliser dans le système MYRTILLE II qui est un système de compréhension de langage naturel (30).
- Le chapitre IV décrit une nouvelle version du système MYRTILLE I articulée autour d'un analyseur ascendant débutant en un point quelconque du treillis. Cet analyseur est couplé à un algorithme de localisation de mots et de vérification d'hypothèses. L'amélioration notable de performance obtenue justifient les choix faits dans la représentation de la syntaxe et dans la méthode de localisation de mots.

.....

CHAPITRE II

LE NIVEAU LEXICOGRAPHIQUE

I) Introduction : Que veut-on réaliser ?

II) Les difficultés de la recherche lexicale

III) Les différentes méthodes de recherche lexicale

A) Le problème de la comparaison de deux formes

a) Comparaison heuristique
b) L'analyse par synthèse
c) L'introduction d'un modèle stochastique

B) Le problème de l'extraction d'une sous-chaîne

IV) La méthode retenue

A) Idées directrices

B) Description du dictionnaire

B-1 Description des mots du lexique
B-2 Description des accès
B-3 Remarques

C) Recherche d'un mot à l'intérieur d'une chaîne de phonèmes

C-0 Données et résultats de l'algorithme de recherche
C-1 Recherche basée sur le principe de similitude local
C-3 Relations entre transitions et altérations phonologiques
C-4 La fonction de Ressemblance entre phonèmes
C-5 Exemples
C-6 Algorithme de recherche des chemins
C-7 Détermination des paramètres
C-8 Performances

I) INTRODUCTION : QUE VEUT-ON REALISER ?

Dans le cadre de la compréhension du discours continu il est nécessaire de retrouver à l'intérieur de la représentation du signal sonore les mots qui constituent ce message. La plupart des systèmes de compréhension travaillent sur une représentation sous forme de treillis ou de chaînes de phonèmes et déterminent la suite de mots employés en segmentant cette chaîne ou ce treillis. Ce problème de segmentation et d'identification ne peut pas être résolu sans mettre en oeuvre des sources d'information acoustiques, syntaxiques et sémantiques. En effet, alors que le segment intervenant dans un phonème a une existence physique plus ou moins marquée, le mot intervenant dans une phrase est un concept sémantique. Tout au plus peut-on retrouver grâce à l'existence de pause et à l'examen des variations du fondamental ou du rythme, des groupes de mots appelés groupes prosodiques (22). Pour illustrer ce problème on peut penser aux difficultés de compréhension d'un discours prononcé dans une langue étrangère inconnue. Nous sommes capables de discerner la plupart des phonèmes et d'isoler certains groupes de mots mais sans la connaissance de la langue, c'est-à-dire de sa syntaxe, de son vocabulaire et sans connaissance à priori du contexte de l'échange oral, il nous est impossible d'identifier et de localiser les mots employés.

Nous pouvons définir maintenant un niveau lexicographique de la façon suivante :

- une base de données constituant le vocabulaire de l'application. Cette base contient les représentations phonétiques de formes de référence et des procédures permettant de déduire dynamiquement de nouvelles formes (par exemple, des procédures conjuguant les verbes à partir de radicaux stockés). Sur cette base sont définis des accès suivant le type de consultation souhaitée (on peut vouloir accéder à un ensemble d'adjectifs de couleur ou à tous les mots débutant par une paire de syllabes donnée).
- un ensemble d'interface avec les autres niveaux : procédures d'échange et d'accès avec les niveaux sémantiques et syntaxiques, algorithmes de localisation et reconnaissance de mots,

....

procédures d'interaction avec les niveaux inférieurs acoustiques et phonétiques.

Dans ce chapitre, nous définirons successivement les composantes du niveau lexicographique réalisé. Ce niveau comme toutes les parties d'un système de reconnaissance automatique de la parole, va travailler à partir de données erronées et à interprétations multiples, c'est pourquoi nous commencerons ce chapitre par un paragraphe sur les difficultés de la recherche lexicale dans le discours continu.

II) LES DIFFICULTES DE LA RECHERCHE LEXICALE

Dans le cadre de la reconnaissance analytique de la parole, nous allons résumer les problèmes que pose la recherche des mots dans la transcription phonétique d'un discours continu telle qu'elle a été définie au chapitre précédent.

Ces difficultés peuvent être classées en quatre catégories suivant leurs origines :

- les frontières des mots ne sont pas apparentes et des altérations phonétiques apparaissent aux fonctions des mots.
- des altérations phonétiques apparaissent à l'intérieur des mots.
- les accents régionaux influencent la prononciation.
- les sytèmes acoustiques fournissent une transcription phonétique entachée d'erreurs.

Les deux premières difficultés sont étudiées en Phonétique Combinatoire. Une explication de ces phénomènes peut être donnée en terme des mécanismes de phonation. La production d'un message parlé est la dernière étape de l'encodage de l'idée que le locuteur veut faire passer. Celui-ci connaît le mot qu'il prononce ainsi que les précédents et les suivants, les mécanismes de prononciation qui sont continus s'influenceront mutuellement et donneront une unité aux mots, groupes de mots et phrases du message.

.....

Cette influence réciproque tend à produire un maximum d'effet pour un minimum d'effort dans l'élocution.
Décrivons plus précisément ces différentes catégories de difficultés :

A) Les frontières de mots ne sont pas apparentes

En effet le signal vocal est un semi continuum, de plus, pour donner une unité à un groupe de mots, on peut les souder grâce aux liaisons ce qui introduit des phonèmes de liaisons que l'on ne peut pas rattacher aux transcriptions phonétiques des mots voisins et qui pourtant dépendent du couple formé.

Par exemple on dira /l e z a r b r (ə)/ : les arbres
/l e a r i k o/ : les haricots

L'organisation des mots du vocabulaire en un lexique devra permettre de retrouver les liaisons d'un mot avec son suivant. Dans le lexique du projet ARIAL(10) chaque forme est représentée à l'aide d'un chaîne linéaire de phonèmes dans laquelle apparaissent les éventuelles liaisons antérieures et postérieures. L'apparition d'un phonème entre deux mots pour créer une unité entre ceux-ci n'est pas toujours due à un phonème de liaison ; souvent, pour faciliter une prononciation d'un groupe de consonnes, un phonème d'appui est intercalé.

Exemple : un arc de triomple / œ̃ n a r k ə d œ t r i ɔ̃ f /

La jonction de deux mots peut amener l'élision du dernier phonème dans le premier mot. Cela se voit dans le cas du [ə] mais cette élison n'est pas systématique, en effet le dernier [ə] permet d'éviter la proximité d'un groupe de consonnes.

Exemple : cet(te) table, une horribl(e) auto mais une horrible grange

B) Des altérations phonétiques apparaissent à l'intérieur des mots.

L'influence mutuelle des mouvements articulatoires voisins provoque à l'intérieur des mots des modifications des traits des phonèmes, les plus touchés étant les consonnes. Ces phonèmes sont depuis longtemps étudiés par les linguistes (11) et sont classés en différents groupes :

.....

assimilation
fusion
coarticulation
anticipation .....

Donnons quelques exemples :

- je voudrais prononcé rapidement perd le [ə], le couple [ʒ] [v] sera souvent fusionné et le système acoustique ne détectera qu'une seule constrictive.
- Dans le mot centenaire, le [ə] s'élide facilement et met en contact la paire ([t], [n]). Ces deux phonèmes ne se différencient que par le trait de nasalité, le passage du [t] au [n] se fera sans bouger la langue en nasalisant l'explosion du [t] ce qui peut provoquer une erreur du système acoustique.
- Le phénomène d'assimilation se rencontre dans l'altération sub quelquefois par le mot une heure et demie. Après l'élision du [ə] de demie la consonne [d] se trouve voisine de la nasale [m] Si cette nasale est anticipée le [d] se nasalise à son tour et devient un [n], si bien que l'on peut entendre : une heure et n-mie.

Certains chercheurs, notamment aux U.S.A. pour la la gue anglaise, ont cherché à décrire l'ensemble de ces phénomènes par un ensemble de règles dites règles phonologiques. Ce travail est fait pour l'anglais mais reste à faire pour le français. La connaissance de ces règles permettrait de mieux assumer les phénomènes de substitution, élision, insertion de phonèmes que ces altérations impliquent.

C) Les accents régionaux influencent la prononciation

Ces problèmes n'ont pas été abordés dans le cadre de la définition d'un niveau morphologique, car tous les systèmes acoustiques n'acceptent qu'un nombre limité de locuteurs coopérants ce qui élimine les locuteurs ayant un parler trop éloigné du parler standard que le niveau acoustique reconnaît.

....

D) Les niveaux acoustiques fournissent une transcription entachée d'erreurs

Ce problème a été abordé dans le premier chapitre. Afin de pallier en partie à ce problème, beaucoup de systèmes utilisent une matrice de confusion de phonèmes établie au vu des résultats des algorithmes de classification. Cette façon de faire est incomplète car elle ne tient pas compte des phénomènes d'élision ou d'insertion de phonèmes qu'une segmentation erronée peut introduire.

E) Conclusion : le rôle d'un lexique

Une grande partie de ces phénomènes d'altération pourrait être formalisée par l'introduction de règles phonologiques. Tous ces phénomènes, d'origines diverses, ont les mêmes conséquences
- substitution de phonèmes
- insertion de phonèmes
- élision de phonèmes
- et plus rarement, inversion de phonèmes.

Une tendance actuelle est de définir et de créer des lexiques évolués qui permettent d'une part d'accéder rapidement aux mots candidats lors d'une reconnaissance et d'autre part d'engendrer le maximum de variantes phonétiques à partir d'un radical et de règles de formation de désinences. Ces règles tiennent compte pour chaque mot : de son utilisation syntaxique (cas des conjugaisons de verbe), de son entourage (cas des liaisons) et éventuellement du type de locuteur (cas des accents régionaux).

Deux réalisations intéressantes ont vu le jour en France, il s'agit du lexique réalisé dans le cadre du projet ARIA à Toulouse par G. GOUARDERES (10) et du système MYRTILLE II du CRIN (12).

Dans le lexique du projet ARIA, un mot est obtenu à partir d'un radical et de l'utilisation de règles de création de suffixe et désinence. De cette façon retrouve t-on les différentes formes de verbe et d'adjectif. A chaque radical et désinence sont associées trois chaînes linéaires représentant les transcriptions graphique, phonétique et syllabique. Ainsi, une reconnaissance

.....

de mots par phonèmes ou par syllabes peut être faite à partir d'une transcription du signal, de même qu'une transcription graphème-phonème ou graphème-syllabe. Pour chaque forme engendrée il est tenu compte des liaisons antérieures et postérieures et de l'élision du /ə/.

Le lexique du système MYRTILLE II utilise le même principe de création des formes à partir d'un radical, de règles stockées dans une base et de procédures mais en plus, les formes sont classées par dépendance conceptuelle ce qui représente une intéraction entre les niveaux lexicaux et sémantiques.

Dans ce chapitre, nous décrirons un algorithme de reconnaissance de mots utilisant le lexique du système MYRTILLE I. La reconnaissance se fera à partir d'une forme de référence fournie a priori et tenant compte des altérations phonologiques survenant à l'intérieur d'un mot telles que élision, substitution et insertion de phonème. Une poursuite de notre travail de recherche consistera à définir un lexique capable de fournir cette forme de référence, soit à partir de divers paramètres d'un mot hypothèse (radical, utilisation syntaxique, possibilité de liaison), soit directement à partir du treillis phonétique lors d'une reconnaissance ascendante (par exemple pendant la création d'un treillis de mots).

III. LES DIFFERENTES METHODES DE RECHERCHE LEXICALE

Il faut distinguer deux problèmes en recherche lexicale, la recherche d'un mot à l'intérieur d'une représentation du signal qui est un problème d'extraction de sous-chaîne et la comparaison d'une forme inconnue avec une forme de référence: Ce dernier problème se rencontre lors d'une validation d'hypothèse quand les extrémités de la forme inconnue sont connues approximativement.

Le deuxième type de recherche est donc un cas particulier du premier puisqu'il ne reste plus qu'à comparer les deux sous-chaînes.

.....

Nous allons faire une étude bibliographique de ces deux problèmes en commençant par la comparaison de deux formes de longueurs sensiblement égales.

A) Le problème de la comparaison de deux formes

La principale difficulté est due au décalage temporel, avant de comparer il faut réaligner les deux formes. Nous allons voir que toutes les méthodes citées résolvent ce problème différemment.

Plusieurs solutions ont été développées à ce jour (1)

- la comparaison heuristique.
- l'introduction de modèle stochastique.
- l'analyse par synthèse.

a) La comparaison heuristique

Il s'agit de tenir compte des différents écarts de prononciation d'un locuteur par rapport à la prononciation standard et des limites des systèmes acoustiques.

Ces erreurs sont de trois types :

- élision de phonèmes
- substitution de phonèmes
- insertion de phonèmes

La forme de référence contient les différentes variantes de prononciation comme le montre l'exemple suivant tiré des travaux de J.P. HATON et J.M. PIERREL (2), (3), (27).

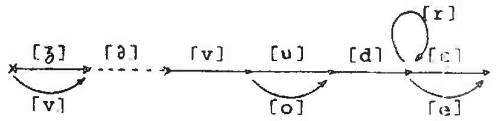


figure II-1

représentation lexicale de : je voudrais

L'élision du [ə] est représentée par une flèche en pointillé.

Une autre façon de définir la forme de référence est d'utiliser une forme standard et d'appliquer les règles phonologiques pour obtenir toutes les variantes de prononciation.

.....

à chaque noeud. La comparaison est un processus markovien qui recherche le chemin le plus probable correspondant à une suite de phonèmes observés dans la chaîne d'entrée.

L'algorithme de Viterbi (8) apporte une solution à ce problème.

Cette méthode est en fait la solution par programmation dynamique du problème. Elle est très utilisée en reconnaissance globale de mots ou en recherche de mots clés.

La suite d'observations est une suite d'échantillons (valeur du signal numérisé ou n-uplet de paramètres). Lorsque la forme de référence est une suite de n échantillons, les points du réseau sont ceux d'un carré (n,n) voisins de la diagonale (figure II. 2) et la probabilité calculée au point (i,j) est l'inverse d'une distance entre le ième échantillon de la forme et le jème échantillon du signal.

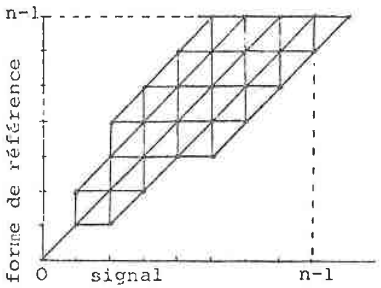


figure II.2

Exemple de réseau stochastique utilisé en programmation dynamique

β) Le problème de l'extraction de sous chaîne

formalisation du problème

Il s'agit de reconnaître à l'intérieur d'une chaîne donnée une sous-chaîne comparable à une chaîne de référence. Cette correspondance peut être représentée par un chemin dans le produit cartésien (forme, chaîne d'entrée) comme le montre l'exemple suivant.

.....

La comparaison peut se faire suivant la méthode "des îlots de confiance" (4, 18) dans laquelle on commence par aligner les phonèmes des deux formes dont les caractéristiques sont identifiables et discriminantes ; les fricatives ou les voyelles accentuées sont de bons candidats. Cette normalisation en longueur permet de s'affranchir de beaucoup de phénomènes d'élision et d'insertion.

Après cette normalisation, un degré de ressemblance est évalué entre ces deux mots. Ce degré est calculé à partir de la ressemblance des phonèmes correspondants dans les deux formes.

Une autre méthode de comparaison a été développée par J.M. PIERREL (3), plus simple, elle est néanmoins efficace. La comparaison se fait de gauche à droite sans normalisation, en cas d'échec le reconnaisseur peut émettre, si le taux de reconnaissance de la partie gauche est suffisament élevé, des hypothèses d'élision, d'insertion ou de substitution de phonèmes et valider ces hypothèses en se resynchronisant plus loin dans les deux formes. Cette méthode est assez semblable à celle développée par RABINOWITZ utilisant l'algorithme de FANO (16). Ces deux méthodes ne s'appliquent qu'après localisation du début d'un mot.

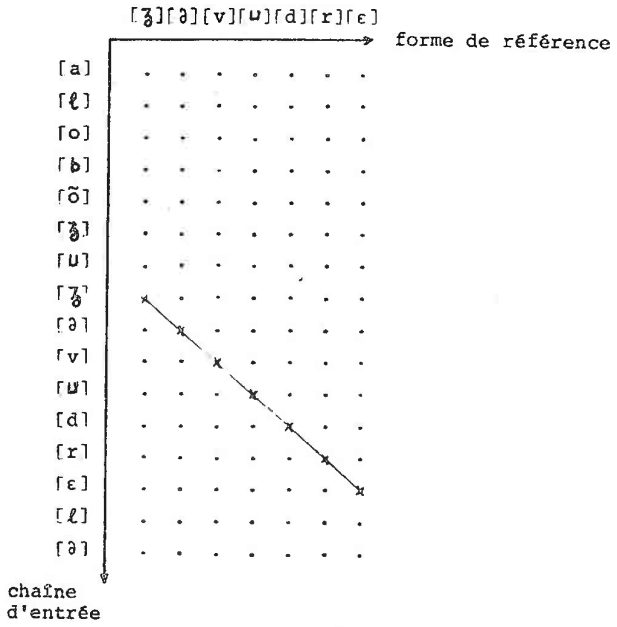
b) L'analyse par synthèse

D'après Reddy (1), cette méthode est intéressante pour son pouvoir discriminant. Elle a été développée par KLATT aux U.S.A. (9). Il s'agit de créer l'ensemble des paramètres acoustiques d'un mot à partir d'une représentation d'un modèle du conduit vocal et d'un ensemble de règles phonologiques puis de comparer cet ensemble avec celui correspondant à une partie du signal d'entrée. D'après KLATT, il est plus facile d'appréhender les traits d'un signal donné en cherchant à synthétiser puis vérifier des hypothèses plutôt que de les retrouver d'une manière analytique.

c) L'introduction d'un modèle stochastique

Ce type de comparaison est utilisé dans le système IBM (5) et dans le système DRAGON (6). Les différentes variantes de prononciation sont représentées par un réseau dans lequel une probabilité est associée à chaque transition et une autre calculée

.....



reconnaissance de je voudrais à l'intérieur de la phrase : allo bonjour je voudrais le ...

Ce produit cartésien peut être muni d'une relation d'ordre partiel notée $<$

On dira que : $(i_1, j_1) < (i_2, j_2) \Leftrightarrow i_1 < i_2$ et $j_1 < j_2$

Cette relation induit un ordre total sur tous les sous-ensembles de points constituant des chemins de pente égale à un.

La construction d'un tel chemin revient à rechercher à l'intérieur d'un ensemble partiellement ordonné (ie : le produit cartésien) une chaîne totalement ordonnée d'origine et d'extrémité égales à $(1,k)$, $(\ell, k+\ell)$.

Si on accepte une chaîne d'entrée comportant des élisions, des insertions ou des répétitions de phonèmes, le chemin peut avoir des transitions parallèles aux axes ce qui revient à rechercher des chaînes dans lesquelles certains points successifs ne sont plus qu'ordonnés partiellement. En chaque point de l'en-

...

semble est défini un poids et les chaînes recherchées doivent avoir un poids total maximal.

Formalisé de la sorte, ce problème est aussi abordé en mathématiques et génétique moléculaire par David SANKOFF (23, 24) mais il semble que les extrémités des chaînes soient approximativement connues ce qui n'est pas le cas en reconnaissance de la parole.

Les différentes approches

Le problème de la localisation de mots en reconnaissance de la parole continue a reçu moins d'attention que le problème de la reconnaissance de mots isolés. On peut toutefois dégager trois approches

. une approche heuristique

Il s'agit de prolonger un chemin en examinant le poids des différents points du réseau dans les trois directions possibles. Deux auteurs : ELLIS (25) et WARREN (26) indiquent qu'une telle méthode peut fournir de bons résultats mais qu'un mauvais choix en un point a des conséquences désastreuses même avec deux formes similaires.

Une autre heuristique est utilisée par R. VIVES (17). Un chemin n'est plus construit étape par étape mais on construit la relation d'égalité entre phonèmes à l'intérieur du produit cartésien. Cette relation fournit une matrice de coïncidence qui aide à déterminer toutes les sous-chaînes communes. Dans une seconde étape les frontières de mots sont approximativement supposées à l'aide des différentes sous-chaînes extraites et un indice de similitude est calculé (18).

Nous citerons en dernier lieu la méthode utilisée à Toulouse par l'équipe de G. PERENNOU (19, 20). Dans cette méthode, la notion de chemin disparaît complètement. Les formes d'entrée et de référence sont des suites de syllabes. La forme d'entrée est parcourue séquentiellement de la gauche vers la droite et à chaque étape deux actions sont accomplies

- Par l'intermédiaire d'une fonction d'adressage sur les paires de syllabe (Hash-code en franglais) on accède dans le lexique à la liste des mots débutant par ces deux premières syllabes.

.....

Les plus longs mots comportant en moyenne cinq syllabes, il ne reste que 3 comparaisons à faire au plus.

- On poursuit la comparaison des transcriptions syllabiques des mots ainsi trouvés avec la portion de forme d'entrée déjà lue.

Le lexique comporte des renseignements qui permettent de tenir compte des phénomènes d'élision et de liaisons entre phonèmes.

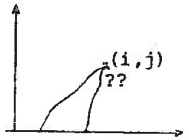
· Approche par programmation dynamique

En 1975 SAKOE et CHIBA (21) effectuaient une reconnaissance de chiffres prononcée à l'intérieur d'un flot continu de parole à l'aide d'un algorithme basé sur ce principe Citons aussi (20) et (27).

Dans cette méthode, on suppose que le meilleur chemin pour arriver au point (i,j) est indépendant de ce qui se passe au delà de (i,j). On associe à chaque chemin P(i,j) s'achevant en (i,j) un score F(P(i,j)) et le meilleur chemin jusqu'à (i,j) noté $\hat{P}(i,j)$ est celui qui maximise les différents scores

$$\hat{F}(i,j) = F(\hat{P}(i,j)) = \max_{P} \left\{ F(P(i,j)) \right\}$$

On définit F à chaque étape en prolongeant dans différentes directions le meilleur chemin arrivant à un point en cumulant des poids des différents points. Mais il n'est pas facile de reconnaître la première partie d'un chemin optimum sans considérer le reste de celui-ci, en effet on peut avoir deux chemins s'achevant en (i,j), l'un court et avec un score bas, l'autre plus long et avec un score élevé. Pour éviter ce problème, il



faudrait effectuer une normalisation en longueur, c'est-à-dire avoir repéré des traits caractéristiques à l'intérieur des transcriptions pouvant permettre un alignement des formes.

Une troisième méthode qui a été un point de départ à notre travail consiste à définir une similitude au voisinage de chaque point, elle a été développée par J.S. BRIDLE (13) puis reprise par R.W.CHRISTIANSEN et C.K.RUSHFORTH (15)

Notre approche diffère sur la définition du réseau. Alors que tous les auteurs cités utilisaient des chaînes linéaires de paramètres (sorties de vocodeurs (13) ou coefficients d'autocorrélation (15)), nous l'avons généralisée à des treillis de phonèmes. De plus la forme de référence tient compte des altérations phonologiques les plus courantes.

IV. LA METHODE RETENUE

A) Idées directrices

Nous avons voulu réaliser un algorithme de localisation et de reconnaissance de mot (en anglais "word spotting") qui travaille sur une transcription phonétique d'un phrase prononcée sans pauses artificielles entre les mots. Cette transcription phonétique est le treillis de phonèmes fourni par le niveau acoustique du système. De cette façon, nous nous différencions des algorithmes développés par (13) et (15).

Les formes de référence sont aussi des treillis de phonèmes qui permettent de tenir compte des altérations phonologiques les plus courantes et des erreurs introduites par le système acoustique. Nous nous différencions ainsi des travaux de R. VIVES et J.Y. GRESSER.

Cet algorithme était destiné au départ à localiser et identifier un mot à l'intérieur du treillis d'entrée qui sert de début à une analyse syntaxique. Nous l'utilisons aussi comme vérificateur d'hypothèses émises par les niveaux supérieurs du système.

Nous avons voulu aussi utiliser la même structure de lexique que celle définie par J.M. PIERREL dans le système MYRTILLE I.

B) Description du dictionnaire

Celui-ci est identique au dictionnaire défini en (2) et (3). Sa description peut se faire en deux parties :

- description des mots du vocabulaire
- description des accès aux formes de référence

B-1) Description des mots du lexique

Un mot du lexique est un graphe dans lequel les arcs sont valués par des phonèmes (voir figure II.1).

Le langage de programmation utilisé, à savoir FORTRAN, nous a obligé à représenter ce graphe par un tableau, TER (LMØT,3)

1	[ʒ]	[v]	
	[-ə]	[a]	
	[v]	[z]	
	[ɯ]	[o]	
	[d]	[b]	
LMØT	[ɛ]	[e]	[r]

Figure II.3
Représentation du mot je voudrais

TER(I,1) représente le phonème correspondant à une prononciation exacte. Si TER(I,1) est négatif cela signifie une possibilité d'élision.
TER(I,2) représente le phonème qui peut se substituer à TER(I,1)
TER(I,3) représente le phonème qui peut venir s'insérer avant TER(I,1)

B-2) Description des accès

Tous les mots du lexique sont fusionnés dans un seul tableau TER(LT,3) et l'accès à un mot du lexique se fait par l'intermédiaire de DEBUT(NBMØT).
DEBUT(N) donne l'indice de début dans TER du mot représenté par l'entier N.
LMØT(N) est la taille du sous tableau représentant N.

.....

B-3) Remarques

Cette représentation appelle deux types de critiques :

- elle ne permet pas de rendre compte des altérations provoquées par la jonction de deux mots (phénomènes de liaison, élision, coarticulation)

- elle oblige la création d'autant de formes que de tournures pour un mot donné car elle ne tient pas compte des radicaux communs, par exemple il faudrait représenter il chante et ils chanteront comme deux formes différentes. Ceci n'est pas gênant quand le nombre de mots est faible mais le devient dès que la taille du vocabulaire augmente et que le langage admet un plus grand nombre de tournures de phrases. C'est pourquoi de nombreux chercheurs (10, 12) tentent de définir des représentations dans lesquelles la forme de référence est construite dynamiquement à partir d'un radical stocké en mémoire et de procédures rendant compte des altérations phonologiques suivant l'entourage phonémique. Nous travaillons actuellement à la réalisation d'un niveau morphologique de cette nouvelle génération qui permettra d'obtenir les mots lexicaux sous la forme d'un graphe (figure II.1), ainsi les algorithmes de comparaison déjà développés seront toujours utilisables.

C) Recherche d'un mot à l'intérieur d'une chaine de phonèmes

C.0) Données et résultats d'un algorithme de recherche

Données :

- La chaîne d'entrée est une suite de triplets de phonèmes. Un triplet représente les trois réponses les plus probables du niveau acoustique. Ce treillis ou ce graphe est représenté par un tableau GRAPH(LG,3). Nous supposons que les trois phonèmes ont la même durée ; plus généralement, si ces durées sont différentes, on a une chaîne d'entrée qui a une structure de treillis.
- La forme de référence sera représentée par une partie du tableau TER. Dans la suite nous supposerons qu'elle occupe les mots d'indice variant de 1 à LMØT.

.....

- Nous utilisons une matrice booléenne de confusion de phonèmes : MATCØN(PH,PHF) = vrai ⇔ PH et PHF peuvent se substituer l'un à l'autre.
  Cette matrice a été établie à partir de la matrice de confusion donnée au chapitre 1

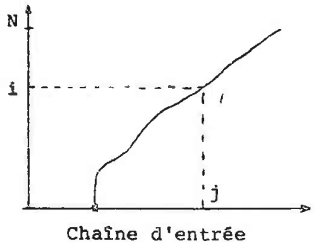
Résultats :

Un tableau RESULT(.,.) contiendra les résultats de la recherche.

RESULT(n,1) = numéro codant le mot recherché
RESULT(n,2) = moyenne arythmétique des ressemblances calculées le long du chemin
RESULT(n,3) } origine et extrémité dans le graphe d'entrée de
RESULT(n,4) } la réalisation du mot

C-1) Recherche des chemins optimaux sur le principe de similitude locale

Pour définir ce principe nous supposerons dans un premier temps que les formes sont des chaînes linéaires



Soit V une fonction de poids sur le réseau. Cette fonction rend compte de la ressemblance entre les paires de phonèmes. V(i,j) vaut 1 quand le phonème pointé par i dans la forme de référence est égal au phonème pointé par j dans la chaîne d'entrée. L'algorithme représentant cette fonction sera donné dans un paragraphe suivant.

Nous noterons par P(i,j) un chemin partiel s'achevant en (i,j) et par S(i,j) la similitude locale de ce chemin au voisinage de (i,j). On a toujours la relation :

$$0 \leq S(i,j) \leq 1$$

Parler de similitude d'un chemin au voisinage d'un point est un abus de langage, il vaudrait mieux parler de similitude entre le voisinage du phonème i de la forme de référence et le voisinage du phonème j de la chaîne d'entrée, mais comme un chemin représente l'ensemble des paires de phonèmes mis en correspondance, on lui attachera la similitude totale entre les paires qui le constituent.

La similitude locale a été définie par J.S. BRIDDLE de la façon suivante :

$$\begin{cases} S(1,j) = V(1,j) \\ S(i+a,\ j+b) = (1-\gamma)\ S(i,j) + \gamma\ V(i+a,\ j+b)\quad K(a,b) \end{cases}$$

où γ est un paramètre constant de l'algorithme (a,b) un couple de la forme (0,1), (1,0), (1,1) qui représente une progression élémentaire du chemin et K un facteur qui permet de pénaliser la similitude quand la progression se fait parallèlement aux axes. K(a,b) vérifie les relations :

$$K(1,1) = 1$$
$$0 < K(0,1) < 1$$
$$0 < K(1,0) < 1$$

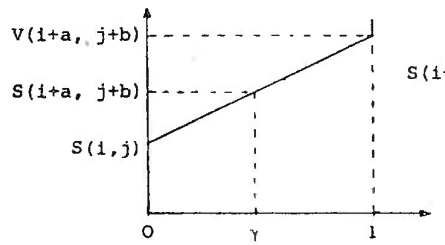
Exemple :

Posons $\gamma = 0.4$ et $K(a,b) = 1\ \forall (a,b)$
Soient V_1, V_2, V_3, V_4 les ressemblances successives calculées au cours des quatre premières étapes. Nous avons :

$$S_1 = V_1$$
$$S_2 = 0.6\ V_1 + 0.4\ V_2$$
$$S_3 = 0.6\ S_2 + 0.4\ V_3 = 0.36\ V_1 + 0.24\ V_2 + 0.4\ V_3$$
$$S_4 = 0.6\ S_3 + 0.4\ V_4 = 0.216\ V_1 + 0.144\ V_2 + 0.240\ V_3 + 0.400\ V_4$$

Nous voyons que dans le calcul de S_4 il y a prépondérance des ressemblances des derniers phonèmes comparés, c'est ce qui nous fait dire que ce calcul heuristique donne une idée de la similitude locale du chemin.

Interprétation géométrique



$$S(i+a, j+b) = (1-\gamma)\, S(i,j) + \gamma\, V(i+a, j+b)$$

La similitude locale au voisinage d'un point (i,j) du réseau rend compte de la correspondance entre phonèmes dans les trois dernières paires constituées lors des trois dernières étapes de prolongement du chemin. Toutes les altérations phonologiques (voir § C-3) influent sur la constitution de trois paires successives ainsi en comparant globalement les trois derniers phonèmes mis en correspondance dans la forme de référence et le signal, on peut s'affranchir de beaucoup d'altérations non prévues dans le lexique.

La figure donnée dans le § "interprétation géométrique" permet d'illustrer les variations du calcul de S(i+a, j+b) en fonction de γ quand V(i+a, j+b) et S(i,j) sont donnés.

Si en un point, la similitude locale d'un chemin tombe en dessous d'un seuil fixé, le chemin est abandonné car la correspondance entre les derniers phonèmes des chaînes n'est plus vraisemblable. Si par contre un chemin atteint un point de coordonnées (LMØT,J), ce qui signifie que l'on a réussi à mettre en correspondance la forme de référence jusqu'au dernier phonème avec une sous-chaîne d'entrée s'achevant sur le phonème d'indice J, alors une occurrence de ce mot a été détectée. L'origine de cette occurrence peut être obtenue comme la dernière valeur de la suite des origines des chemins partiels composant ce chemin.

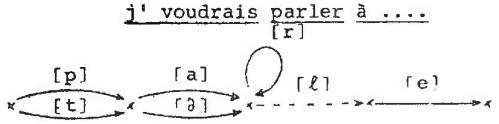
Lorsque les formes sont constituées de suites d'échantillons de mesures (au minimum 20 échantillons de plusieurs mesures dans (15)), la fonction K suffit pour permettre une correspondance globale. Les occurrences détectées dans le signal d'entrée ont à peu près la même longueur que la forme de référence. Mais, dans

.....

notre application où les formes sont des treillis de phonèmes (un mot a en moyenne 5 phonèmes), nous n'avons pas pu trouver une fonction K qui permette d'empêcher de mettre en correspondance des formes de longueur trop différente comme le montre l'étude de l'exemple suivant.

C-2) Les limites de la notion de similitude locale

Il s'agit de reconnaître le mot parler à l'intérieur de la phrase :



représentation lexicale de parler

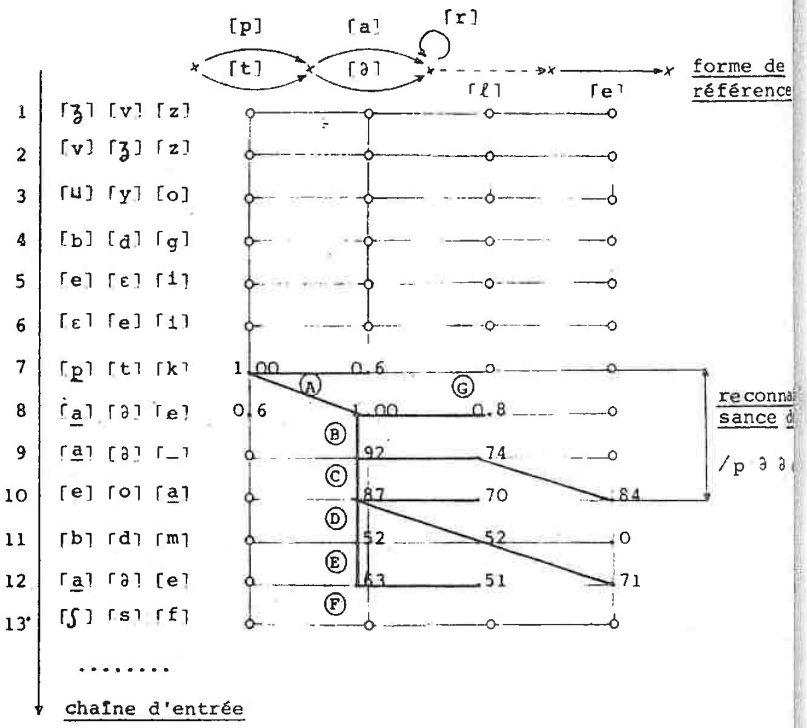
La transition en pointillé marquée par [ℓ] indique une élision possible de ce phonème.

La transition formant une boucle marquée par [r] indique une insertion possible [r].

Ces deux altérations sont dues aux limites du système acoustique qui détecte très mal ces deux phonèmes.

Nous avons après exécution de l'algorithme construit les chemins suivants :

.....



Les valeurs associées aux points du réseau sont les similitudes des chemins en ces points.
Nous avions fixé :

$\gamma = 0.4$

$K(0,1) = 0.8$

$K(1,0) = 0.5$

$K(1,1) = 1.$

Les transitions (B,C), (C,D) rendent compte de la répétition du [a] dans la chaîne d'entrée. Les transitions (D,E) (E,F) rendent compte d'une insertion non prévue du triplet ([b], [d], [m]) entre deux [a], ceci est possible car la similitude locale en D est suffisamment élevée. En effet, les valeurs de γ et du seuil d'abandon permettent au chemin d'atteindre le point F même si la ressemblance entre le triplet ([b], [d], [m

de la chaîne et le couple ([ɑ], [ə]) du treillis de la forme est nulle.

Pourtant la portion de chemin s'achevant en F ne représente pas une correspondance acceptable entre forme de référence et sous-chaîne. Pour éliminer un tel phénomène nous pouvions diminuer la valeur de K(0,1) mais de cette façon nous pénalisons de la même façon la similitude lors des transitions (B,C) et (C,D), ce qui n'est pas logique car le chemin s'éloigne de plus en plus de sa trajectoire idéale, à savoir une droite de pente égale à un. D'autre part la valeur de K(1,0) dépend des performances du système acoustique. Si celui-ci introduit fréquemment des répétitions dues à une mauvaise segmentation il faudra que K(1,0) soit assez élevé.

En conclusion, la similitude locale est insuffisante pour apprécier la correspondance globale entre deux formes. Par contre elle permet une comparaison élastique entre deux sous-chaînes de faible longueur (trois à quatre phonèmes suivant la valeur de γ). Cette propriété servira à s'affranchir des problèmes causés par les altérations phonologiques à l'intérieur d'un mot, comme nous le verrons au paragraphe suivant. C'est pour avoir une idée de la correspondance globale entre une sous-chaîne extraite du treillis et un début de forme de référence que nous introduisons une contrainte de pente sur chaque chemin partiel. Après chaque transition parallèle aux axes, la pente du chemin est évaluée et celui-ci abandonné dès que la pente diffère trop de un. Physiquement celà signifie que les formes comparées ont presque le même nombre de phonèmes, c'est-à-dire la même durée. Ceci n'est pas génant car on peut faire l'hypothèse d'une vitesse d'élocution sensiblement constante dans la prononciation d'un mot.

C-3) Relations entre transitions et altérations phonologiques

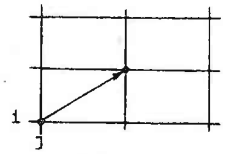
Dans ce paragraphe, nous représenterons verticalement la forme de référence par un treillis (représentation définie dans le paragraphe B) et horizontalement la chaîne d'entrée par la suite des triplets de phonèmes représentant les trois meilleurs réponses des algorithmes de classification.

.....

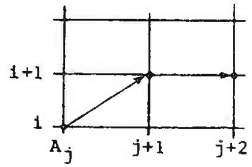
Nous allons préciser comment se traduit, en termes de transitions dans le réseau, chaque altération phonologique. Nous distinguons deux types de transitions, celles qui sont en accord avec la représentation lexicale de la forme de référence et celles qui correspondent à des altérations phonologiques non prévues dans le lexique.

Nous noterons par Ph(j) un élément du triplet d'indice J dans le pseudo-graphe d'entrés et par Phf(i) un élément de TER(I,*)

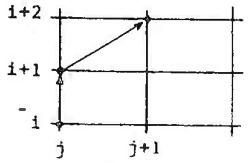
Cas des altérations prévues



cas idéal : égalité entre Ph(j+1) et |TER(I,1)| ou TER(I,2)

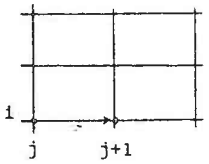


insertion prévue de Ph(j+1) avant Ph(j+2)
égalité de Ph(j+1) et de TER(I,3)
égalité de Ph(j+2) et de |TER(I,1)| ou TER(I,2)



élision prévue de TER(I+1,1)
il y a correspondance entre Phf(i) et Ph(j) d'une part et Phf(i+2) et Ph(j+1) d'autre part.

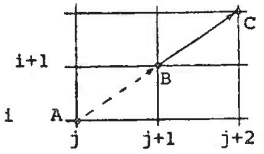
.....



répétition de Phf(i)

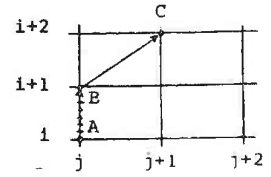
Cas des altérations non prévues

Nous représenterons par ----→ une transition dont l'origine et l'extrémité ont des similitudes supérieures au seuil d'abandon et où la ressemblance entre les phonèmes mis en correspondance à l'extrémité est nulle.
C est le point de resynchronisation où la ressemblance est suffisamment élevée.



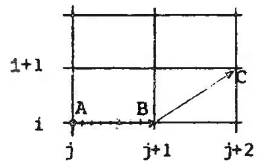
Substitution de TER(I+1,1) par Ph(j+1)
égalité de Phf(i+2) et Pf(j+2)

Pour accomplir la transition AB il faut qu'il y ait confusion entre Phf(i+1) et Ph(j+1) donc il faut :
matcon (Ph(j+1), Phf(i+1)) = vrai



élision de TER(I+1,1)

.....



insertion de Ph(j+1) avant Phf(i+1)
égalité de Phf(i+1) et Ph(j+2)

C-4) La fonction de ressemblance entre phonèmes

Cette fonction évalue la meilleure ressemblance, à un instant donné, entre les phonèmes GRAPH(J,*) et TER(I,*). Cette ressemblance calculée en chaque point du réseau permet le prolongement ou l'abandon des chemins en accord avec les altérations phonologiques prises en compte. Les altérations phonologiques permises sont codées par les valeurs de TER(I,*). Les données de cette fonction sont :

. Les paires de phonèmes créés à partir de TER(I,*) et GRAPH(J,*)
. Le type de transition que l'on cherche à valider
. La similitude du chemin que l'on cherche à prolonger
. Une fonction de confusion entre phonèmes pour traiter le cas de la substitution de deux phonèmes.

Algorithme

1er cas : (il y a égalité entre un élément de TER(I,*) et un élément de GRAPH(J,*)) RESSEMBLANCE = 1.

2ème cas : (il n'y a pas d'égalité,
TER(I,*) indique une élision,
on essaie de valider une transition parallèle à l'axe des I)

/* Il s'agit donc de prendre en compte une élision dans la chaîne d'entrée. Dans ce cas nous préférons que la similitude du chemin n'augmente pas car il n'y a pas plus de phonèmes mis en correspondance.
Nous avons la relation :

$$S_{n+1} = (1 - \gamma)\ S_n + K(1,0)\ \gamma\ V_{n+1} = S_n$$

$$S_n(1 - \gamma - 1) = - K(1,0)\ \gamma\ V_{n+1}$$

$$S_n = K(1,0)\ V_{n+1}$$

Comme K(1,0) est toujours inférieur à un, nous poserons : */

RESSEMBLANCE = S_n

3ème cas : (il n'y a pas d'égalité,
il n'y a pas d'élision
on essaie de valider une transition diagonale)

/* Nous sommes en présence d'une substitution dans la chaîne d'entrée. Pour tester la plausibilité de cette substitution nous utilisons une matrice booléenne de confusion de phonèmes */

si (MATCØNF(GRAPH(J,*), TER(I,1)) alors RESSEMBLANCE = 0
sinon RESSEMBLANCE = -∞ fsi

Remarque :

Nous avons pensé, dans un premier temps, comparer les spectres fréquentiels des portions de signal et de forme mis en correspondance, mais la place mémoire demandée était trop importante, il faut cent fois plus de place pour stocker le spectre d'une forme que pour stocker la représentation du treillis.

Une autre méthode envisagée est d'associer par apprentissage une énergie à chaque phonème pour la comparer à l'énergie du phonème traité. Il est facile de trouver par heuristique une ressemblance (28).

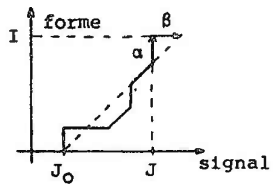
4ème cas : (dans tous les autres cas)
RESSEMBLANCE = 0.

C-5) Exemples

Nous avons fixé par expérimentation sur un ensemble de 13 phrases comportant 6 mots en moyenne les valeurs des paramètres de la façon suivante :

$\gamma = 0.4$
$K(1,0) = 0.5$ = taux de pénalisation en cas d'élision
$K(0,1) = 0.8$ = taux de pénalisation en cas de répétition ou d'insertion
seuil d'abandon = 0.5

Si ORIGINE (I,J) est l'origine du chemin s'achevant en (i,j) (quand il n'y a pas de chemin, origine (i,j) est nul) alors la contrainte de pente s'écrit :



ORIGINE(I,J) = J_O

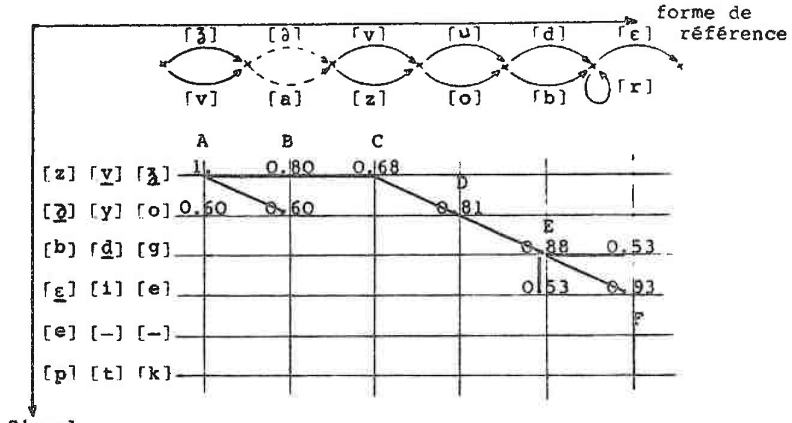
$\alpha = |I - ORIGINE(I,J)| \leq m$

$\beta = |J - ORIGINE(I,J)| \leq n$

m = si LMØT ≤ 5 alors 1 sinon 2

n = si LMØT ≤ 5 alors 2 sinon 3

Exemple 1



reconnaissance de je voudrais dans
j' voudrais poste ....

commentaire

La transition AB est due à l'élision du groupe [ə], [a], dans ce cas la vraisemblance en B est égale à la similitude en A. nous avons donc :

$S_B = 0.6 \times S_A + 0.5 \times 1. \times 0.4 = 0.8$

.....

La transition BC n'est possible que parce que la contrainte de pente n'a pas encore été présente car la longueur du mot est supérieure à 5 phonèmes. Nous avons :

vraisemblance en C = 1.

$S_C = 0.6 \times S_B + 0.5 \times 0.4 \times 1. = 0.6 \times 0.8 + 0.2 = 0.68$

On obtient successivement :

$S_D = 0.6 \times 0.68 + 0.4 \times 1. = 0.81$

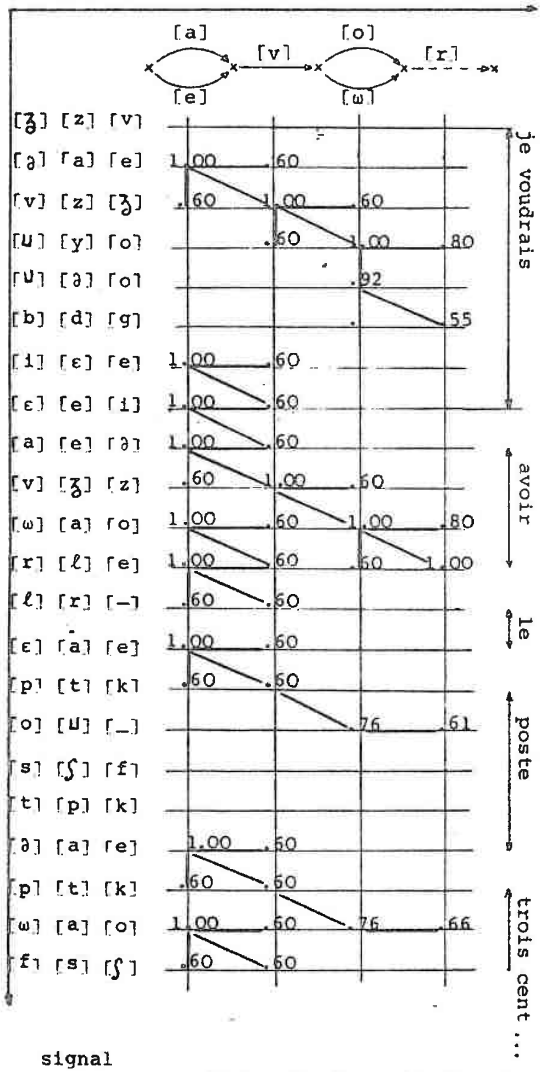
$S_E = 0.6 \times 0.81 + 0.4 \times 1. = 0.88$

Deux chemins s'achèvent en (LMØT,J). Celui qui possède une similitude locale de 0.53 en ce point correspond à une hypothèse d'élision du [ɛ] car en E la similitude est assez forte pour prolonger le chemin parallèlement aux axes contrairement à ce qui se passe en D.

Exemple 2 :

Cet exemple illustre bien un problème rencontré en reconnaissance de la parole. Le mot recherché apparaît à plusieurs endroits dans la chaîne même s'il n'a pas été prononcé.

....



Recherche de avoir dans :
je voudrais avoir le poste 340

Remarque :

Au point de coordonnée i = 3, j = 5 la contrainte de pente empêche de poursuivre une transition verticale bien que la similitude élevée l'autorise.

C-6) L'algorithme de construction des chemins

Le principe de cet algorithme est très simple. Il doit créer un début de chemin en regardant si chaque phonème du pseudo-graphe d'entrée est un début de forme et chercher à prolonger chaque chemin partiel par des transitions diagonales ou parallèles aux axes. Il n'est pas tenu compte dans le prolongement d'un chemin des différents schémas d'enchaînement correspondant aux altérations phonologiques. C'est la procédure RESSEMBLANCE qui pénalise les constructions incorrectes.

Cet algorithme est décrit précisément en (13) et (15). Nous allons donner une idée de son fonctionnement en décrivant un algorithme abstrait, pour plus de détails on pourra se reporter en (13) et (15).

On explorera la chaîne d'entrée de DEBUT jusqu'à FIN.
L'indice J pointe dans le pseudo-graphe d'entrée
L'indice I pointe dans le treillis de la forme de référence.
LMØT repère la fin du treillis.

1ère étape : Recherche des chemins débutant en GRAPH(DEBUT,*) parallèles à l'axe des I en accord avec les contraintes de pente.

2ème étape : Pour J de DEBUT + 1 à FIN répéter

- Prolongement en accord avec les contraintes de pente des chemins partiels s'achevant aux points de coordonnées (I,J-1), 1 < I < LMØT par des transitions horizontales et diagonales.
- Recherche des chemins débutant sur GRAPH(J,*) et parallèles à l'axe des I et prolongements des chemins s'achevant en (I,J), 1 < I < LMØT par des transitions parallèles à l'axe des I en accord avec les contraintes de pente.
- Recherche des chemins s'achevant en (LMØT,J)

Conclusion :

Pour prolonger un chemin partiel, il nous suffit de connaître :

. sa similitude partielle avant le prolongement
. son origine, afin de connaître les limites de chaque occurrence dans la chaîne d'entrée
. la ressemblance en sa nouvelle extrémité.

Dans la construction n'interviennent pas les schémas d'enchaînement des transitions relatifs aux différentes altérations phonologiques. Mais comme la fonction ressemblance est paramétrée par le type de la transition que l'on cherche à valider, nous avons quand même en chaque point du réseau une idée de la forme du chemin. Les contraintes de pente s'intéressent à la pente globale du chemin et permettent l'élimination de ceux qui correspondent à des sous-formes qui sont de longueur trop différente.

L'originalité de notre version comparée à l'algorithme de J.S. BRIDLE vient de l'utilisation de la fonction de ressemblance entre phonèmes et des contraintes de pentes apportées aux chemins. Ceci constitue une adaptation au cas de l'extraction d'une sous-chaîne à l'intérieur d'un treillis de phonèmes. Un indice de ressemblance à une forme de référence est associée à cette sous-chaîne ce qui ressemble aux travaux de R. VIVES (18).

C-7) Détermination des paramètres

Cette détermination est importante, car c'est par le biais de la fonction K que l'on peut éliminer localement les constructions qui ne correspondent pas à une altération phonologique.

Détermination de γ

Chez les deux auteurs déjà cités (13, 15), nous avons trouvé γ fixé à une valeur de 0.3 mais la recherche s'effectuait en comparant les formes plus longues. Nous avons vu que la rigueur de la correspondance variait comme γ, lorsque le seuil d'abandon

restait constant. Comme les formes que nous utilisions étaient courtes, il nous fallait être plus rigoureux dans la correspondance, donc nous avons fixé γ à 0.4. Des résultats expérimentaux ont montré que cette valeur convenait.

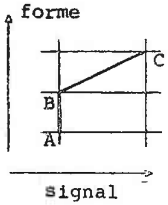
Détermination de K(0,1)

Ce facteur pénalise les répétitions donc dépend fortement des résultats du niveau acoustique celui-ci n'étant pas encore opérationnel. Nous l'avons fixé arbitrairement à 0.8. Il intervient aussi lors d'une insertion mais son influence peut être combattue par un accroissement de la ressemblance chaque fois que TER(I,3) est comparé.

Détermination de K(1,0)

Ce facteur intervient lors d'une élision ou d'une fusion phonétique (exemple 1). A la suite d'une transition le long de l'axe des I il faut que la transition suivante, qu'elle soit du même type ou diagonale, soit validée (ie ; la ressemblance à l'extrémité est suffisamment forte) sinon le chemin doit être abandonné. Celà correspond à l'impossibilité d'avoir deux élisions successives et à la nécéssité de se resynchroniser sur le couple de phonèmes suivants.
K(1,0) affaiblit la similitude afin de la faire descendre sous le seuil en-dessous duquel on ne peut plus prolonger des chemins par des transitions non validées.

Nous avons les relations suivantes



$S_B = (1-\gamma)\ S_A + \gamma\ V_B\ K(1,0)$

$V_B = S_A$

Il faut $S_C \leq$ seuil d'abandon dès que $V_C = 0$ même si $S_A = 1$.

Nous avons donc la relation qui détermine K(1,0) :
$((1-\gamma) + \gamma\ K)\ (1-\gamma) \leq$ seuil d'abandon.

Soit en posant : seuil d'abandon = 0.5

$(0.6 + 0.4\ K)\ (0.6) \leq 0.5$

$0.36 + 0.24\ K \leq 0.5$

$0.24\ K \leq 0.14$

$K \leq 0.58$

Nous avons pris $K = 0.5$

Les valeurs adoptées définitivement pour la fonction K sont celles que nous venons de calculer, ce qui valide notre approche en vue de traiter les altérations phonologiques.

Détermination du seuil d'abandon

Cette valeur a été déterminée expérimentalement. Plus le seuil d'abandon est élevé, moins on reconnaît de mots et vice versa. Il s'agit d'un compromis qui dépend du locuteur, du système de paramétrisation, des conditions de prise de son donc de l'application toute entière.

C-8) Performances

Les essais ont été effectués sur l'ordinateur IRIS 80 CII - HB de l'I.U.C.A.L. Nous avons construit les treillis de mots (voir figure II.4) à partir de treillis de phonèmes : ; ant aux treillis de phonèmes, ils ont été construits à l'aide d'une transcription graphème-phonème de phrase tirée du langage de communication avec un standard téléphonique (voir chapitre IV) en tenant compte des erreurs introduites par le niveau acoustique. Le taux de phonèmes erronnés est de 30 %.

La recherche d'un mot à l'intérieur d'un treillis de phonème est proportionnelle à la longueur du treillis et à la longueur du mot. A titre d'exemple, le treillis de mots de la figure II.4 a été construit en 1 seconde. Sachant qu'il y a 30 mots différents recherchés, il faut compter, pour une phrase d'une durée de 4 secondes, un temps moyen de localisation de 23 ms.

Une autre caractéristique d'un algorithme de localisation de mots est dans la taille du treillis de mots construit lors de la recherche de tout le vocabulaire (voir figure II.4).

.....

L'algorithme de localisation idéal mais irréalisable construirait un treillis de mots égal à la suite des mots prononcés par le locuteur. Dans notre réalisation, l'utilisation d'une matrice de confusion entre phonèmes pour empêcher les hypothèses "grossières" de substitution (voir § C.4) a augmenté la qualité de localisation des mots possédant des traits acoustiques caractéristiques. (Par exemple une succession : plosive, voyelle, fricative). Ces mots pourront servir de points d'ancrage dans une reconnaissance par "îlots de confiance".

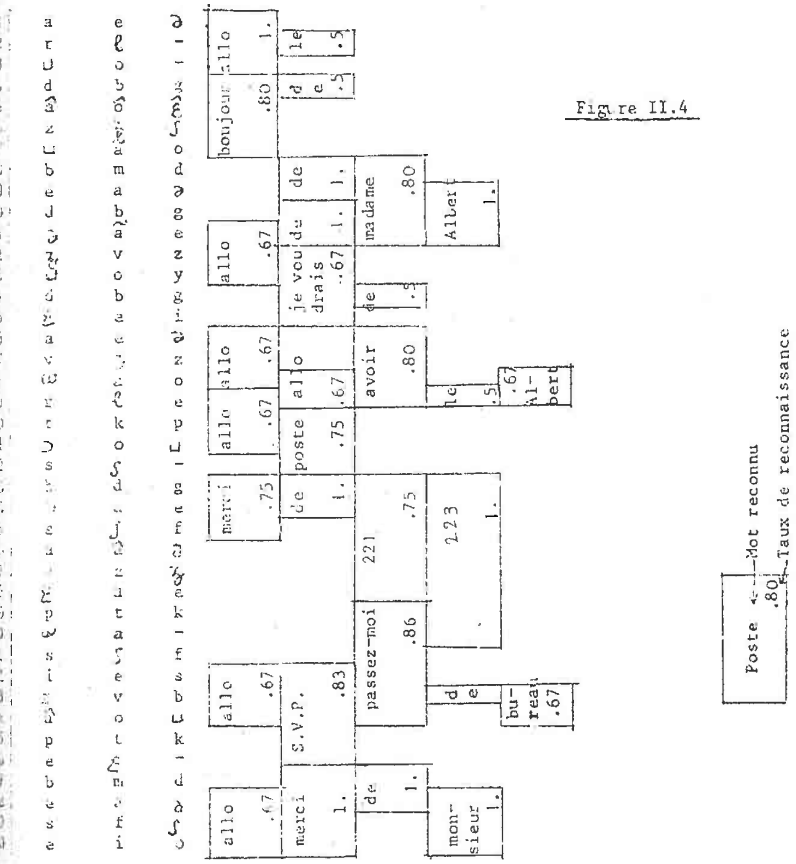


Figure II.4

« Allo bonjour madame je voudrais avoir le poste 223 S.V.P. Merci ».

CHAPITRE III

LE NIVEAU SYNTAXIQUE

A) Rôle du niveau syntaxique

B) Problèmes spécifiques à l'analyse syntaxique d'un message parlé

C) Représentation de la grammaire

D) Méthodes d'analyse syntaxique

E) Aperçu des méthodes utilisées dans les systèmes existants

F) Description d'un analyseur syntaxique procédant à partir d'un ilôt de confiance

-:-:-:-:-:-:-

A) ROLE DU NIVEAU SYNTAXIQUE

Dans le cadre de la reconnaissance analytique du discours continu, il est nécessaire de retrouver les mots à l'intérieur du treillis phonétique. Nous avons vu que le passage : treillis de phonèmes → treillis de mots se fait au niveau lexicographique. Le passage suivant, à savoir treillis de mots → phrases reconnues se fera par la mise en oeuvre de processus utilisant des informations syntaxiques, sémantiques et pragmatiques.

Nous allons maintenant préciser de quelle façon la syntaxe peut contribuer à la reconnaissance.

La syntaxe est liée à la structure du message, régie par la grammaire du langage utilisé. En définissant une grammaire nous nous donnons une description a priori de la structure des phrases à reconnaître, il devient ainsi possible de limiter la recherche au cours de la reconnaissance, en éliminant les tournures incorrectes du point de vue syntaxique. Concrètement, cela signifie que nous effectuerons une analyse syntaxique à partir du treillis de mots pour en dégager une suite constituant une phrase syntaxiquement correcte.

B) PROBLEMES SPECIFIQUES A L'ANALYSE SYNTAXIQUE D'UN MESSAGE PARLE

Initialement, l'analyse syntaxique a été développée en informatique pour guider la compilation des langages de programmation et diverses méthodes d'analyse syntaxique ont été mises au point puis formalisées.

L'analyse syntaxique est aussi utilisée dans la reconnaissance automatique des textes écrits ou des méthodes spécifiques ont été développées (AS 9).

Mais dans ces deux champs d'application, la donnée à analyser est formée d'une suite d'unités non ambiques et distinctes ; ainsi, en reconnaissance de texte, contrairement à ce qui se passe en reconnaissance de la parole, il est par exemple possible de faire la distinction entre les mots lit et lie ou les groupes de mots le maladroit et le mâle a droit.

.....

Pour mieux cerner les problèmes spécifiques de la reconnaissance de la parole, nous allons donner les différences entre deux représentations d'un message, l'une écrite, l'autre parlée.

. Dans une représentation parlée, les mots sont soudés entre eux, alors qu'il existe des espaces entre les mots écrits. Ceci introduit des segmentations multiples de groupe de mots comme nous venons de le voir.

. La présence d'homophones dans le vocabulaire introduit une ambiguité sur l'utilisation de ceux-ci.

. Les mots courts, comme les articles, conjonctions ou prépositions sont largement utilisés dans l'analyse d'un texte écrit car porteurs d'information syntaxique alors que leur reconnaissance acoustique n'est pas suffisamment sûre pour que l'analyse puisse s'appuyer sur eux.

. La syntaxe d'un message parlé est moins stricte que celle d'un texte écrit et souvent des interjections et des mots non significatifs comme : bon, heu et bien..., sont ajoutés par le locuteur.

. Il n'existe pas de ponctuation rigoureuse dans un message parlé. L'intonation, la mélodie, plus généralement, la prosodie donnent beaucoup de renseignements mais son utilisation n'est pas bien formalisée et dépend de beaucoup de paramètres de situation comme la fatigue, l'âge, etc...

. Les algorithmes de segmentation et de classification du niveau accoustique introduisent des erreurs qu'il faut corriger

Nous pouvons illustrer ces différences dans l'exemple suivant :

- Il est possible de lire à haute voix :

"Gal, amant de la Reine, alla, tour magnanime"

et d'entendre :

"Galamment de l'arène à la tour Magne, à Nîmes"

Victor Hugo

Il sera donc nécessaire d'adapter les méthodes d'analyse syntaxique pour effectuer une reconnaissance du discours continu guidée par la syntaxe. Nous décrivons dans la suite une telle méthode d'analyse.

.....

C) REPRESENTATION DE LA SYNTAXE

Les langages utilisés en reconnaissance peuvent en gros être représentés par trois types de grammaires de puissance et de complexité croissantes (29).

1 - Les grammaires d'états finis : la grammaire est représentée par un automate possédant un nombre fini d'états (A S3) et engendre un langage régulier très pauvre en tournures syntaxiques. Ce type de grammaire peut être utilisé dans les systèmes de reconnaissance de la parole assurant un contrôle de processus simple comme la commande d'une machine outil par exemple.

2 - Les grammaires hors-contexte : ces grammaires engendrent les langages à contexte libre. La classe des langages à contexte libre peut contenir des sous-ensembles de langues naturelles quand ceux-ci sont orientés vers une activité précise, par exemple le langage d'interrogation d'un standard téléphonique utilisé dans le système MYRTILLE I (2).

3 - Les grammaires de langages naturels : un gros travail a été fait par les linguistes et les chercheurs travaillant sur les textes écrits pour définir ces grammaires. Malheureusement tout n'est pas applicable au discours parlé comme nous l'avons vu au paragraphe B. Le système MYRTILLE II développé dans notre équipe (30), destiné à comprendre un sous-ensemble assez vaste du français parlé, utilise une grammaire de ce type avec quelques restrictions simplificatrices.

Une représentation courante des grammaires consiste à utiliser un réseau. Woods (A S4) propose un réseau de transitions augmenté (Augmented Transition Network) pour tenir compte des définitions récursives de langage. Les réseaux A.T.N. incorporent à leurs noeuds, des procédures mettant en oeuvre d'autres sources d'information dont l'activation modifie le cours de l'analyse.

.....

Un paramètre intéressant pour mesurer la complexité d'une grammaire est le facteur de branchement moyen.

Ce facteur représente en fait le nombre de mots (au sens d'unités syntaxiques minimales pour la grammaire considérée) qui peuvent apparaître à la suite d'une portion de phrase. Les grammaires de ce type 2 ont un facteur de branchement de quelques dizaines, celui des grammaires de type 3 peut dépasser 1000.

Dans certains réseaux de systèmes de reconnaissance américain (système I.B.M.(S, 31)) par exemple, la grammaire est considérée comme une fonction stochastique d'un processus de Markov. Chaque transition est valuée par une probabilité "a priori" et des procédures implantées aux noeuds calculent des probabilités à partir de la donnée.

L'analyse syntaxique est vue comme un processus markovien et "la meilleure" phrase s'obtient grâce à des algorithmes de programmation dynamique. Le modèle sous-jaccent est bien formalisé, mais il est difficile d'évaluer les probabilités a priori et la taille du réseau croit vite avec celle du vocabulaire pour atteindre des valeurs déraisonnables. Une autre façon de représenter la syntaxe est d'utiliser une matrice d'adjacence. Cette matrice binaire indique la possibilité d'avoir une suite de n mots donnés (pour les matrices à n dimensions). Mais cette méthode ne convient qu'aux langages de petite taille et ne permet pas de tenir compte de tout le contexte de l'analyse dès que l'on a reconnu des suites de mots de longueur supérieure à n.

D) METHODES D'ANALYSE

Le rôle de l'analyse syntaxique est de dégager du treillis de mots qui constitue la donnée, une phrase syntaxiquement correcte. Il peut être utile de construire simultanément une représentation syntaxique, généralement sous forme d'un arbre, de la phrase en vue d'une exploitation par d'autres niveaux.

.....

Rappelons que l'analyse se fera à partir d'une donnée entachée de bruits qu'elle devra autant que possible corriger. Par bruits, nous entendons :

- erreurs de paramétrisation dues au système accoustique
- présence de phonèmes erronés dans la phrase prononcée.

Ceprocessus d'analyse peut fonctionner selon deux principes opposés. Soit ce processus est guidé par les données, soit ce processus est guidé par le contexte.

Processus guidé par les données (ascendant)

C'est le cas lorsque l'analyseur travaille sur un treillis préalablement construit. A chaque étape de l'analyse, l'analyseur sélectionne dans le treillis, à la suite du dernier mot pris en compte, l'unité syntaxique qui permet la poursuite de l'analyse. Le processus sera peu influencé par des discontinuités de treillis dues à la présence de bruits dans la chaîne phonétique. Le principal inconvénient de cette méthode est qu'elle nécessite la création préalable du treillis, œ qui signifie que tout le vocabulaire sera recherché en tout point du treillis phonétique, d'où une grande perte de temps.

L'analyseur du système de reconnaissance KEAL du CNET, parmi d'autres, fonctionne sur ce principe (AS 8).

Processus guidé par le contexte (descendant)

Il s'agit de créer le treillis de mots à chaque étape de l'analyse à partir du contexte déjà reconnu. Le processus d'analyse peut être réalisé par itération du schéma d'hypothèse-test :

1 - Construction d'un ensemble d'hypothèses
2 - Recherche dans le treillis phonémique de chaque hypothèse
3 - Validation et détermination d'un nouveau contexte d'analyse.

Explicitons davantage chacune des étapes.

La première étape se fait en déterminant toutes les unités syntaxiques qui permettent de continuer l'analyse, sachant qu'une suite de mots (constituant le contexte) a été déjà reconnue

.....

La deuxième étape est en fait une itération sur l'appel du module de reconnaissance de mots. Le module tente de reconnaître chaque hypothèse à la suite du dernier mot reconnu. Dans la troisième étape, l'analyse se poursuit avec les hypothèses validées.

De cette façon il est possible de réduire la recherch lexicographique mais l'analyse devient plus sensible aux bruits dans la chaîne phonémique.

En effet, une insertion de phonèmes peut empêcher toute reconnaissance à partir d'un phonème donné alors qu'un analyseur guidé par les données aurait pu ignorer cette zone puisqu'elle n'appartient à aucune transcription phonétique des mots du vocabulaire. Avant de décider s'il y a lieu de tenter une resynchronisation dans le reste de la chaîne, il est nécessaire de s'assurer de la validité du contexte en cours d'analyse en essayant de réécrire différemment les mots déjà reconnus. S'il est possible, à l'aide d'autres contextes d'identifier cette zone phonétique, c'est que l'analyse était ambigue jusqu'à cet endroit, et que le contexte initial était erroné, sinon il est nécessaire de se resynchroniser plus loin dans la chaîne.

Les systèmes MYRTILLE fonctionnent sur ce principe et contiennent une procédure d'émission d'hypothèses activée dès que la reconnaissance échoue en un endroit précis.

Ce principe "hypothèses-tests" est couramment utilisé dans les systèmes de reconnaissance et de compréhension, les systèmes américains HWIM, HEARSEAY, DRAGØN fonctionnant tous sur ce principe.

Tous les algorithmes d'analyse syntaxique développés pour les compilateurs peuvent fonctionner selon ces deux principes. De l'ensemble de ces algorithmes on peut dégager deux grandes familles : la classe des analyseurs descendants, et la classe des analyseurs ascendants.

.....

Ces deux méthodes se différencient dans leur façon de construire l'arbre syntaxique de la phrase. Dans la méthode ascendante, les mots qui constitueront les feuilles de l'arbre (en informatique les arbres ont la tête en bas) sont d'abord regroupés en syntagmes, puis les syntagmes sont regroupés à leur tour jusqu'à obtenir une phrase. Dans la méthode descendante, on part de la racine de l'arbre pour prédire les constructions syntaxiques en accord avec la donnée et obtenir les différents syntagmes.

E) APERCU DES METHODES UTILISEES DANS LES SYSTEMES EXISTANTS

Après avoir posé les problèmes et donné des méthodes susceptibles de les résoudre, nous allons décrire les niveaux syntaxiques de quelques systèmes de reconnaissance ou compréhension. Cette tâche n'est pas aisée car il est difficile de cerner la méthode d'analyse utilisée dans un système, le niveau "syntaxique" mettant souvent en oeuvre des informations d'origines diverses (syntaxiques, sémantiques ou pragmatiques) et les différents niveaux interagissent fortement.

Pour chaque système de reconnaissance ou compréhension décrit nous répondrons aux questions suivantes :

- quels sont les données et les résultats de l'analyseur ?
- comment est représentée la syntaxe ?
- l'analyse est-elle guidée par les données ou le contexte ?
- quelle est la place du niveau syntaxique dans le système ?

Notre but n'est pas de dresser une liste exhaustive des systèmes existants mais de présenter un éventail des solutions adoptées à ce jour.

E.1 - Le système HWIM de BBN (32)

Ce système est construit autour d'un superviseur qui dirige l'activation de niveaux classiques de traitement : acoustique, lexical, sémantique, syntaxique. Une hiérarchie existe entre ces différents niveaux alors que dans le système HEARSAY

.....

que nous allons décrire plus loin, elle est implicite. Le niveau syntaxique est activé entre deux appels au niveau sémantique.

Initialement le superviseur demande la création d'un treillis de mots de plus de trois phonèmes afin d'éviter de retrouver de "petits mots" gênants. La construction de ce treillis fait intervenir les niveaux acoustique, phonologique et lexical.

De ce treillis, le niveau sémantique dégage une suite de mots en fonction de relations existant entre ces mots. Une telle ébauche d'interprétation de la phrase, appelée une théorie, est soumise au niveau syntaxique. Celui-ci vérifie la validité de cette théorie au point de vue de la grammaire du langage, puis tente de la compléter en y adjoignant de nouveaux mots. Le superviseur demande ensuite au niveau sémantique de valider cette nouvelle théorie.

Nous voyons que l'analyse est guidée, tout à tour, soit par les données, soit par le contexte. La syntaxe est représentée sous forme d'un réseau A.T.N. qui incorpore des informations sémantiques et pragmatiques. Les unités syntaxiques minimales ne sont pas seulement des catégories syntaxiques comme noms, adjectifs ou suites verbales mais aussi certains mots. Ainsi n'importe quel nom ne peut pas précéder n'importe quel adjectif ou verbe et le réseau dépend étroitement de l'application retenue.

Quant à l'analyseur dont le rôle est de compléter la théorie, il peut débuter en n'importe quel mot du treillis. Un analyseur du même type a été décrit par MILLER en 1975 (AS 5) il est dommage qu'aucune publication n'est révélée ses performances ou les détails de fonctionnement.

Le système HWIM reconnait des phrases en anglais parlé tirées d'un langage spécifique d'une application. Le vocabulaire contient à peu près 1000 mots.

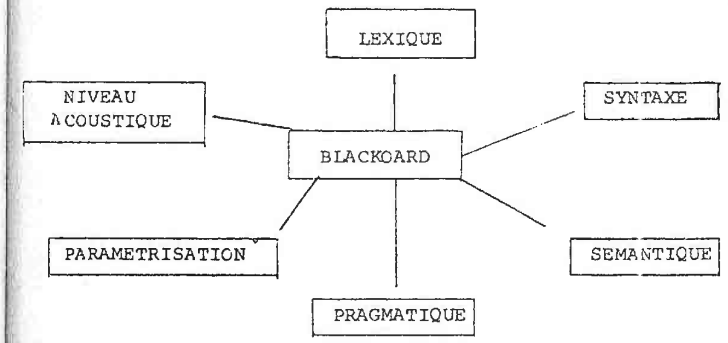
.....

F. 2 - Le système HEARSAY II

Tout comme le précédent, ce système a été développé, dans le cadre du projet "Speech understanding" de l'ARPA de 1971 à 1976. Il comprend des enquêtes posées en anglais à une base de données. Le vocabulaire contient 1011 mots.

Le système fonctionne suivant le principe "hypothèses-tests", tous les processus sont donc guidés par le contexte. Ce schéma est généralisé à tous les niveaux de traitement et s'applique aussi bien à l'intérieur d'un niveau qu'entre deux niveaux.

Pour assurer une souplesse et une efficacité maximale à ce modèle, il est nécessaire que les différents niveaux de traitement (c'est-à-dire les différentes sources d'information, (SI) soient les plus indépendantes possibles. Dans le système HEARSAY II, les différents niveaux coopèrent par le biais d'une base de données commune, le blackboard dont l'élément constitutif est l'ypothèse. L'interaction entre les différents SI est externes à celles-ci.

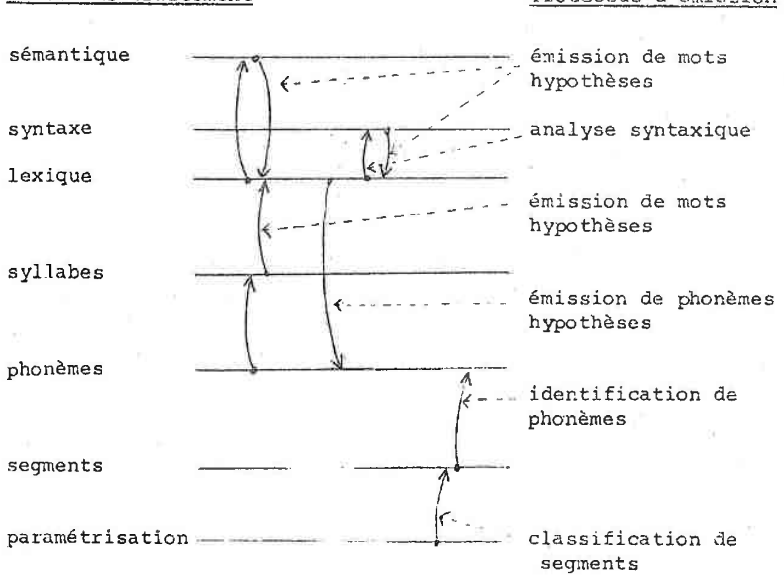


Le modèle HEARSAY II

Chaque SI ignore les autres ; elle ne peut qu'accéder à la base de données pour ajouter, modifier ou supprimer une hypothèse dont la structure est identique quelque soit le niveau d'émission (phonème, mot, phrase).

L'activation d'une S I est dirigée par les données, en ce sens qu'elle dépend d'une précondition qui doit être satisfaite dans le blackboard. Cette précondition partielle est la description d'un état du blackboard. L'action de cette S I fait apparaître d'autres préconditions donc entraîne l'activation d'autres S I. Les auteurs de ce système ont envisagé l'implantation de ce système sur une machine multi-processeur car les processus mettant en oeuvre chaque S I et le processus d'activation des S I sont asynchrones.

Le blackboard est décomposé en niveaux distincts correspondants aux diverses S I disponibles : sémantique, syntaxique, lexical, morphologique et acoustique. Chaque niveau travaille sur des éléments spécifiques, (segments, phonèmes, mots, etc...) qui sont des concaténations d'éléments de niveaux inférieurs (un mot est une suite de phonèmes, un phonème est une suite de segments,...). Chaque S I émet des hypothèses à un ou deux niveaux et ignore les autres.



Structure du blackboard dans HEARSAY II et relation du niveau syntaxique avec les autres niveaux de traitement.

La grammaire du langage est une grammaire hors contexte du même type que la grammaire du système HWIM, elle est spécifique de l'application et incorpore des informations pragmatiques. L'analyse syntaxique se fait en deux temps. Dans une première étape, une matrice d'adjacence à deux dimensions permet de retrouver des paires de mots consécutifs dans le treillis. Ces paires de mots seront la donnée d'un analyseur qui retrouvera les phrases après avoir émis des hypothèses sur les mots manquant initialement dans le treillis.

E.3 - Le système HARPY

Le système HARPY contrairement au système HEARSAY II est plus un système de reconnaissance que de compréhension. En effet, il ne fait pas appel à des contraintes pragmatiques et/ou sémantiques pour interpréter une phrase. Tous les mots ont le même poids lors de la reconnaissance et la syntaxe est très rigide. C'est en partie une des raisons qui permet au système d'avoir de très bonnes performances (95 % de phrases reconnues pour 1011 mots).

L'intérêt de ce système (tout comme le système d'IBM déjà cité au paragraphe C) est d'incorporer dans un réseau unique toutes les variantes possibles des phrases reconnues. Toutes les sources d'informations y sont regroupées et pré-compilées. Chaque transition est pondérée et le processus de reconnaissance et un processus de recherche du meilleur chemin par programmation dynamique. Le principal inconvénient est la taille du réseau qui doit être stocké sur disque et judicieusement paginé.

E.4 - Les systèmes MYRTILLE

Notre équipe s'est intéressée dès 1970 au problème de la reconnaissance puis de la compréhension de la parole ; le système MYRTILLE 1 constitue une étape dans ce travail. Ce système accepte des phrases d'un langage artificiel, sous-ensemble du français parlé. La taille du vocabulaire est plus petite puisque celui-ci possède à peu près 100 mots.

.....

Contrairement aux systèmes américains, MYRTILLE 1 est un système entièrement paramétré où le changement d'application est simple. Ce système est articulé autour d'un analyseur descendant et fonctionne suivant le schéma itératif "hypothèses-tests" au niveau syntaxique.

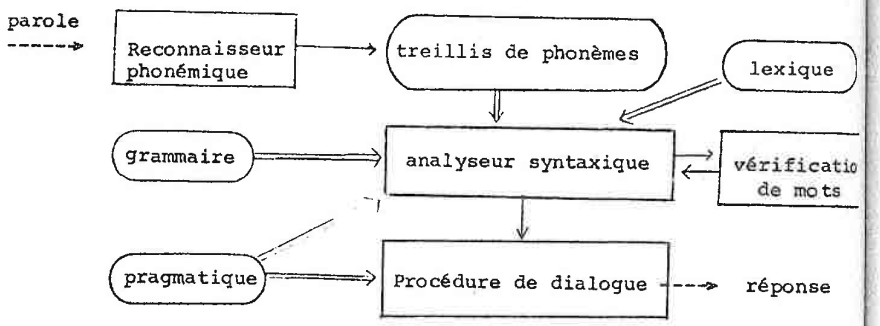


Schéma fonctionnel de MYRTILLE 1

S I ⟹ utilisation d'une SI

----→ transfert de commande et de données

Les données de l'analyseur sont un treillis de phonèmes, résultat de l'analyse phonémique et une grammaire hors contexte représentée à l'aide de règles de réécriture.

L'analyseur fonctionne classiquement de la gauche vers la droite en essayant de retrouver dans le treillis phonétique, à la suite du dernier phonème pris encompte, la transcription de chaque hypothèse : ces hypothèses sont émises en accord avec le contexte de la partie gauche déjà reconnue. Si en un point, la reconnaissance d'un mot sémantiquement faible échoue alors, ce mot est supposé manquant ou une hypothèse d'insertion de phonèmes parasites est faite, une resynchronisation plus loin dans la chaîne est alors tentée.

Le processus de reconnaissance est entièrement guidé par le contexte et des résultats expérimentaux ont montré que le nombre de retours arrière dans la reconnaissance est faible, ce qui valide cette approche dans le cas de vocubulaire de taille moyenne (100 mots) et d'une syntaxe relativement rigide.

.....

Quant aux performances, elles sont honorables puisque MYRTILLE 1 reconnait en temps réel la phrase prononcée à partir d'une transcription phonétique de qualité moyenne - 60 % de bons phonèmes en première position dans le treillis - (temps de réponse de 2 à 4 secondes sur Mitra 125 pour une phrase de durée analogue).

E.5 Le système MYRTILLE II

Présentation générale

Les caractéristiques du langage traité par MYRTILLE II sont les suivantes

a) une syntaxe décrivant un sous-ensemble assez vaste du français parlé (la principale restriction étant la non prise en compte des références pronominales et des propositions relatives).

b) un vocabulaire de plus de 300 mots sans compter les diverses occurences provenant d'un même mot (conjugaisons des verbes, pluriels des noms et des adjectifs, féminins des adjectifs)

C) une sémantique restreinte à un univers particulier : l'application test choisie correspond à un centre de renseignements météorologiques.

Résultats du système

Le résultat souhaité est une représentation syntaxico-sémantique de l'énoncé oral traité. Cette représentation est la plus précise possible ; néanmoins, des résultats incomplets sont acceptés, spécialement en cas d'échec de la reconnaissance sur une partie du signal de départ. Dans ce sens, ce système est plus un système de compréhension que de reconnaissance mot à mot.

Données du système

Les données prises en compte par ce système sont de deux ordres :

1) une représentation acoustico-phonétique de l'énoncé à traiter

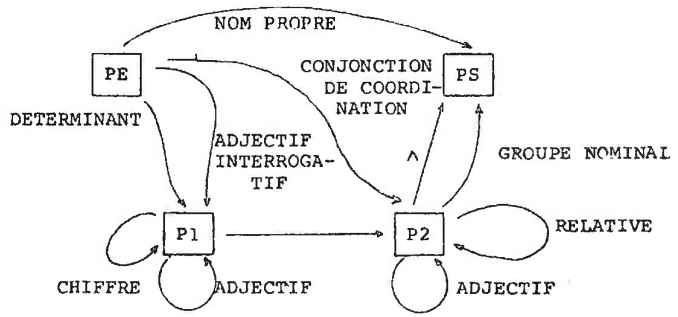
2) les définitions syntaxique , sémantique, et lexicale du langage utilisé.

.....

Définition des structures syntaxiques du langage

MYRTILLE II n'utilise pas, à proprement parlé, de structures syntaxiques mais des structures syntaxico-sémantiques car l'information syntaxique est complétée par des informations sémantiques.

La représentation choisie est assez proche d'un A.T.N. puisque les structures syntaxiques sont représentées par un réseau à noeuds procéduraux. Ces procédures incorporent des informations sémantiques prosodiques et pragmatiques et permettent d'influer sur le déroulement de l'analyse alors que ceci ne se fait pas dans un A.T.N. La figure, ci-dessous, donne un exemple d'une portion de réseau correspondant à un groupe nominal, G N. (30).



P E = procédure d'entrée dans le réseau G N

P S = procédure de sortie du réseau G N

P 1 = procédures internes ; P 2 teste l'éventuelle présence
P 2 de la proposition "de" introduisant un complément du nom.

RELATIVE :relative commençant par qui ou que.

Ce réseau rend compte, par exemple, de la structure de la phrase française suivante :

"de gros risques de formations rapides de petites plaques éparses de verglas".

.....

Les procédures internes ont pour but essentiel de déterminer une hiérarchie pour la prise en compte de sorties possibles du noeud, compte tenu des entrées et du contexte syntaxico-sémantique.

Position de l'analyseur dans le système

Le système de compréhension MYRTILLE II fonctionne suivant le principe "hypothèses-tests". L'ensemble des mots hypothèses est déterminé de la façon suivante : initialement, tous les mots du vocabulaire sont hypothèses et chaque processus mettant en oeuvre une source d'information cherche à réduire cet ensemble. Comme le système est conçu pour être implanté sur une machine multiprocesseur, l'analyseur "PARSER" travaillera en parallèle avec les processus "SEMAN" et "PRØPHØN" qui mettent en oeuvre respectivement les sources d'information sémantique et prosodique.

Alors que "PARSER" et "SEMAN" ont le pouvoir d'enlever des mots à l'ensemble des hypothèses, "PRØPHØN"ne peut que les pondérer, car les informations prosodiques ne sont pas assez sûres pour supprimer un mot d'un ensemble, par contre elles peuvent confirmer une hypothèse en lui attribuant un bon poids. Cet ensemble d'hypothèses servira de donnée à un module de reconnaissance de mots qui les recherchera dans le treillis à la suite du dernier mot reconnu. Après cette étape de validation, les trois processus "SEMAN", "PARSER" et "PROPHON" seront réactivés pour une étape suivante.

L'analyse se fait de la gauche vers la droite mais, en cas d'échec de la procédure de validation, une reconnaissance par "ilots de confiance" peut être tentée. Dans ce type de recherche, il s'agit de retrouver dans le treillis, en fonction de dépendances conceptuelles, des mots avec un bon score puis de rechercher d'après des critères syntaxico-sémantiques des mots adjacents; certains mots sémantiquement faibles pouvant être omis. Cette technique permet de prendre des libertés vis-à-vis de la syntaxe, de la même façon qu'un locuteur peut le faire.

.....

P) DESCRIPTION D'UN ANALYSEUR SYNTAXIQUE PROCEDANT A PARTIR D'UN ILOT DE CONFIANCE

A) Rappels sur la thérorie des langages

B) Généralités sur les algorithmes d'analyse syntaxique

- B.1 définition de l'analyse syntaxique.
- B.2 la méthode descendante ou prédictive
- B.3 la méthode ascendante
- B.4 comparaison entre ces deux méthodes

C) Rappels sur l'analyse ascendante L R

- C.1 théorème sur le mot des racines
- C.2 construction de la pile de l'analyseur
- C.3 l'algorithme L R (k)
- C.4 réduction de l'indéterminisme dans le cas ou il n'y a pas d'ambiguité sur les terminaux.

D) Algorithme d'analyse syntaxique ascendant débutant en un point quelconque de la chaine d'entrée.

- D.0 intérêt
- D.1 exemple
- D.2 principe
- D.3 algorithme abstrait
- D.4 description des objets
  - D.4.1 la procédure enracinement
    - a) rôle
    - b) principe
  - D.4.2 la procédure Lexico
  - D.4.3 la procédure mise-à-jour
  - D.4.4 la procédure choix
  - D.4.5 représentation d'un arbre
  - D.4.6 représentation des objets P et E
  - D.4.7 Représentation des automates
    - a) détermination de l'automate associé à une analyse latérale
    - b) représentation physique

D.5 Algorithmes des procédures

1) enracinement

2) lexico

3) mise à jour

E) Possibilité de prendre de la liberté vis-à-vis de la syntaxe

E.1 solution proposée

E.2 la procédure hypothèse

F) approche du déterminisme

G) situation de la méthode par rapport aux méthodes ascendantes et descendantes

.....

A) Rappel sur la théorie des langages

a) Monoïde libre sur un alphabet

Soit V un ensemble appelé alphabet. On appelle $V^{\oplus}$, le monoïde libre sur V, l'ensemble des suites finies d'éléments de V muni de la concaténation notée $\|$. L'élément neutre pour cette opération sera noté Λ, c'est la suite de longueur nulle.

Exp. : V = (a, b, c,), abc $\in V^{\oplus}$

ac $\in$ V

abc $\|$ ac = abcac

b) automate

soit S un ensemble dont les éléments seront appelés états

A un ensemble appelé alphabet

$\mathcal{T}$ un sous-ensemble de S x A $\cup \{\Lambda\}$ x S

dont les éléments seront appelés "transition"

S' un sous-ensemble de S

s_0 un élément de S

Le quadruplet $(S, s_0, S', \mathcal{T})$ sera appelé un automate sur A.

Représentation d'un automate

soit $\mathcal{A} = (S, s_0, S', \mathcal{T})$

$\mathcal{T}$ définit une relation ternaire sur S x A $\cup \{\Lambda\}$ x S.

On peut projeter cette relation sur S^2 et obtenir une relation binaire que l'on peut représenter, de la même façon qu'un graphe, à l'aide d'arcs joignant des points de S.

Si chaque arc (s, s') est étiqueté par l'élément a de A tel (s,a,s') appartienne à $\mathcal{T}$, alors on a une représentation de $\mathcal{T}$.

Exp. :

S = $\{n \in N,\ n \leq 14\}$

A = $\{E, T, F, +, *, (,), Id\}$

s_0 = 0

S' = $\{3, 4, 8, 9, 11, 14\}$

.....

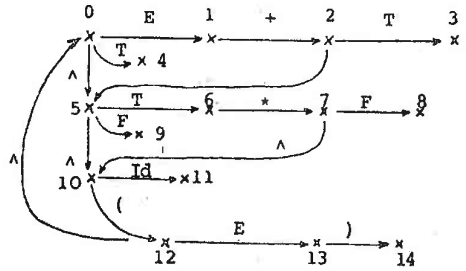


Figure III.1

Définition 1

On appelle calcul de $\mathcal{A}$, une suite finie de transitions (s_i, a_i, s'_i) $i = 0, n$ telles que : $s'_i = s_{i+1}$ $i = 0, n-1$

s_o est l'origine,

s_n est l'extrémité du calcul.

Définition 2

Soit $\mathcal{A}$ un automate. On appelle donnée relative à un calcul $((s_i, a_i, s'_i))_{i = 0, n}$ le mot $a_0 \ldots a_n$.

.....

Définition 3

Un calcul de $\mathcal{A}$ est réussi si son origine est s_0 et son extrémité est dans S'.

Remarque

Si on représente un automate à l'aide d'un ensemble d'arcs joignant des points de S, les calculs sont représentés par des chemins du graphe et les données relatives à ces calculs sont les mots formés en concaténant les éléments de V étiquetant les arcs formant un chemin.

Un calcul réussi sera représenté par un chemin d'origine s_0 et d'extrémité dans S'.

Langage régulier

Soit A un alphabet, $A^{\boxtimes}$ le monoïde libre sur A. Un sous-ensemble L de $A^{\boxtimes}$ est un langage régulier s'il existe un automate sur A tel que L soit formé des données relatives aux calculs réussis de $\mathcal{A}$. On dira que $\mathcal{A}$ reconnaît L.

Définitions relatives aux ramifications

Soit V un alphabet. Nous noterons $\hat{V}$ l'ensemble des ramifications sur V. Nous rappelons que $\hat{V}$ est muni de deux lois de composition

Une loi interne + : $V \times V \to V$

une loi externe $\boxtimes$: $V \times V \to V$

et que toute ramification r non vide s'écrit de manière unique

$r = a \boxtimes s + t$

$a \in V$

$s, t, r \in \hat{V}$

Exp. : $s = \begin{matrix} c \\ d\ e \end{matrix}$, $t = \begin{matrix} b \\ | \\ e \end{matrix}$, $r = \begin{matrix} a \\ c\ \ b \\ d\ e\ \ e \end{matrix}$

Λ désigne la ramification élément neutre pour le monoïde $(V, +)$.

Mot des racines d'une ramification

Considérons l'application :

$\rho : \hat{V} \to V^{\boxtimes}$

définie de la façon récurrente suivante :

$\rho(\Lambda) = \Lambda$

$\rho(a \boxtimes s + t) = a\, \rho(t)$

Mot des feuilles d'une ramification

Considérons l'application définie de la façon suivante :

$\varphi : \hat{V} \to V^*$

$r \to \varphi(r)$

$\varphi(\Lambda) = \Lambda$

$\varphi(a * s + t) =$ si $s = \Lambda$ alors $a\ \varphi(t)$

sinon $\varphi(s)\ \varphi(t)$

$\forall a \in V$

$\forall s, t \in \hat{V}$

Exp. :

$r =$

, $\varphi(r) = bc$.

Mot des feuilles dans T

Soit $T \subset V$

et φ_T définie de la façon récurrente

$\varphi_T(\Lambda) = \Lambda$

$\varphi_T(a * s + t) =$ si $s = \Lambda$ alors

si $a \in T$ alors $a\ \varphi_T(t)$

sinon $\varphi_T(t)$

sinon $\varphi_T(s)\ \varphi_T(t)$

(Cela revient à ne prendre dans $\varphi(r)$ que les feuilles qui sont dans T)

Famille de prédécesseur y de V

Soit $y \in V$

$r \in \hat{V}$ et $r = a * s + t$

$a \in V$

$s, t \in \hat{V}$

On définit $F_y : \hat{V} \to P(V^*)$

$r \to F_y(r) \subset V^*$

par récurrence, de la façon suivante

$F_y(\Lambda) = \emptyset$

$F_y(a * s + t) =$ si $a = y$ alors $\{\rho(s)\} \cup F_y(s) \cup F_y(t)$

sinon $F_y(s) \cup F_y(t)$

Exp.

$r =$

$F_E(r) = \{E+T, T\}$.

$F_y(r)$ est appelé "famille de prédécesseur y de r".

d) Rappels et notations utilisées sur les grammaires à contexte libre

Définition d'une grammaire à contexte libre

Soit :

N un ensemble appelé ensemble de non terminaux

T un ensemble appelé ensemble de terminaux

X un élément de N appelé "axiome"

::= une relation binaire entre N et $(NUT)^*$

V=NUT un ensemble appelé alphabet,

alors le quadruplet G = (N, T, ::=, X) s'appelle une grammaire.

Un couple $A ::= \lambda$ $(A \in N, \lambda \in V^*)$ s'appelle une règle de grammaire.

On dit que A se réécrit en λ et que λ derive de A.

Une grammaire sera définie par la donnée de N, T, X et de l'ensemble des règles.

Exp. : N = {E, T, F}

T = {+, Id, (,),* }.

X = E

$E \to E+T$

$T \to T*F$

$F \to Id$

$F \to (E)$.

S'il n'existe pas de règle

$A ::= \Lambda$

la grammaire est dite de Chomsky ou à contexte libre.

On la note souvent C-grammaire.

Nous supposerons en plus, afin de rendre une analyse syntaxique ascendante possible, qu'aucune règle n'est du type

A::=A.

$\mathcal{D}(G)$: l'ensemble des ramifications engendrées au sens large sur

Soit G = (N, T, ::=, X) une C-grammaire.

On définit $\mathcal{D}(G)$ comme l'ensemble des ramifications vérifiant :

$\forall r \in \mathcal{D}(G)$

$\forall A \in N$

$\forall B \in T \qquad (\alpha \in F_A(r) \Leftrightarrow A::=\alpha\ )$ et $F_B(r) = \emptyset$.

R(G)

R(G) est le sous ensemble de $\mathcal{D}(G)$ tel que le mot des racines de chacun de ses éléments soit égal à X.

Langage défini par une grammaire

Soit G une C-grammaire, alors $\Phi_T(R(G))$ est le langage engendré par G ; il est noté L(G).

Notation

Soit G = (N, T, ::=, X) une C-grammaire.

On définit $G_1 = (N_1, T_1, \rightarrow, X_1)$ en posant

$N_1 = N \cup \{X_1\}$

$T_1 = T \cup \{\vdash, \dashv\}$.

et en ajoutant aux règles de G la règle $X_1 \rightarrow \vdash X \dashv$

Il est évident que si α est un mot de L(G), alors $\vdash \alpha \dashv$ est un mot de $L(G_1)$ et tous les mots de $L(G_1)$ ont cette forme, $\vdash$ et $\dashv$ ont donc des sens de marqueurs de début et fin de phrase.

Dans notre application, ces symboles seront représentés par les extrémités du treillis phonémique d'entrée.

Grammaire réfléchie d'une grammaire

Soit G = (N, T, $\rightarrow$, X) une C-grammaire, alors on considère la relation binaire entre N et V^* notée $\tilde{\rightarrow}$ définie par :

$\forall A \in N$

$\forall \lambda \in V^*$

$A \tilde{\rightarrow} \lambda \Leftrightarrow A \rightarrow \tilde{\lambda}$

$\tilde{\lambda}$ désigne le réfléchi de λ . Le concept de mot réfléchi est illustré par l'exemple :

$\lambda = aN'+d \qquad a, N', +, d \in V$

$\tilde{\lambda} = d+N'a$

$\tilde{G} = (N, T, \tilde{\rightarrow}, X)$ définit la grammaire réfléchie de G, $\tilde{G}$ est aussi une C-grammaire.

Proposition (sans démonstration)

$\widetilde{L(G)} = L(\tilde{G})$

Automate associé à une grammaire

Soit G = (N, T, $\rightarrow$, X) une C-grammaire. Quelque soit A, élément de N, considérons tous les seconds membres de règle

$A \rightarrow \lambda$

On définit un langage fini, donc régulier.

Soit $\mathcal{A}_A$ l'automate reconnaissant ce langage

$\mathcal{A}_A = (S_A, s_{0_A}, S'_A, \mathcal{T}_A)$.

Remarque :

$\forall A, B \in N$, si $A \neq B$ alors $S_A \neq S_B$ (par construction)

Soit $S_G = \bigcup_{A \in N} S_A$

$s_{0_G} = s_{0_X}$

$S'_G = \bigcup_{A \in N} S'_A$

Considérons $\mathcal{T}$ l'ensemble formé de toutes les transitions des $\mathcal{T}_A$ quand A décrit N et des transitions vides vérifiant les deux propriétés :

- $\forall A \in N$, $\forall B \in N$ $\quad (s_1, B, s_2) \in \mathcal{T}_A \Rightarrow (s_1, \Lambda, s_{0_B}) \in \mathcal{T}$
- Il n'y a pas dans $\mathcal{T}$ d'autres transitions vides.

Exp. :

l'automate de la figure III est relatif à la grammaire

E → E + T / T

T → T * F / F

F → Id / (E).

De la même façon, on peut définir $\mathcal{A}_{\tilde{G}}$

Dans la suite, nous noterons $\mathcal{A}$ l'automate $\mathcal{A}_G$ et $\mathcal{A}'$ l'automate $\mathcal{A}_{\tilde{G}}$

$\mathcal{A}$ = (S, s_0, S_F, $\mathcal{C}$)

$\mathcal{A}'$ = (S', s'_0, S'_F, $\mathcal{C}'$)

Proposition fondamentale

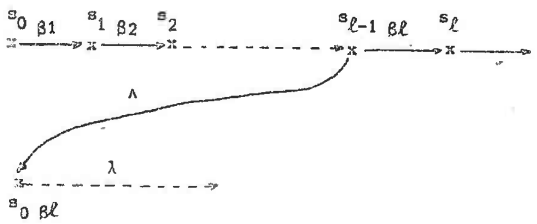
Si $\beta_1 \ldots \beta_\ell$ est une donnée relative à un calcul de $\mathcal{A}$ et si β_ℓ est un élément de N, alors $\beta_1 \ldots \beta_{\ell-1}\ \lambda$ est une donnée relative à un calcul de $\mathcal{A}$, quand λ est un second membre de la règle $\beta_\ell \rightarrow \lambda$

Démonstration :

Il suffit de trouver deux calculs de $\mathcal{A}$, l'un d'origine s_0 et d'extrémité $s_{\ell-1}$ dont une donnée relative est $\beta_1 \ldots \beta_{\ell-1}$ et un autre d'origine $s_{\ell-1}$ et dont la donnée relative est λ .

On remarque que puisque $(s_{\ell-1}, \beta_\ell, s_\ell)$ est une transition de $\mathcal{A}$, alors $(s_{\ell-1}, \lambda, s_{0\ \beta_\ell})$ est une transition de $\mathcal{A}$. Or, $s_{0\ \beta_\ell}$ est l'origine de tous les calculs ayant pour données les seconds membres de règles $\beta_\ell \rightarrow \lambda$

On a bien le résultat.



Automates indéterministes sans transitions vides et automates déterministes.

L'automate associé à une grammaire, tel qu'il vient d'être défini est un automate indéterministe avec transitions vides. En effet, si on se reporte à la figure (III.3), on remarque que la lecture de T permet de passer de l'état 2 à l'état 3 ou 6.

Il existe des algorithmes (AS 3) qui permettent de transformer un tel automate en un automate sans transition vide, possédant le même ensemble d'états et reconnaissant le même langage régulier. La figure III.2 représente l'automate indéterministe sans transition vide déduit de l'automate de la figure III.1.

On peut aussi transformer cet automate en un automate déterministe (la lecture d'un caractère fait passer d'un état donné à un seul état d'arrivée), mais l'ensemble d'état est modifié. La figure III.3 représente l'automate déterministe. Pour plus de détails sur ces transformations, on peut se reporter à (AS3) et (AS2).

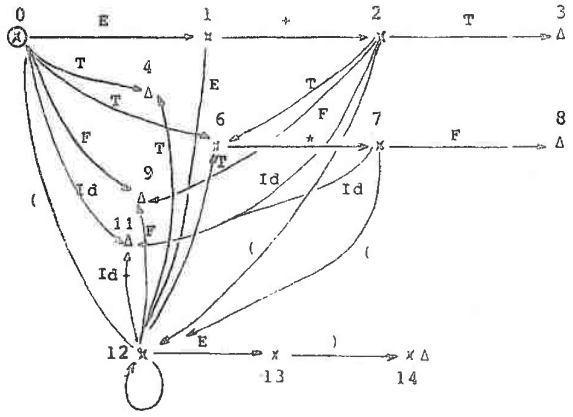


Figure III.2

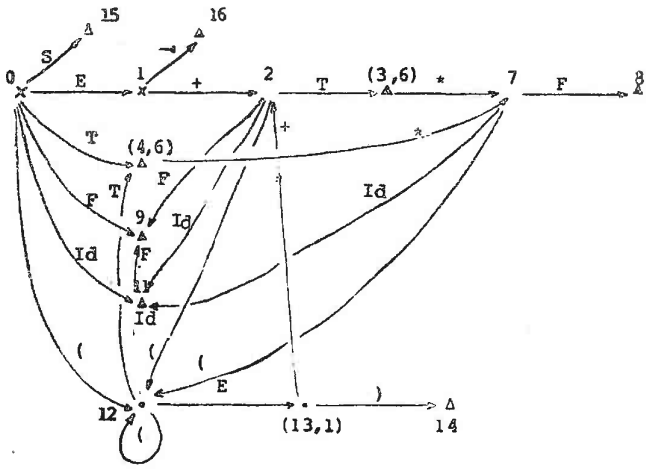


figure III.3

Cet automate est associé à la grammaire des expressions arithmétiques augmentées de la règle S ⟶ E ⊣, ⊣ étant le marqueur de fin de donnée et S le nouveau non terminal qui sert d'axiome.

B) Généralités sur les algorithmes syntaxiques

B.1 Définition de l'analyse syntaxique

Soient G = (N, T, ⟶, X) une grammaire à contexte libre et $\alpha = a_1 \ldots a_n$ un mot de $T^{\boxtimes}$. Faire une analyse syntaxique de α revient à chercher, s'il existe, un arbre r de R (G) tel que $\varphi_T^{-}(r) = \alpha$

Il existe parmi l'ensemble des méthodes d'analyse, deux méthodes de construction de l'arbre syntaxique.

B.2 La méthode d'analyse descendante ou prédictive

Dans cette méthode, l'arbre syntaxique est construit de la racine vers les feuilles (de haut en bas) et de la gauche vers la droite.

Une procédure d'analyse descendante transforme l'axiome de la grammaire en une suite de mot appelée "cible" de $V^{\boxtimes}$ jusqu'à obtenir la donnée α. Chaque étape de la transformation consiste en la dérivation d'un non terminal dans la dernière cible obtenue.

Pour des raisons de simplicité d'algorithme, c'est toujours le premier non terminal à partir de la gauche que l'on dérive et la comparaison avec la donnée s'effectue au fur et à mesure qu'apparaissent des terminaux dans la cible. Décrivons la procédure ANALYSE (A) qui effectue une suite de transformations sur une suite débutant par A afin de faire disparaître ce non terminal.

ANALYSE (A)

- si A∈T alors, lire le premier caractère gauche de la donnée restante ; soit C ce caractère
- si A = C alors écrire A sinon erreur fsi
- supprimer C de la donnée

sinon ! ⊠ A appartient à N et il faut choisir une règle dont le membre de gauche est A et faire une dérivation.

C'est maintenant qu'apparaît le caractère prédictif de l'analyse

.....

choix de $B_k \longrightarrow B^1_k \; B^2_k \; \ldots \; B^{\ell_k}_k$

|⊠ la cible débute par B^1_k ⊠|

. pour j de 1 à ℓ_k répéter

appel Analyse (B^j_k)

fait

. écrire (k) |⊠ nous mémorisons le numéro de la règle appliquée ⊠|

fsi

FIN ANALYSE

La procédure d'analyse syntaxique s'écrira donc :

- ANALYSE (X) /* X est l'axiome de la grammaire */
- lire le caractère suivant dans la donnée
- si C ≠ ⊣ alors erreur fsi

| ⊠ ⊣ est le marqueur de fin de donnée ⊠|

Théorème

L'appel d'ANALYSE (X) aura pour résultat l'impression de la représentation postfixée de l'arbre de $\mathcal{Q}$.

Ce théorème est démontré dans (AS3) et l'algorithme est bien développé dans (AS 1)

B.3 La méthode d'analyse ascendante

Dans cette méthode, l'arbre syntaxique est construit de bas en haut (ie des feuilles vers la racine) et l'analyse n'a plus de caractère prédictif. Au cours d'une étape de l'analyse, sera construite une nouvellle arborescence, notée R_n à partir de R_{n-1} et des prochains terminaux à lire dans la donnée. Il faut considérer deux sortes d'étapes

a) les étapes de lecture :

Dans ce cas on a : $R_{n+1} = R_n + C$ (+ est l'opérateur l'addition entre arbres)

Cela signifie aussi que le nouveau mot des racines est formé de l'ancien auquel a été concaténé C.

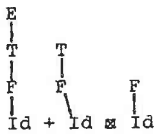
B) les étapes de réduction :

Une partie droite du mot des racines, quand elle est un second membre de règle, peut être remplacée par le non terminal qui est le membre de gauche de cette règle.

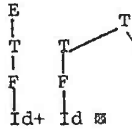
Exemple :

supposons avoir la règle $T \longrightarrow T \boxtimes F$ et

$Rn =$



nous avons $\rho(Rn) = E + T \boxtimes F$, après la réduction par la règle $T \longrightarrow T \boxtimes F$ nous aurons :



et $\rho(R_{n+1}) = E + T$

Quelle que soit l'étape de l'analyse, c'est toujours sur la partie droite du mot des racines que s'effectue les modifications si bien que la variable qui représente le mot des racines sera une pile. L'analyse se termine quand on a un mot des racines de la forme $x \dashv$.

Algorithme

|⊠ initialisation ⊠|

$\rho(r) = \emptyset$

tant que ($\rho(r) \neq x \dashv$) faire

choix(

a : |⊠ lecture ⊠|

lire (C)

$\rho(r) = \rho(r)$ C

écrire (C)

b : |⊠ réduction ⊠|

choix de $\lambda \; V^{@}$ tel que $\rho(r) = \partial\lambda$

choix de $B_k \in N$ tel que $B_k \longrightarrow \lambda$

$\rho(r) = \partial B_k$

écrire (k)

C : erreur)

fait

Théorème

Quand la donnée est un mot du langage, l'éxécution de cet algorithme a pour résultat la représentation post-fixée de l'arbre syntaxique de la donnée.

Ce théorème est démontré en (AS3).

.....

Comparaison entre les deux méthodes

Ces deux algorithmes qui sont des algorithmes de construction d'arbres syntaxiques peuvent s'intégrer dans un niveau syntaxique d'un système de reconnaissance de la parole. Le système MYRTILLE du C.R.I.N incorpore un analyseur descendant et le système KEAL du C. N.E.T. utilise un analyseur ascendant. Les deux algorithmes peuvent construire l'arbre syntaxique dans un processus guidé par le contexte ou par les données.

Parmi les méthodes d'analyse ascendante, il faut citer la méthode LR (1) qui permet d'analyser tous les langages déterministes, chose que les analyseurs descendants les plus puissants (méthodes LL (k)) ne peuvent pas faire. (voir AS 1 et AS 2)

Un analyseur L R (1) prend ces décisions en fonction de tout le contexte déjà analysé et du prochain caractère dans la donnée ce qui fait qu'une analyse, suivant le principe "hypothèse-test", sera déterministe s'il n'y a pas d'ambiguité sur les terminaux. En réalité, comme on ne peut pas s'affranchir de cette ambiguité, cela nous permettra d'avoir une zone moins vaste de sauvegarde des contextes ambigus et un déroulement plus rapide de l'analyse.

C) Rappel sur l'analyse ascendante L R

Reprenons ce qui vient d'être dit pour l'expliciter davantage.

Au cours de l'analyse, il est évident que l'état de l'arborescente déjà construite sera un paramètre des trois instructions choix.

Nous allons voir qu'il existe une représentation du mot des racines qui permette de décider rapidement si une configuration peut être réduite et dans ce cas quelle est la règle à appliquer.

C.1 Théorème sur le mot des racines

Considérons l'automate associé à G. Nous allons montrer qu'à chaque étape d'une analyse réussie le mot des racines

$\rho(r) = b_1\ b_2 \ldots\ldots\ b_m \qquad b_i \in V$

peut être remplacé par une pile d'états de l'automate

$P = s_0\ s_1 \ldots\ldots s_m$

telle que la transition de s_i à s_{i+1} soit due à la lecture de b_{i+1}.

.....

Exp. : dans l'automate de la figure 3, au mot E + (Id est associé la pile 0, 1, 2, 12, 11.

Démonstration

Soit $\rho(r_o)$, $\rho(r_1)$...., $\rho(r_n)$ la suite des mots des racines d'une analyse réussie, montrons par récurrence que chaque élément de cette suite vérifie la propriété.

Nous avons : $\rho(r_n) = X$

Ce mot par construction est une donnée associée à un calcul réussi de $\mathcal{A}$ (voir figure III.3 où S est l'axiome).

Supposons que $\rho(r_{n-k})$ vérifie la propriété de récurrence, c'est-à-dire que si $\rho(r_{n-k}) = b_1 \ldots\ldots b_m$ alors il existe $P_{n-k} = s_o \ldots\ldots s_m$ telle que $(s_i, b_{i+1}, s_{i+1}) \in \mathcal{C}, \forall i = o, m-1$ et montrons que $\rho(r_{n-k-1})$ vérifie la même propriété.

- Soit on passe de $\rho(r_{n-k-1})$ à $\rho(r_{n-k})$ par lecture de b_m, donc $\rho(r_{n-k-1}) = b_o\ b_1 \ldots\ldots b_{m-1}$ et nous aurons $P_{n-k-1} = s_o \ldots\ldots s_{m-1}$
- Soit on passe de $\rho(r_{n-k-1})$ à $\rho(r_{n-k})$ par réduction par la règle $b_m \rightarrow \lambda$, donc si nous avons $\rho(r_{n-k}) = b_1 \ldots\ldots b_m$ avec $b_m \in N$ et $P_{n-k} = s_o \ldots\ldots s_{m-1}\ s_m$ alors nous aurons : $\rho(r_{n-k-1}) = b_1 \ldots\ldots b_{m-1}\ \lambda$. Or d'après la proposition fondamentale s_{m-1} est l'origine d'un calcul admettant λ pour donnée, soit $e_o \ldots e_2$ la suite d'états de ce calcul, nous aurons : $P_{n-k-1} = s_o \ldots\ldots s_{m-1}\ e_o \ldots\ldots e_2$, de plus e_2 est un état final de $\mathcal{A}$.

C.2 Construction de la pile de l'analyseur

Le théorème C.1 démontre que toute expression du mot des racines est nécessairement une donnée relative à un calcul de $\mathcal{A}$ et que les configurations réductibles appartiennent au langage régulier reconnu par $\mathcal{A}$.

Nous pouvons tirer trois conclusions sur le déroulement d'une analyse :

1) on peut remplacer le mot des racines par une pile P d'états de initialisée avec l'état initial de .

.....

2) si c est le caractére lu quand s est le sommet de pile, alors $(s, c, s') \in \delta$.

3) Si la règle $B \longrightarrow \lambda$ est utiliséedans une réduction quand s est le sommet de pile alors nécessairement s est un état final tel que :

$a_1 \ldots a_n = \lambda$

L'ensemble des états de $\mathcal{A}$ peut être partitionné en 3 sous-ensembles

. l'ensemble des état de lecture, EL :
Quand le sommet de pile appartient à EL, il est nécessaire, pour continuer l'analyse, d'effectuer une lecture.

. l'ensemble des états de réduction, ER:
C'est un sous ensemble d'états finaux de $\mathcal{A}$.
Quand le sommet de la pile appartient à ER, il faut effectuer une réduction par une règle associée à cet état.

. l'ensemble des états inadéquats, EI :
quand un état de ce type est sur le sommet de pile, il y a ambiguité sur l'action à accomplir:
On peut réduire par plusieurs règles,
On peut lire ou réduire (exemple état (3,6) de la figure III.3).
En compilation cette ambiguité se lève en prenant connaissance des caractères suivants dansla donnée et de l'état de la pile.

C.3 l'algorithme LR (k)

Nous donnons maintenant l'algorithme dans le cas où il n'y a pas d'interminisme sur les terminaux en donnée.

```
/¤ initialisation ¤/
/¤ S est le premier état à empiler
   SF est l'état de fin d'analyse ¤/
P = ∅
EMPILER (S)
tant que (sommet-de-pile ≠ SF) faire
S = sommet-de-pile
si (S ∉ E L) alors LIRE (C)
```

.....


```
/ ¤ C est le prochain terminal à lire dans la donnée ¤ /
CALCUL(S,C,S2)
/ ¤ calcul (.) est la fonction qui calcule l'état S2 tel que
(S,C,S2) soit une transition de l'automate, dans le cas où
cette transition n'existe pas, il y a appel au module ERREUR ¤ /
ECRIRE (C)
EMPILER (S2)
sinon si (S ∈ E R) alors / ¤ il faut réduire l'arborescence, une
fonction NØM définie sur chaque état final associe le membre de
gauche de la règle à appliquer, la fonction LREGLE définit la
longueur du membre de droite ¤ /
A = NØM (S)
Pour k de 1 à LREGLE (S) faire DEPILER ; fait
CALCUL (sommet-de-pile, A, S2)
EMPILER (S2)
sinon / ¤ S est un état ambigu ¤ /
CHOIX (P, LECTURE, LREGLE,A)
/ ¤ la fonction CHØIX teste les k terminaux suivants dans la
donnée et la pile P pour décider du type de l'étape. Si le
booleen LECTURE est vrai il faut lire sinon LREGLE et A
indiquent la réduction à effectuer ¤ /
si (LECTURE = VRAI) alors   LIRE (C)
                            CALCUL (S, C, S2)
                            EMPILER (S 2)
sinon pour k de 1 à LREGLE faire
      DEPILER
      fait
      CALCUL (sommet-de-pile, A, S 2)
      EMPILER (S2)
      fsi fsi fsi fait
```

C.4 Réduction de l'indéterminisme dans le cas où il n'y a pas d'ambiguité sur les terminaux

La réduction de l'indéterminisme est nécessaire quand l'automate possède des états inadéquats où il y a des conflits sur l'action à accomplir.

.....

Il est possible de définir des classes de grammaires (LR (0), SLR (k), LALR (k)) pour lesquelles la connaissance du sommet de pile et éventuellement des k caractères suivants dans la donnée permet de lever l'ambiguité. Pour cela, on définit une fonction sur l'ensemble des couples (états de A, k-upplets de terminaux) qui associe à chaque couple l'action correspondante. La grammaire est déterministe si les images inverses de chaque action sont deux à deux disjointes (AS2). Mais il existe des grammaires dans lesquelles une décision à un état ambigu ne peut se prendre qu'avec la connaissance de toute la pile, c'est-à-dire de tout le contexte analysé et non pas du seul sommet de pile. Un tel exemple se trouve en (AS6, PP630).
Dans notre application d'analyse syntaxique du discours parlé fondé sur le principe "hypothèse-test", il est impossible d'avoir un déroulement déterministe de l'analyse à cause de l'ambiguité qui règne sur les terminaux, une solution consiste à définir une stratégie de retour-arrière ; la zone de sauvegarde des points de reprise sera plus grande dans le cas d'une analyse à l'aide d'une grammaire LR (k) ($k > 1$) qu'avec une grammaire LR (1) car en plus viendront s'ajouter dans la zone de sauvegarde les contextes syntaxiquement ambigus.
En fait, on montre que tout langage déterministe possède une grammaire LR (1), donc la zone de sauvegarde des points de reprise dans une stratégie "hypothèses-tests" ne contiendra que des contextes d'analyse dans lesquels il y a ambiguité sur les terminaux reconnus.
Intuitivement, nous préférons utiliser une méthode d'analyse LR (1) plutôt qu'une méthode LL (k) puisque la méthode ascendante tient compte de la totalité du contexte déjà analysé lors de la prédiction d'hypothèses.

.....

D) ALGORITHME D'ANALYSE SYNTAXIQUE ASCENDANT DEBUTANT EN UN POINT QUELCONQUE DE LA CHAINE D'ENTREE

D.0 Intérêt

Le début d'un treillis phonétique est souvent mal défini et les premiers segments sont souvent d'une qualité médiocre. Ceci peut provenir du bruit ambiant lors de la prise de son ou du fait que le locuteur n'a pas encore posé sa voix. Dans certains cas, des parasites comme un raclement de gorge ou une inspiration accentuée peuvent être interprétés comme un premier phonème. Toujours est-il que les premiers phonèmes ne constituent pas forcément un bon point de départ pour une analyse et il est peut être préférable de localiser et de bien identifier un mot à l'intérieur du treillis afin d'y débuter l'analyse.
De toute façon, même dans un système travaillant de gauche à droite, une analyse du type de celle que nous allons décrire peut être très utile en permettant de travailler localement à partir d'îlots de confiance en cas d'ambiguité. C'est une des possibilités que nous avons prévue dans le système MYRTILLE II.

Le système de reconnaissance américain HWIM fonctionne comme nous l'avons vu, sur ce principe. Nous allons décrire un analyseur ascendant qui peut débuter en un point quelconque de la chaîne d'entrée. Il faut préciser tout de suite que cette stratégie n'a d'intérêt qu'en intelligence artificielle, domaine où les formes à analyser peuvent être constituées de primitives ambigues, car nous avons vu au paragraphe précédent que tous les langages déterministes sont analysables d'une façon LR (1), donc sans retour arrière, de gauche à droite à partir de la première unité lexicographique.

.....

D.1 Exemple d'analyse syntaxique débutant à partir d'un point quelconque de la phrase

Considérons le langage sur $\{a, b, c\}$

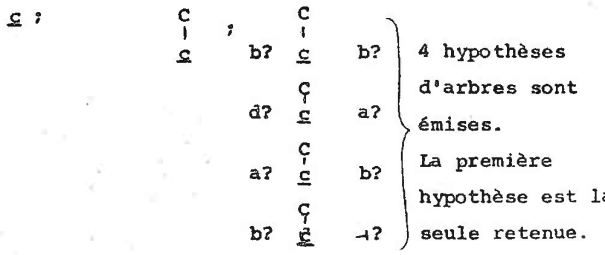
$L = \{a\ b^+\ c^+\} \cup \{d\ b^n\ c\ b^n a\}$ $n \geq 0$, (b^+ indique une répétition non nulle de b).

Une grammaire engendrant L peut être :

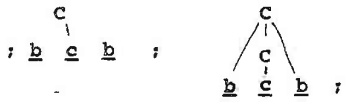
X ⟶ a BC /1/
X ⟶ a CB /2/
X ⟶ dCa /3/
B ⟶ b B /4/
B ⟶ b /5/
C ⟶ b Cb /6/
C ⟶ c /7/

Nous allons décrire la construction de l'arbre syntaxique de la phrase ⊢ abbcbbb ⊣ lorsque le premier caractère pris en compte est c. Cette construction s'effectuera suivant le principe d'hypothèses-tests.

étape ① ② ③

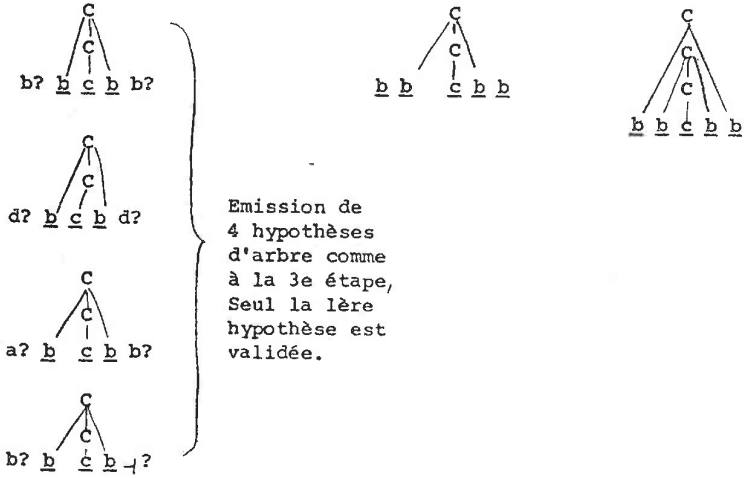


④ ⑤

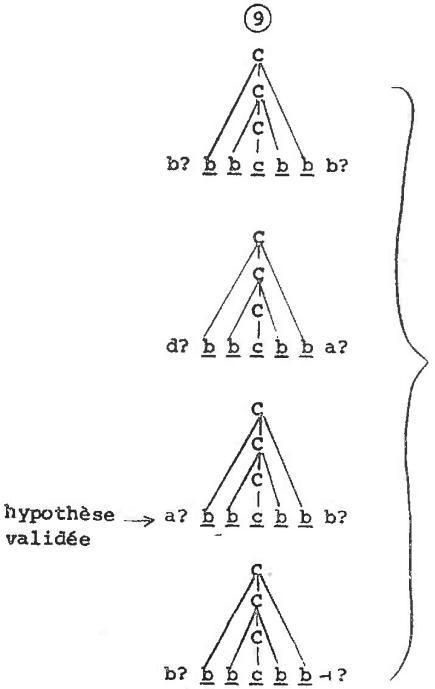


.....

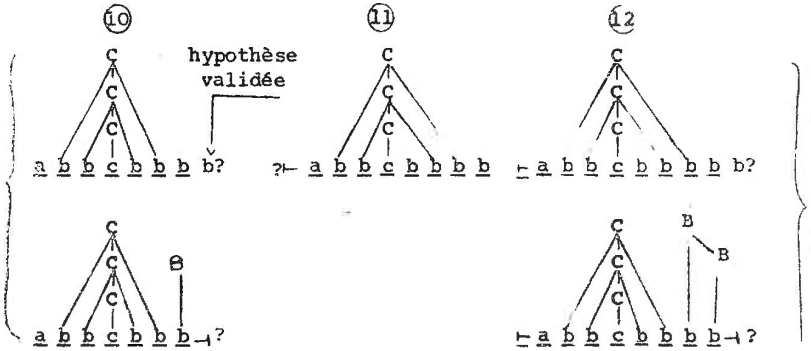
⑥ ⑦ ⑧

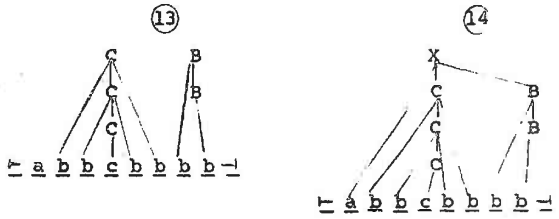


⑨



.....





D.2 Principe de l'algorithme

Supposons donnés une grammaire à contexte libre et un mot ω du vocabulaire terminal. Nous appellerons a le caractère par lequel l'analyse débutera. Dans l'arbre syntaxique (s'il existe) de ω il existera un sous-arbre dont le mot des feuilles contiendra le caractère de début et dont la racine sera le membre de gauche d'une règle où a apparaît dans le membre de droite. Ainsi, dans l'exemple traité, la grammaire contient C→c et l'arbre syntaxique incorpore le sous-arbre C | c

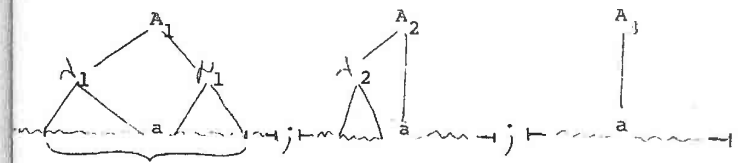
Plus généralement, si on cherche à débuter une analyse en a et que la grammaire contient les règles :

$A_1 \longrightarrow \lambda_1 \; a \; \mu_1$

$A_2 \longrightarrow \lambda_2 \; a$

$A_3 \longrightarrow a$

il faudra essayer de construire un des 3 arbres :



partie de la donnée réécrite en A_1

Le processus de construction sera itéré en considérant que A_1 ou A_2 ou A_3 est reconnu et en recherchant les règles dans lesquelles apparaît ce non terminal. L'analyse s'arrête quand on a reconnu l'axiome et lu toute la donnée ou si une erreur dans la donnée a été détectée.

En conclusion, la construction de l'arbre syntaxique d'une phrase peut être vue comme une suite de constructions d'arbres dans lesquels sont prédits les deux premiers niveaux, ie, la racine et ses fils, le caractère de début de construction étant la racine du dernier arbre construit et/ou le caractère de départ. Dans l'exemple traité, les étapes 2, 5, 8, 14 sont des étapes où un sous arbre de ce type a été achevé.

D.3 Algorithme abstrait

Nous appellerons P l'ensemble des arbres en construction. Si r est un élément de P alors nous noterons $\rho(r)$ son mot des racines

/ ⊠ initialisation ⊠ /

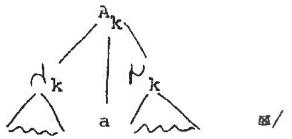
P = ∅

pour chaque règle $A_k \longrightarrow \lambda_k \; a \; \mu_k$ répéter

P = P ∪ {(a, k)}

fait

/ ⊠ (a, k) est la première étape de la construction d'un arbre du type



⊠/

arrêt = faux

jusqu'à arrêt répéter

. CHOIX de (r, k) appartenant à P

. E= ∅ .....

.ENRACINEMENT (r, k, E , arrêt)

/ ◙ ENRACINEMENT effectue une suite d'enracinement sur (r, k) jusqu'à émettre des hypothèses sur les prochains terminaux à reconnaître, les résultats se trouvent dans E qui est formé de triplets $(r_1, k_1, (C_g, C_d)$. C_g et C_d sont les terminaux à valider à gauche et à droite, un des deux terminaux au plus peut-être le mot vide (ie, il n'y a rien à valider de ce côté). ◙ /

. E = ∅

. LEXICØ (E, F)

/ ◙ certains arbres sont rejetés, les arbres acceptés sont dans F ◙ /

. MISE-A-JOUR (F , P)

/ ◙ les arbres mis à jour sont rangés dans P ◙ /

fait

D.4 description des objets

Nous allons décrire le rôle et le principe de chaque procédure afin de mieux comprendre les algorithmes et l'intérêt des structures de données choisies. Ceux-ci seront décrits dans le paragraphe suivant.

D.4.1 La procédure ENRACINEMENT

a) Rôle

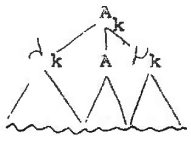
Le rôle de cette procédure est de produire des arbres à partir d'un arbre donné. Dans l'exemple traité, on peut ainsi produire jusqu'à 4 arbres. Les arbres produits possèdent un mot des racines dont au moins une extrémité est un terminal que le reconnaisseur de mot aura à valider, ainsi l'arbre produit sera accepté ou rejeté.

b) Principe

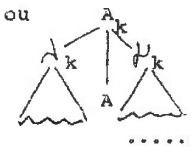
Supposons que (r,k) est l'arbre donné en paramètre d'entrée à la procédure ENRACINEMENT, cette procédure essaiera de construire un arbre du type

$A_k \rightarrow \lambda_k \; A \; \mu_k$ / k /

$\lambda_k, \mu_k \in V^*$



ou



.....

A étant soit le premier caractère pris en compte, soit la racine du dernier arbre construit dans la suite définie au § D.2 On peut distinguer dans r trois parties :

. La partie à droite de A dont le mot des racines se réécrira en μ_k

. La partie à gauche de A " " " " " en λ_k

. et A.

Les enracinements intervenant sur la partie à droite de A sont ceux rencontrés dans une analyse gauche-droite sachant que μ_k est l'axiome et les "voisins" de A_k, les terminateurs. (La relation "voisins" est la relation sur (N x T) définie par $F^{*-1} \circ S \circ \overline{J}^*$; F, S, étant les relations "final", "suivant", et "initial" (AS3)).

Les enracinements intervenant sur la partie à gauche de A sont ceux rencontrés dans une analyse gauche-droite de la donnée réfléchie en considérant la grammaire réfléchie sachant que λ_k est l'axiome et les voisins-gauches de A_k les terminateurs. (On a la relation qui définit les voisins gauches de A_k :

$x \in$ voisins-gauches$(A_k) \Leftrightarrow x \in T$ et $x \; F^{*-1} \circ S \circ \overline{J}^* . A_k$).

Le mode de construction choisi de l'arbre fait que la seule réduction dans laquelle intervient A, est la dernière réduction par la règle

$$A_k \longrightarrow \lambda_k \; A \; \mu_k \quad / k /$$

La procédure ENRACINEMENT initialisera et achevera deux processus indépendants d'enracinement qui travailleront sur chaque côté de l'arbre.

D.4.2 La procédure LEXICØ

Le travail de la procédure ENRACINEMENT sur un arbre s'arrête quand un terminal peut s'y ajouter. Ce terminal est émis alors comme hypothèse. La procédure LEXICØ est chargée de valider ou de rejeter cette hypothèse par un appel au module de reconnaissance de mots RECØN (voir chapitre II). De cette façon se réalise

.....

l'action du niveau syntaxique sur le niveau lexicographique.
Lors de l'appel à la procédure LEXICØ, le processus de reconnaissance est véritablement guidé par le contexte.
Cette procédure entretient des tables qui mémorisent les résultats du reconnaisseur de mots. Ainsi, lors d'un retour-arrière du processus de reconnaissance dans le cas d'une analyse indéterministe, une hypothèse ne sera pas deux fois confirmée.

D.4.3. La procédure MISE-A-JOUR

A partir des terminaux reconnus, cette procédure calcule les nouveaux arbres à partir des résultats de ENRACINEMENT.
Il peut y avoir, à cause des ambiguités syntaxiques ou inhérentes au discours et des limites du système de paramétrisation, plusieurs réponses qui seront rangées dans la zone des points de reprise.

D.4.4. La procédure CHOIX

Nous avons vu que le fonctionnement de l'analyse est obligatoirement indéterministe. La procédure CHOIX est chargée de mettre en oeuvre la stragégie de reconnaissance.
Nous avons essayé les différentes stratégies suivantes.

- "meilleur d'abord" ou "best first". Dans cette statégie, une plausibilité est calculée pour chaque arbre et l'arbre ayant la plus forte plausibilité est choisi.
- "recherche en faisceau" ou "beam search". Dans cette stratégie, un petit nombre d'arbres ayant les plus fortes plausibilités est choisi . Dans notre application l'heuristique qui calculait la plausibilité ne dépendait que des taux de reconnaissance des terminaux constituant le mot des feuilles de l'arbre, ainsi en choisissant tout les arbres ayant la plausibilité la plus forte, on arrivait à mener, en parallèle, la construction des arbres syntaxiquement ambigus.
  Les résultats de ces deux stratégies sont donnés st discutés dans le chapitre suivant.

.....

D.4.5. Représentation d'un arbre

Nous avons choisi d'effectuer sur chaque partie d'un arbre, une analyse de type LR qui permet de tenir compte de tout le contexte déjà reconnu et de repousser le plus tard possible les choix d'ordre syntaxique (AS2). Pour représenter un arbre, il faudra deux piles d'états associés aux deux analyses latérales
Un arbre (r,k) sera représenté par :
(P_g, ip_g, $liste_g$, $score_g$, P_d, ip_d, $liste_d$, $score_d$)
P_g est la pile des états concernant l'analyse latérale gauche.
ip_g pointe dans la chaîne phonétique sur l'extrémité gauche de la sous-chaîne déjà réécrite.
$Liste_g$ est la liste des terminaux reconnus à gauche du point de départ
$score_g$ est une plausibilité de la partie gauche de l'arbre, calculée à partir des taux de reconnaissance des terminaux de $liste_g$. $Score_g$ sert à estimer la qualité de la reconnaissance des terminaux localisés à gauche du point de départ, le nombre est calculé par heuristique.
P_d, ip_d, $liste_d$, $score_d$ sont définis de la même façon et concernent l'analyse de la partie droite de l'arbre.

D.4.6. Représentation des objets P et E

Elle dépend beaucoup du langage de programmation utilisé.
Dans notre application programmée en FØRTRAN nous représentons un arbre par un vecteur à une dimension. Tous les arbres en construction sont regroupés dans un tableau à deux dimensions PILERE qui sert de pile de point de reprise et représente P.
Les variables suivantes caractérisent le premier arbre dans P.

PILERE (1,I)	pointe vers le sommet de pile gauche
PILERE (2,I)	" sur le dernier phonème gauche pris en compte
PILERE (3,I)	" vers la liste des terminaux reconnus à gauche
PILERE (4,I)	est une plausibilité de la partie gauche de l'arborescence

.....

PILERE (5,I) pointe vers le sommet de pile droite

PILERE (6,I) " sur le dernier phonème droit pris en compte

PILERE (7,I) pointe sur la liste des terminaux reconnus à droite

PILERE (8,I) est une plausibilité de la partie droite de l'arborescence

E est représenté par 2 tableaux T1 et CØNTEX.

CØNTEX a la même organisation que PILERE car il contient les arbres fournis en résultat par ENRACINEMENT.

Le tableau T1 (n, 2,2) possède les informations suivantes :

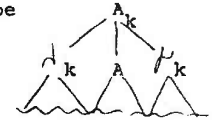
T1 (i, 1, sens) est le sommet de la pile des états de l'arbre qui se trouve à l'indice T1 (i, 2, sens) dans CØNTEX. On considère la pile gauche si sens vaut 1, sinon on considère la pile droite.

Il est intéressant de posséder le sommet de pile car les hypothèses peuvent s'en déduire facilement comme nous le verrons dans les paragraphes suivants.

D.4.7. Représentation des automates

a) <u>détermination de l'automate associé à une analyse latérale</u>

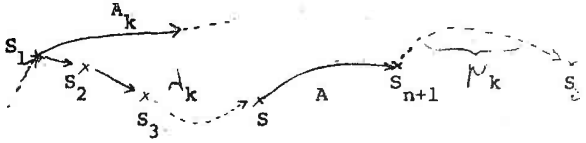
supposons qu'un arbre du type



soit en cours de construction et déterminons l'automate chargé de réécrire en μ_k la suite des terminaux reconnus à droite du caractère de début.

Cet automate peut être déduit facilement de l'automate associé à G à l'aide des deux remarques :

- L'automate associé à G possède l'enchaînement de transition suivant :



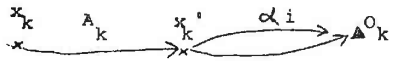
S_ℓ est l' état de réduction par la règle

$A_k \rightarrow \alpha_k \; A \; / \; \mu_k \; / \; k /$

- Lorsque dans une analyse gauche-droite, la pile contient la suite d'états $S_{n+1}\, S_{n+2} \ldots S_\ell$ cela signifie qu'un arbre de racine μ_k a été achevé.

Il suffit donc pour effectuer l'analyse latérale demandée de considérer que l'état initial de l'automate est S_{n+1}.

Pour arrêter l'analyse latérale il est nécessaire que la pile ait la configuration suivante $S_{n+1}\, S_{n+2} \ldots S_\ell$ et que le caractère suivant (donc l'hypothèse suivante) soit un voisin de A_k. Afin de pouvoir réduire en l'état S_ℓ de la même façon que dans une analyse débutant sur le premier caractère (car la pile possède alors au moins ℓ états $S_1, S_2, \ldots, S_n \ldots S_\ell$) nous initialiserons celle-ci avec n états n'apparaissant pas dans $\mathcal{A}$, soit x_k cet état . Une réduction en l'état S_ℓ par la règle $A_k \rightarrow \alpha_k \; A \; \mu_k$ fait apparaître l'état x_k au sommet de la pile ; il suffit de rajouter le sous-automate suivant $\mathcal{A}_k$ pour pouvoir accéder aux voisins de A_k.



sous-automate $\mathcal{A}_k$

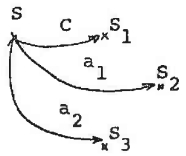
Les α_i sont les voisins de A_k. O_k est l'état final de l'analyse latérale.

Pour chaque non terminal A_k de G nous construirons en plus de $\mathcal{A}$ les sous-automates $\mathcal{A}_k$.

b) <u>Représentation physique</u>

Les automates sont représentés par des listes de transitions. L'accès aux têtes de listes se fait par une fonction suivant l'état d'origine.

Exemple : si l'automate contient le schéma de transition suivant :



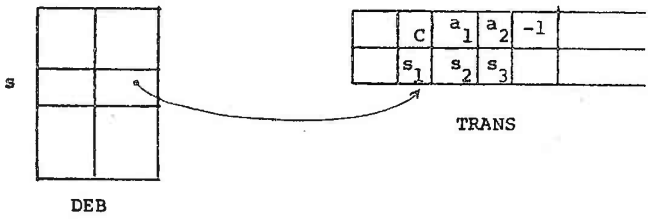
Nous aurons la liste : Ls = x → x → x → nil

(C, S_1) (a_1, S_2) (a_2, S_3)

.....

Nous représenterons ces listes en FØRTRAN par deux tableaux DEB (nétat,2) et TRANS (ntrans, 2). DEB (s , sens) est l'indice dans TRANS de la première transition d'origine s dans l'automate $\mathcal{A}$ si sens vaut 2 sinon dans l'automate $\mathcal{A}'$.

Exemple :



D.5 Algorithmes des procédures

1) ENRACINEMENT (CONTEX, IC, T1, NBE, FINI)

/ ⊠ on va effectuer une séquence d'enracinement sur CØNTEX (⊠, IC).
Si l'enracinement conduit à un arbre syntaxique complet, FINI est mis à VRAI.
Le résultat consiste en NBE état origines de transitions de lecture et rangés dans T1
T1 (1, n, SENS) est un état (n⩽ NBE)
T2 (2, n, SENS) est le pointeur vers CONTEX ⊠ /
DEPILER (CØNTEX (1, IC), Sg)
DEPILER (CØNTEX (5,IC), Sd)
/ ⊠ on affecte à Sg et Sd les sommets de pile des états ⊠ /
cas (1, 2, 3, 4, 5)

1 { Sg = fin analyse droite-gauche
  { Sd = fin analyse gauche-droite
REDUCTION GLØBALE

2 { Sg = fin analyse droite-gauche
  SENS = 2 ; S = Sd
MØDIFICATION

3 { Sg = 0
  { Sd = 0 / ⊠ fin d'analyse générale, on a reconnu ⊢ <axiome> ⊣ ⊠ /
FINI = VRAI

.....

4 { Sd = fin analyse gauche-droite
  { SENS = 1 ; S = Sg
  MØDIFICATIØN

5 { autre cas
  CHOIX DU SENS
  MØDIFICATION

sac
FIN

REDUCTION GLØBALE

Pour chaque couple (S_1, S_2) appartenant à REDUCG (Sg, Sd) faire
/ ⊠ REDUCG (Sg, Sd) donne la liste des états initiaux ainsi que la liste des états à empiler préalablement après une réduction globale ⊠ /
/ ⊠ on recopie l'arbre ⊠ /
ICTL = ICTL + 1
CØNTEX (⊠, ICTL) = CØNTEX (⊠, IC)
/ ⊠ on initialise les deux piles, soit S l'état à empiler k fois ⊠ /
Pour I de 1 à K_g répéter ; EMPILER (CØNTEX (1, ICTL), S); fait
Pour I de 1 à K_d répéter ; EMPILER (CØNTEX (5, ICTL), S); fait
EMPILER (CØNTEX (1, ICTL), S_1)
EMPILER (CØNTEX (5, ICTL), S_2)
/ ⊠ on enracine le nouvel arbre ⊠ /
I = ICTL
ENRACINEMENT (CØNTEX, I, T1, NBE, FINI)
si FINI alors RETOUR fsi
fait

MODIFICATION

/ ⊠ Ce module travaille sur un côté de l'arbre. Il effectue une séquence d'enracinement jusqu'à obtenir un état de lecture sur le sommet de la pile.
Si un état ambigu apparaît au sommet, il y a recopie de l'arbre et chaque possibilité est traitée ⊠ /

si (S est un état de lecture) alors

.....

```
NBE (SENS) = NBE (SENS) + 1
T1 (1, NBE (SENS), SENS) = S
T1 (2, NBE (SENS), SENS) = IC
/ ¤ la procédure s'arrête pour cet arbre ¤ /
sinon si (S est un état de réduction) alors
        / ¤ il faut réduire par la règle de nom
        A = NØM (S) et de longueur L = LREGLE (S) ¤ /
        I = IC
        REDUCTION
        ENRACINEMENT (CONTEX, I, T1, NBE, FINI)
        Si FINI alors RETOUR fsi
        sinon si ambiguité = lecture_réduction alors
        NBE (SENS) = NBE (SENS) + 1
        T1 (1, NBE (SENS), SENS) = S
        T1 (2, NBE (SENS), SENS) = IC
        fsi
        Pour chaque possibilité de réduction répéter
        / ¤ soient A et L les noms et longueurs de la
        règle courante dans l'ensemble des règles en conflit ¤ /
        ICTL = ICTL + 1 ; I = ICTL
        CØNTEX (¤, ICTL) = CØNTEX (¤, IC)
        REDUCTION
        ENRACINEMENT (CØNTEX, I, T1, NBE, FINI)
        si FINI alors RETOUR fsi
        fait

    fsi
fsi
```

REDUCTION

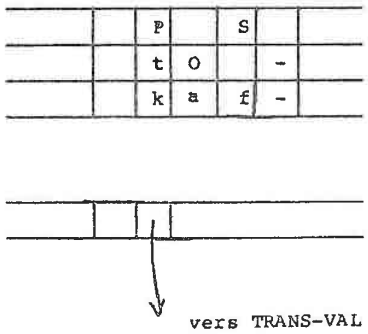
/ ¤ on effectue une réduction latérale sur l'arbre d'indice I situé dans le tableau CONTEX.
L est la longueur de la règle,

.....


```
A est le nom du membre de gauche ¤ /
pour K de 1 à L répéter
si SENS = 1 alors  DEPILER (CONTEX (1,I))
sinon DEPILER (CONTEX (5, I)) fsi
fait
CALCUL du nouvel état S2 tel que
(sommet de pile, A, S2) soit une transition de
l'automate correspondant
si SENS = 1 alors EMPILER (CØNTEX (1, I), S2)
sinon EMPILER (CØNTEX (5, I), S2) fsi
```

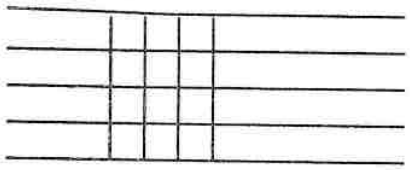
2) LEXICØ

A partir des états origines de transitions de lecture rangés dans T1, cette procédure détermine ces transitions et cherche à les valider grâce au reconnaisseur de mots. Au préalable, LEXICØ consulte la table TRANS-VAL pour vérifier si lors d'un retour arrière, les transitions de même origine n'avaient pas été validées. L'organisation de la table TRANS-VAL est décrite en figure III.5.
Les transitions validées grâce au reconnaisseur de mots sont rangées dans TAB et TRANS-VAL si le taux de reconnaissance du terminal est suffisament élevé.
La variable BASE donne le nombre de transitions validées. La structure du tableau TAB est illustrée par la figure III.6.



.....

depuis ADRESSE-TRANS-VAL



TRANS-VAL

TRANS-VAL (i, 1) = état d'origine

TRANS-VAL (i, 2) = taux de reconnaissance du terminal

TRANS-VAL (i, 3) = rang du dernier phonème

TRANS-VAL (i, 4) = pointeur vers TRANS

figure III.5

terminal	taux de reconnaissance	rang du dernier phonème	rang du premier phonème	pointeur vers trans	pointeur vers conte

figure III.6

Algorithme

LEXICØ (T_1, NBE, TAB, BASE, CONTEX)

BASE = 0

pour k de 1 à NBE (SENS) faire

S = T_1 (1, K, SENS)

IC = T_1 (2, K, SENS)

si SENS = 1 alors P = CONTEX (2, IC) - 1 sinon

P = CONTEX (6, IC) + 1 fsi

/ ◙ p est le rang du premier phonème à prendre en compte ◙ /

.....

RECHERCHE dans TRANS-VAL à partir de ADRESSE-TRANS-VAL (P) des transitions d'origine S, soit L cette sous-liste.

si L ≠ Ø alors RANGER chaque élément de L dans TAB

RETOUR

fsi

Pour chaque transition (s1, s2) telle que s2 soit un terminal faire

On essaye de reconnaître s2 à partir de P dans le treillis par un appel à RECØN (s2). Nous obtenons NT réalisations dont les taux de reconnaissance sont supérieurs à un seuil donné. Ces réalisations sont rangées dans RESULT.

/ ◙ RESULT (N, 1) = n° du terminal reconnu

RESULT (N, 2) = taux de reconnaissance

RESULT (N,3) = rang du dernier phonème pris en compte

RESULT (N, 4) = rang du premier phonème (on s'autorise une marge de ± 1 phonème pour ne pas être gêné par les altérations phonologiques). ◙ /

si nt ≠ 1 alors début 3

Pour chaque réalisation i faire

BASE = BASE + 1

RANGER la transition dans TAB

si RESULT (i, 2) ⩾ 0.8 alors RANGER la transition dans TRANS-VAL fsi

fait

fin 3 fsi

fait fait

FIN LEXICØ

3) MISE A JOUR

A partir de chaque arborescence et chaque transition validée, on met à jour CØNTEX

.....

On empile l'extrémité de la transition validée sur la pile des états. On calcule la plausibilité de la nouvelle arborescence en fonction des plausibilités de l'ancienne, du terminal pris en compte et du nombre de phonèmes ignorés dans la chaîne d'entrée. Cette plausibilité est une heuristique. On met à jour le pointeur phonémique et la liste des terminaux reconnus. Chaque nouvelle arborescence est rangée sur la pile des points de reprise

E) Possibilité de prendre de la liberté vis-à-vis de la syntaxe

Nous entendons par là :

- la possibilité de remplacer certains mots par d'autres proches ne faisant pas partie du vocabulaire de l'application
- la possibilité d'insérer un mot
- la possibilité d'élider un mot

Nous voulons permettre cette liberté pour les trois raisons suivantes :

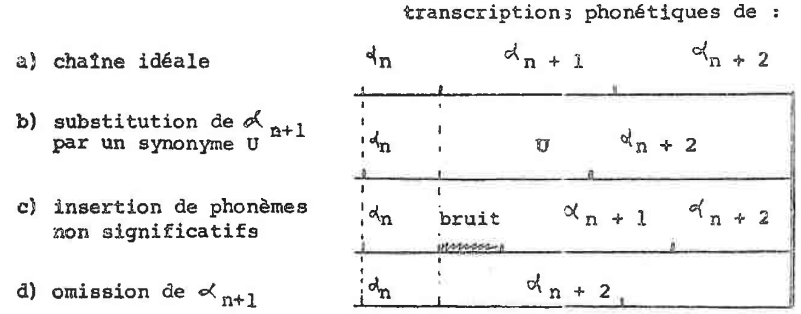
- la syntaxe du message parlée est lâche et bien qu'une adaptation d'un locuteur à un système de reconnaissance automatique soit à envisager, sous forme d'une maîtrise plus grande du langage, il semble nécessaire d'adapter en partie le système de reconnaissance au locuteur.
- le message parlé est redondant, il est possible de comprendre le sens de ce message sans reconnaître tous les mots qui le composent.
- certains mots comme des conjonctions ou préposition ont une existence acoustique incertaine et variable suivant le locuteur, d'autres, comme des interjections, apparaissent aléatoirement et doivent être ignorées.

Concrétement, au niveau de la chaîne phonémique, ces trois altérations se traduisent par :

- des insertions de phonèmes
- des omissions de phonèmes correspondant à la transcription du mot élidé
- des subsitutions de phonèmes

.....

La chaîne phonémique sera formée à partir des quatres types de base :



E.1 La solution proposée

Dans le cas d'une progression de la gauche vers la droite, après la reconnaissance de α_n si la chaîne d'entrée est du type b, c ou d la reconnaissance de α_{n+1} sera impossible dans un voisinage droit proche du dernier phonème pris en compte. Si aucune hypothèse n'a été validée (la base variable BASE de la procédure LEXICØ est alors nulle), une hypothèse d'insertion est émise et LEXICØ est appelé à nouveau. Le module de reconnaissance de mots essaye de localiser alors les différentes hypothèses émises dans la portion du treillis restant à explorer.

Ceci permet de résoudre le problème matérialisé dans le cas C. Pour traiter les cas b et d, il suffit de pouvoir se resynchroniser sur α_{n+2}. A ce niveau là, il faut distinguer deux cas :

- la présence de α_{n+1} est fondamentale pour la compréhension du message.
- l'absence de α_{n+1} est sans importance.

Dans le premier cas, il faut interdire de poursuivre la reconnaissance et dans le deuxième, on peut assumer la présence de α_{n+1}.

Pour apprécier l'importance du terminal, il est possible de partitionner le vocabulaire en deux sous-ensembles : les mots sémantiquement forts et les autres.

.....

c'est ce qui a été fait dans le système MYRTILLE 1 (3) où seul les mots du deuxième ensemble sont assumés. La faiblesse de cette approche vient du fait qu'elle ne tient pas compte du contexte dans lequel doit apparaître α_{n+1}. Nous avons voulu affiner cette notion de contribution d'un mot à la comphréhension en prenant en compte son rôle syntaxique. Dans notre application où syntaxe et sémantique sont liées, nous aurons une idée plus précise de l'importance sémantique de chaque mot. Concrètement cela revient à partitionner l'ensemble des transitions des automates. Les transitions marquées dans le tableau TRANS ne pourront être effectuées que si le terminal correspondan est effectivement présent dans le treillis à l'endroit demandé quant aux terminaux correspondant aux transitions assumées, ils sont rangés dans le tableau TAB avec un taux de reconnaissance qui pénalise la plausabilité de l'arbre.

E.2 Procédure HYPOTHESE

Principe

Cette procédure est appelée à la suite de LEXICØ dans le cas où aucune hypothèse n'a été validée sur un côté des arbres. Dans un premier temps, on tente de localiser toutes les hypothèses dans la portion de treillis restante, afin d'ignorer les phonèmes venus s'insérer entre deux transcriptions. Dans un deuxième temps, toutes les hypothèses sont validées sans progresser dans le treillis avec un taux de reconnaissance arbitraire qui pénalise la plausibilité de l'arbre syntaxique. Ces nouveaux arbres amèneront l'émission du mot α_{n+2} qui permettra, s'il est reconnu, une resynchronisation du processus de reconnaissance ; si ce terminal n'est pas reconnu, l'arbre syntaxique possède deux terminaux non reconnus donc sa plausibilité calculée à partir de ces feuilles est trop faible, ce qui provoque son abandon .
La plausibilité d'un arbre est évaluée à l'aide de deux indices de similitude locale analogues à l'indice de similitude locale d'un chemin (cf. le chapitre concernant l'algorithme de reconnaissance de mots).

.....

L'indice associé au côté droit (respectivement gauche) de l'arbre évalue la similitude entre les derniers terminaux reconnus à droite (respectivement à gauche) et les hypothèses émises à droite (respectivement à gauche) par l'analyseur syntaxique.
Soient r_n un arbre syntaxique et $r_{n+1} = r_n \cdot \alpha$, l'arbre syntaxique obtenu en ajoutant à droite de r_n le terminal α.
Appelons $P_d(r)$ l'indice de similitude droit de r, nous définissons P_d de la façon suivante :

$$P_d(r_o) = SC(\alpha_o) \text{ et } r_o = \alpha_o$$

$$P_d(r_{n+1}) = 0.6\ P_d(r_n) + 0.4\ SC(\alpha)$$

$SC(\alpha)$ est le score de reconnaissance du mot α calculé par l'algorithme de vérification d'hypothèses.
La relation $r_o = \alpha_o$ signifie que l'arbre syntaxique est réduit au premier terminal localisé.

Algorithme

```
MOTCLE = VRAI
appel LEXICØ (T1, NBE, TAB, BASE, CØNTEX)
/ ¤ la variable MOTCLE est utilisé par l'algorithme
de reconnaissance de mot pour tenter la localisation
des hypothèses émises dans la partie du treillis non
encore analysée ¤ /
MOTCLE = FAUX
si BASE = 0 alors début 1
/ ¤ il faut supposer une élision de mot ¤ /
AJOUTER dans TAB les hypothèses qui peuvent s'élider.
Le taux de reconnaissance de chaque terminal est nul.
fin 1
```

FIN HYPOTHESE

.....

F) Algorithme du système de reconnaissance

Principe "meilleur d'abord"

```
INITIA
/ ⊠ fait une recherche de mots clé et initialise la zone des
points de reprise ⊠ /
MOTCLE = FAUX
FINI   = FAUX
début : si la zone des points de reprise est non vide alors
        REPRIS (CONTEX)
        / ⊠ on choisit "le meilleur" point de reprise ⊠ /
        IC = 1 ; ICTL = 1 ; NBE = 0
        / ⊠ ICTL est le pointeur vers le dernier arbre mis dans
        CØNTEX, NBE est le nombre d'arbre mis dans T1 ⊠ /
        ENRACINEMENT  (CØNTEX, IC, T1, NBE, FINI)
        si (FINI = VRAI) alors / ⊠ la reconnaissance est finie ⊠ /
        sinon pour sens de 1 à 2 faire
        LEXICØ (T1, NBE, TAB, BASE,CONTEX)
        si BASE = 0 alors
        HYPOTHESE (TAB, BASE, CØNTEX, T1, NBE) fsi
        MISE A JOUR (TAB, BASE, CØNTEX)
             fait
        aller à début
        fsi
      sinon
    / ⊠ à ce niveau là de l'algorithme, la zone des points de
        reprise est vide et il faut se brancher à une procédure
        de dialogue ⊠ / fsi
        FIN
```

G) Approche du déterminisme

Il est impossible d'arriver à un déterminisme de l'analyse à cause de la nature même des données, mais il est utile d'éliminer les ambiguités syntaxiques dues à la grammaire et au fonctionnement de l'analyseur. En limitant le nombre de retours arrière, la réponse de l'analyseur sera plus rapide.
Nous allons donner les conditions à imposer à la grammaire pour que la zone des points de reprise ne contiennne pas de contextes syntaxiquement ambigus.

.....

- condition 1

Il faut, pour que le processus d'analyse soit borné, qu'il n'y ait pas de séquence infinie d'enracinement. Il ne faut pas de règle $A \overset{\boxtimes}{\rightarrow} A$ (AS3).

- condition 2

Il faut que chaque analyse latérale soit LR (1), ceci se produit quand les grammaires G et $\check{G}$ sont LR (1) simultanément mais cette condition est difficile à réaliser.

- condition 3

Il faut qu'à la suite d'une réduction globale ou du début de l'analyse, un seul arbre soit validé ceci implique que la condition donnée en (AS7 , page 5) soit satisfaite. Cette condition formalise le fait que pour qu'un seul arbre soit validé après une émission par les modules INITIA et REDUCTIØN GLØBALE, il faut que les couples de terminaux à reconnaître à chaque extrémité soient différents.

Exemple :

La grammaire donnée en D.1 vérifie ces trois conditions, par contre, la grammaire LR (1) suivante qui décrit le même langage ne vérifie pas la troisième condition.

X ⟶ a Bc B /1/

X ⟶ d Ca /2/

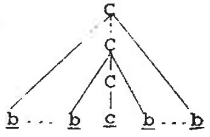
B ⟶ bB /3/

B ⟶ b /4/

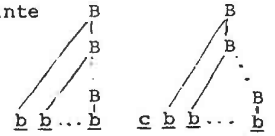
C ⟶ bCb /5/

C ⟶ c /6/

En effet, après avoir reconnu c, on ne sait pas s'il faut enraciner les b voisins de la façon suivante :



ou de la façon suivante



et le nombre de test à faire de chaque côté peut être arbitrairement grand.

.....

Nous ne savons pas si ces trois conditions déterminent un sous-ensemble strict de l'ensemble des langages déterministes donc, nous ne savons pas s'il existe des règles de transformation de grammaire qui permettent d'obtenir, à partir d'un langage déterministe donné, une grammaire vérifiant ces trois conditions.

Il faut signaler les travaux de R. MOHR (AS10) sur le déterminisme des algorithmes d'analyse syntaxique descendant débutant n'importe où.

Il faut aussi rappeler que dans certains cas, cela nous amènera à nous appuyer sur des terminaux dont l'existence acoustique est incertaine, il faut donc traiter particulièrement chaque langage d'application en essayant de dégager "le meilleur compromis" vis-à-vis de ces critères. C'est pour cela qu'une structure de donnée mémorisant les résultats intermédiaires de l'analyse peut permettre d'accélérer le processus de reconnaissance, de même que des mécanismes de pondérations d'hypothèses par les niveaux prosodiques et pragmatiques.

H) Situation de la méthode par rapport aux méthodes ascendantes et descendantes

Cet algorithme n'est pas véritablement ascendant car on y rencontre un aspect de prédiction. Dans l'analyse ascendante toutes des décisions sont reportées afin d'accumuler un maximum de contexte matérialisé par le mot des racines alors que le module REDUCTION-GLOBALE crée autant d'arbres qu'il y a de possibilités d'analyses latérales. Une méthode véritablement ascendante n'aurait construit qu'un seul arbre pour choisir plus tard le couple d'analyses latérales à terminer. Nous n'avons pas utilisé ce principe de construction à cause de sa complexité.

Cet algorithme n'est pas descendant à cause du mode de construction des arbres.

Un intérêt de la méthode d'analyse choisie est de permettre d'utiliser l'aspect prédictif des procédures INITIA et REDUCTION GLOBALE pour y mettre en oeuvre des informations d'autres niveaux afin de pondérer les arbres émis en hypothèse.

.....

Un autre intérêt se trouve dans le mode de représentation des automates qui permet une émission aisée d'hypothèses. Un état est en fait une adresse d'un ensemble d'hypothèses et il n'y a pas de création d'ensemble d'hypothèses comme dans MYRTILLE 1. De plus on peut tenir compte du rôle syntaxique de chaque terminal (§E). Le chapitre suivant donnera des renseignements sur l'encombrement et la rapidité de cette méthode.

CHAPITRE IV

APPLICATION A UN LANGAGE DONNE

I - Introduction

2 - Définition du langage d'application

2.1 Les raisons du choix

2.2. Définition syntaxique

2.3 Définition lexicale

3 - Caractérisation de l'univers d'application : informations pragmatiques

3.1 Particularités dans la forme des phrases

3.2 Mise en oeuvre

4 - Exemples

5 - Conclusion

1) INTRODUCTION

Dans ce chapître, nous allons décrire l'implantation dans un système de reconnaissance des algorithmes d'analyse syntaxique et de localisation de mots. Ces deux algorithmes constituent le noyau d'un système de reconnaissance paramétré par le réseau récursif définissant la grammaire et par le vocabulaire du langage d'application.

De plus l'expérience a montré que la prise en compte d'informations pragmatiques, c'est-à-dire propres à l'application considérée, permettait une reconnaissance plus rapide en limitant le nombre d'hypothèses en certains points, et plus robuste en augmentant le nombre de points d'ancrage.

Le système actuellement implanté effectue une reconnaissance, non pas à partir d'un seul point mais à partir de plusieurs terminaux bien reconnus.

Dans ce chapître nous définirons tout d'abord le langage de l'application, en justifiant notre choix, puis nous mettrons en évidence les particularités relatives à la syntaxe et au vocabulaire de l'application qui constituent la source d'information pragmatique, enfin nous décrirons les adaptations apportées aux algortihmes afin qu'ils utilisent cette source d'information.

2) Définition du langage

Nous avons choisi le langage qui a servi à expérimenter le système MYRTILLE 1 (3). Ce langage est formé des demandes adressées à une standardiste comme par exemple : "allo, bonjour madame, je voudrais avoir le poste 339, S.V.P.".

La figure IV.1 représente l'ensemble des demandes. En réalité, cet ensemble est plus important car, comme nous l'avons vu dans le chapitre II, le système peut émettre des hypothèses d'élision, substitutions ou insertions de mots.

2.1 Les raisons de ce choix

Le but de ce travail est de présenter des outils et des stratégies d'analyse syntaxique adaptés à la reconnaissance automatique du discours continu. Nous utiliserons ces outils dans le système MYRTILLE II en cours de conception pour la compréhension de langages pseudo-naturels. En attendant la réalisation effective de ce système, il nous fallait tester notre approche à l'aide d'un langage d'essai. Nous n'avons pas voulu rechercher et définir un nouveau langage d'application car dans notre cas, la définition d'un langage de communication homme-machine comprend trois étapes difficiles :

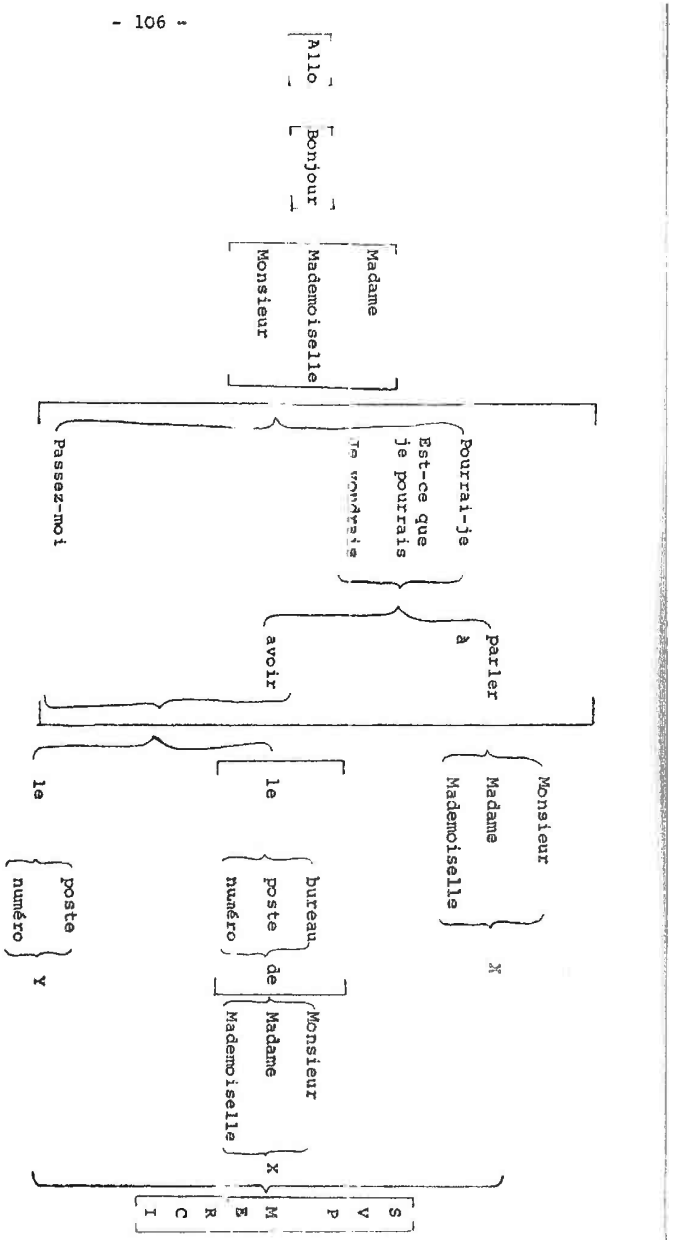
- recensement des requêtes posées à la machine ;
- définition d'une grammaire adéquate qui engendre un ensemble de demandes aussi proche possible de l'ensemble déterminé dans la première étape ;
- définition du lexique phonétique et phonologique associé au vocabulaire.

C'est pour ces raisons que nous avons repris le langage de communication avec un standard téléphonique (3).
Ce choix présente quand même un inconvénient. Le langage adopté est un langage régulier pour l'analyse duquel un analyseur à pile est un outil trop évolué. Nous avons tourné cet inconvénient en compliquant la grammaire jusqu'à la rendre LR (2) pour nous placer dans des conditions analogues à celles rencontrées lors de la reconnaissance de langages plus vastes.

.....

{ } indique un choix possible entre plusieurs alternatives

[ ] indique la possibilité d'absence du groupe de mots correspondant



Aperçu général du langage utilisé

Figure IV.1

Figure IV.2

Grammaire du langage

<AXIOME>	: : =	⊢ <PHRASE> ⊣
<PHRASE>	: : =	<APPEL> <DEMANDE> <DEMANDE>
<APPEL>	: : =	Allo <SALUT> allo <SALUT>
<SALUT>	: : =	Bonjour <TYPE> Bonjour <TYPE>
<DEMANDE>	: : =	<REQUETE> <POLI> <REQUETE>
<POLI>	: : =	s'il vous plaît Merci s'il vous plaît Merci
<REQUETE>	: : =	<I N T> <AFFIR>
<INT>	: : =	Pourrais-je <DEM> est-ce que je pourrais <DEM>
<AFFIR>	: : =	Je voudrais <DEM> passez-moi<OBJET> <OBJET>
<DEM>	: : =	Parler à <PERS> avoir <OBJET>
<PERS>	: : =	<TYPE> <NOM>
<OBJET>	: : =	le <QUALIF> de <PERS> <PERS> le <NUM> le <QUAL 1> <NUM>
<QUALIF>	: : =	Bureau <QUAL 1>
<QUAL 1>	: : =	Poste numéro
<TYPE>	: : =	Monsieur / Madame / Mademoiselle
<NOM>	: : =	Albert / Dupont / Durand / Pierre / Pierrel
<NUM>	: : =	221 / 222 / 223 / 239 / 240 / 241

2.2. Définition syntaxique

Le langage utilisé comporte des phrases à structure assez souple dont la syntaxe de base est donnée par une grammaire à contexte libre. Cette grammaire est représentée dans le formalisme de Backus-Naur sur la figure IV.2.

Il faut considérer la grammaire comme une donnée permettant de guider la reconnaissance et non pas comme une définition rigoureuse du langage.

La figure IV.1 permet d'évaluer le facteur de branchement du réseau associé à cette grammaire. Ce facteur de branchement varie entre 2 et 11. En réalité un nombre supérieur d'hypothèses pourra être émis par l'analyseur lorsque la requête du locuteur ne fait pas partie du langage. En effet, lorsqu'il est impossible de reconnaître un mot à la suite du contexte déjà analysé, l'analyseur augmente l'ensemble H1 des hypothèses émises et momentanément infirmées, de l'ensemble H 2 des hypothèses à émettre à l'étape suivante de l'analyse après avoir supposé chaque mot de H1 reconnu. Si le facteur de branchement du réseau est n alors en prenant des libertés avec la syntaxe il devient égal à $n^2 + n$.

Dans notre application, l'analyseur peut construire ainsi en début de phrase un ensemble d'hypothèses comprenant tous les mots du langage exceptés S.V.P. et merci. Le facteur de branchement est alors égal à 28, mais sa valeur moyenne est de l'ordre de 10.

2.3. Définition lexicale

Pour la reconnaissance des demandes présentées à la machine, le système n'utilise que le lexique formé des terminaux de la grammaire : soit 18 terminaux vrais et 2 terminaux de type lexique, c'est-à-dire à réalisation multiples. Ces derniers regroupent les noms de personne et les numéros de poste téléphonique inscrits dans l'annuaire du standard. Le vocabulaire ainsi utilisé comporte 31 mots ou groupes de mots donnés dans le tableau ci-dessous.

(La codification de ces différents mots est donnée entre parenthèses).

a) les terminaux vrais de la grammaire noyau

(1)	ALLO	(10)	AVOIR
(2)	BONJOUR	(11)	LE
(3)	S'IL VOUS PLAIT	(12)	DE
(4)	MERCI	(13)	BUREAU
(5)	POURRAIS-JE	(14)	POSTE
(6)	EST-CE-QUE-JE POURRAIS	(15)	NUMERO
(7)	JE VOUDRAIS	(16)	MONSIEUR
(8)	PASSEZ-MOI	(17)	MADAME
(9)	PARLER A	(18)	MADEMOISELLE

b) les terminaux de type lexique de la grammaire noyau

(19)	NOM	(20)	NUMERO
	ALBERT (220)		221 (230)
	PIERREL (210)		223 (240)
	BOUCHET (190)		339 (250)
	DURAND (200)		340 (260)
			341 (206)

c) les terminateurs

(21) le début de la phrase : $\vdash$

(22) la fin de la phrase : $\dashv$

Ces deux terminaux apparaissent dans la grammaire et leurs réalisations phonétiques dépendent de l'endroit dans le treillis où ils sont recherchés. Leur taux de reconnaissance est calculé de la façon suivante :

I est le rang du phonème à partir duquel ils sont recherchés.

LG est la longueur de la chaîne d'entrée.

SC_{21} = MAX (0., 1 - 0.2 ⊠ I)

SC_{22} = MAX (0., 1 - (LG - I) ⊠ 0.2)

Ce taux a été fixé par essais successifs. Lorsque ce taux était mal calculé, le fonctionnement du système était défaillant aux extrémités du treillis d'entrée ;

.....

. Soit le système était éxagérement sensible aux bruits (voir exemple 3),

. Soit les derniers mots étaient "oubliés".

Contrairement à ce qui se faisait dans le système MYRTILLE I, aucun traitement n'est fait sur les frontières de mots pour prendre en compte les phénomènes de liaison ou de répétition du dernier phonème. Pour nous affranchir de ces problèmes, nous acceptons que les limites des transcriptions de deux mots successifs ne soient pas les mêmes en tolérant un recouvrement ou un décalage. Cette façon de faire introduit des ambiguités supplémentaires dans la reconnaissance comme le montrera l'exemple 5. Dans la poursuite de ce travail, à l'aide de la phonologie, nous aborderons d'une façon plus satisfaisante ce problème de jonction de mots.

3) CARACTERISATION DE L'UNIVERS D'APPLICATION

3.1 Particularités dans la forme des phrases

Une requête posée par une personne au système se compose de trois parties comme le montre la figure IV.1 :

- une partie appel , éventuellement absente,
- une partie demande , toujours présente et qui contient le sens du message,
- une partie politesse , éventuellement absente.

Nous avons pensé, dans un premier temps, effectuer la reconnaissance d'une phrase de la façon suivante :

. reconnaissance de demande

. si la longueur de la partie gauche encore inexplorée du treillis est suffisante alors reconnaissance de appel fsi

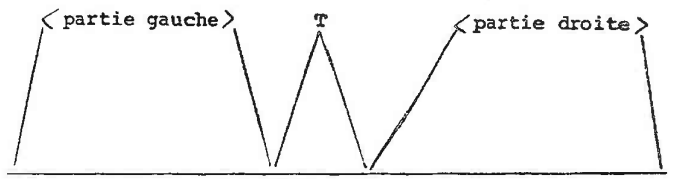
. si la longueur de la partie droite encore inexplorée du treillis est suffisante alors reconnaissance de politesse fsi

Il fallait définir trois grammaires, chacune associée à un sous langage afin d'en paramétrer la procédure MANEGE qui effectue une reconnaissance à partir d'un point quelconque et écrire la procédure d'analyse descendante donnée quelques lignes plus haut.

.....

L'intérêt de cette méthode est dans la multiplication du nombre de points d'ancrage et dans le fait qu'elle donne une vue plus globale de la reconnaissance puisqu'elle fait intervenir les longueurs des portions de treillis non encore analysées ce qui permet d'éliminer certains appels au reconnaisseur de mots. Mais ce découpage de la phrase en trois syntagmes ne tient pas compte des performances du niveau lexical.

En effet, la reconnaissance d'un non terminal T à l'intérieur d'un treillis phonétique aux limites imprécises implique qu'il faille déterminer avec un minimum d'ambiguité le mot sur lequel débutera la reconnaissance et les mots qui auront un rôle de terminateur. Ces terminateurs seront, soit les extrémités de treillis, soit les mots initiaux du non terminal suivant à reconnaître à droite de T, soit les mots finaux du non terminal situé à gauche de T.



treillis phonétique.

Or dans notre application, ces conditions n'étaient pas réunies.

1) la partie demande peut ne pas contenir de mots localisables, sans ambiguité pouvant servir de point de départ

Exemple : Monsieur Mari à l'intérieur de la requête "allo Monsieur, Monsieur Marí, merci".

Il est possible de reconnaître Monsieur trois fois (notamment à la place de merci). Il faut donc envisager une reconnaissance gauche-droite dans certains cas.

2) Les terminaux finaux de < appel > , c'est-à-dire : Monsieur, madame, mademoiselle, allo, bonjour ne possèdent pas des traits acoustiques permettant une localisation sans ambiguité. Dans le treillis, pourtant de bonne qualité, donné figure II.4 on se rend compte que

allo utilisé 1 fois est localisé 7 fois

.....

Finalement la statégie de reconnaissance adoptée est la suivante :

3.2. Mise en oeuvre

a) Recherche de s'il vous plaît

b) si ce terminal est trouvé à partir du phonème k alors

. s'il reste une zone inexplorée à droite alors

recherche de merci fsi

. réduire la longueur du treillis phonétique à k-1

fsi

c) Recherche de la meilleure réalisation d'un des mots suivants

poste

je voudrais

passez moi

est-ce que je pourrais

d) si un terminal est trouvé alors

/ soit ∝ la réalisation de ce terminal /

POINT-DE-DEPART = ∝

sinon POINT-DE-DEPART = début de phrase fsi

e) reconnaissance du groupe appel demande à partir de POINT-DE-DEPART.

Les instructions a, b, c, d, sont exécutées par la procédure INITIA.

Un cas intéressant correspond à un treillis dans lequel sont localisés en premier les terminaux poste et S.V.P. Il reste peu de place pour trouver un nom de personne ou un numéro. On limite ainsi simplement le nombre d'ambiguité.

4) EXEMPLES

Nous allons donner cinq exemples de communication. Le treillis phonétique relatif à chaque communication a été construit artificiellement à partir de la transcription

.....

graphème-phonème de la phrase et des performances des systèmes acoustiques actuels.

Le treillis est en fait une suite de triplets de phonèmes. Chaque triplet représente, à chaque instant, les trois meilleurs choix des algorithmes de classification du niveau acoustique.

La qualité moyenne de chaque treillis est telle que dans un treillis formé de 10 triplets, 7 seulement contiennent le bon phonème, soit un taux de reconnaissance de 70 %.

Deux stratégies de reconnaissance sont essayées
- une reconnaissance de gauche à droite,
- une reconnaissance par îlots de confiance.

Les essais ont été effectués sur l'ordinateur CII-HB IRIS 80 de l'I.U.C.A.L. Le programme complet écrit en FORTRAN IV, effectue avant chaque reconnaissance de phrase la lecture de 3 fichiers-disque :
- la lecture de la grammaire,
- la lecture du lexique,
- la lecture du treillis d'entrée.

Les temps d'accès viennent donc s'ajouter aux temps de reconnaissance proprement dit.

Avant chaque exemple, nous donnons la version écrite de la requête, la suite des codes des terminaux ainsi que le treillis phonétique constitutant la donnée.

Après chaque exemple, la liste des terminaux reconnus à chaque étape est donnée.

Pendant l'éxécution, l'impression du message :
RECHERCHE LEXICALE A PARTIR DU PHONEME n
signifie qu'un appel au module LEXICO a eu lieu. Il y a autant d'impressions du message que d'arbres syntaxiques émis en hypothèse.

Les réponses de l'analyseur lexicographique s'interprètent de la façon suivante :
HYPOTHESES CONFIRMEES
n° du terminal, probabilités, origine et extrémité de la réalisation phonétique

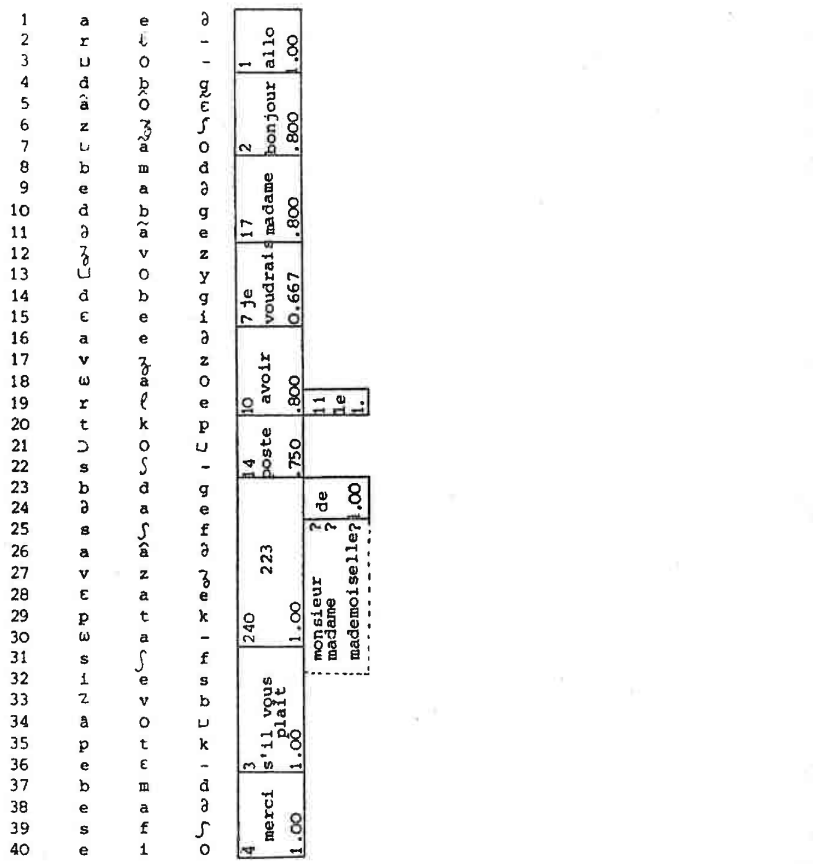
.....

Si un terminal est codé par un entier négatif, cela signifie qu'il a été supposé élidé et que la reconnaissance s'est poursuivie en assumant son absence.

Le mot par lequel débute la reconnaissance est noté deux fois.

Exemple 1

allo, bonjour madame j'voudrais avoir le poste 223 SVP merci
1 2 17 7 10 11 14 240 3 4



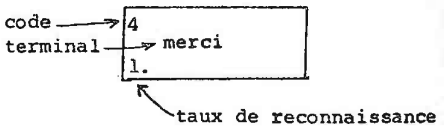


Figure IV.3

Treillis de mots construit par le système.

.....


```
EXECUTION F4LTB VERSION V10-05 01*08*24*

J AI RECONNU   3. AVEC UN SCORE DE  1.00 ENTRE LES PHONEMES 36. 31.
J AI RECONNU   4. AVEC UN SCORE DE  1.00 ENTRE LES PHONEMES 40. 37.

 TERMINAUX RECONNUS
   14  14              début de la reconnaissance sur Poste (noté 14).
 RECHERCHE LEXICALE A PARTIR DU PHONEME 22
+                                          HYPOTHESES DROITES
   20   12
 HYPOTHESES CONFIRMEES
  240.1.000   30.   23.  recherche et reconnaissance du numéro 223
   12.1.000   24.   23.  et du terminal de

 TERMINAUX RECONNUS
   14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME 20
+                                          HYPOTHESES GAUCHES
   11
 HYPOTHESES CONFIRMEES            émission du terminal le  (11)
   11. .500  19.  19.             reconnaissance de le

 TERMINAUX RECONNUS
   14  14  12
 RECHERCHE LEXICALE A PARTIR DU PHONEME 24
+                                          HYPOTHESES DROITES
   16   17   18
 RECHERCHE LEXICALE A PARTIR DU PHONEME 24   tentative de localisation
+                                          HYPOTHESES DROITES
   16   17   18
 HYPOTHESES CONFIRMEES       échec de la reconnaissance
  -16. .000  24.  24.        les terminaux 16, 17, 18 sont
  -17. .000  24.  24.        supposés élidés
  -18. .000  24.  24.

 TERMINAUX RECONNUS
   11  14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME 19
+                                          HYPOTHESES GAUCHES
   10
 HYPOTHESES CONFIRMEES
   10. .800  16.  19.

 TERMINAUX RECONNUS


   10  11  14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME 16
+                                          HYPOTHESES GAUCHES
     5    6    7
 HYPOTHESES CONFIRMEES
    7. .667  12.  15.

 TERMINAUX RECONNUS
    7  10  11  14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME 12
+                                          HYPOTHESES GAUCHES
   21   16    1   17   18    2
 HYPOTHESES CONFIRMEES
   21. .000   1.  11.
   17. .800   8.  11.
    2. .800   4.   7.

 TERMINAUX RECONNUS                                          ....
```

```
  1-  -  10  11  14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME  8
 RECHERCHE LEXICALE A PARTIR DU PHONEME  8
+                                                 HYPOTHESES GAUCHES
   2    1   21
 HYPOTHESES CONFIRMEES
   2. .600   4.   7.
  21. .000   1.   7.
 TERMINAUX RECONNUS
   2  17   7  10  11  14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME  4
 RECHERCHE LEXICALE A PARTIR DU PHONEME  4
+                                                 HYPOTHESES GAUCHES
   1   21
 HYPOTHESES CONFIRMEES
   1.1.000   1.   3.
  21. .400   1.   3.

 TERMINAUX RECONNUS
   1   2  17   7  10  11  14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME  1
+                                                 HYPOTHESES GAUCHES
  21
 HYPOTHESES CONFIRMEES
  21. .800   1.   1.
 TERMINAUX RECONNUS
  21   1   2  17   7  10  11  14  14 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME 30
+                                                 HYPOTHESES DROITES
  22
 HYPOTHESES CONFIRMEES
  22.1.000  31.  31.

 TERMINAUX RECONNUS
  21   1   2  17   7  10  11  14  14 240  22

 PHRASE RECONNUE AVEC UNE PLAUSIBILITE DE  .87
*STOP* 0
*************************************************************************
STPTIME VIRT REL ACT TRA CRM  FAULTS   INS   OUTS   RATE-P/S UC-E RL-E
 0.030   60   44  38  03  31      40    00     00       22.8 1.76 1.78

T537    STEP  WAIT  CORE    END     DISC     TIME    START  CODE  PROGRAM

         01   00.01  34   01*08*28   00      00.03 01*08*23 TERM MHATON11-OBJE
****************************************************************************
****************************************************************************
    TIME      TIME*CORE      CORE-USE   TIME*DISC     DISC-USE
   00.03        01.38           71%       00.92          00%
     DS          DS           SHR        SHR        CR        CP        LP
  I/O-BYTES I/O-CALLS     I/O-BYTES I/O-CALLS     CARDS     CARDS     PAGES
    90828       45           00         00         00        00        03
RTEN JOB TERMINATED     MARJ....X512 HATO  01*08*28*
```

(annotations manuscrites : « échec de la reconnaissance du début de phrase à partir du phonème 8 » ; « reconnaissance du début de phrase »)

échec de la reconnaissance du début de phrase à partir du phonème 8

reconnaissance du début de phrase

....


```
 14
 14 240
 14  12
 14  14 240
 11  14  14 240
 10  11  14  14 240
  7  10  11  14  14 240
 17   7  10  11  14  14 240
  2  17   7  10  11  14  14 240
  1   2  17   7  10  11  14  14 240
  1   2  17   7  10  11  14  14 240  22
```

Schéma de la progression de la reconnaissance.

Remarques

En comparant la figure IV.3- à la figure II.4 on remarque le filtrage important réalisé par l'analyseur syntaxique sur le treillis de mot.

Il n'y a qu'un seul retour arrière dans la reconnaissance. A la troisième étape, le système tente la reconnaissance de monsieur (16), madame (17), mademoiselle (18) à la suite de de (12). Dans un premier appel au module LEXICØ, le système essaye de reconnaître ces terminaux immédiatement à droite de de. Comme aucune hypothèse n'a été confirmée, un deuxième appel a lieu et le module de reconnaissance de mot tente de localiser ces terminaux entre les phonèmes 24 et 30 (puisque l'on a reconnu S.V.P. à partir du phonème 31). Après ce deuxième échec, une hypothèse d'élision a lieu mais la probabilité de cette phrase n'est plus la meilleure ce qui provoque un retour arrière selon la stratégie "meilleur d'abord" que nous avons implantée.

Le temps d'éxécution est de 0,03 minute, soit 1,8 seconde mais il faut déduire les temps d'entrée-sortie (écriture des résultats et lecture des données).

La taille mémoire nécessaire est de 34 pages soit 17 k mots.

.....

Exemple 2

Il s'agit de la reconnaissance du même treillis amputé des transcriptions des terminaux S.V.P. et merci, mais selon une stratégie différente. La reconnaissance s'effectue de gauche à droite, le premier terminal pris en compte est donc 21. Le système de reconnaissance est le même dans les deux stratégies et le point de départ de l'analyse est un paramètre de la procédure INITIA.

Dans cet exemple, le mot poste (14) a été trouvé en même temps que 223 (240) mais avec un score inférieur, il n'est donc pas pris en compte.

Notons aussi qu'avec cette stratégie, le système a construit la phrase "allo, bonjour, madame Albert" qui

1 2 17 20

n'apparaît pas dans la reconnaissance par îlots de confiance.

Le temps d'exécution et l'occupation mémoire sont comparables dans ces deux exemples mais la méthode gauche-droite n'effectue pas la recherche de S.V.P. merci.

Sur cet exemple la recherche par îlots de confiance est donc plus complète, plus rapide, et diminue le nombre d'ambiguités de reconnaissance.

.....


```
     JOB ACTIVATED    . RECON...X542 HATO  02*05*47*
]JOB,T RECON...X542,HATO,GEHAAAJM,30
]EXEC RECON,%ESSAI=ESSAI14-JM
*!LIMIT (CORE,60)
*!ASSIGN 100,FIL,(STS,OLD),(NAM,MHATON11-GRAMM)
*!ASSIGN 101,FIL,(STS,OLD),(NAM,MHATON11-DONNE)
*!ASSIGN 102,FIL,(STS,OLD),(NAM,ESSAI14-JM)
*!RUN (LNN,MHATON11-OBJET1)
EXECUTION F4LIB VERSION V10-05 02*05*49*


 TERMINAUX RECONNUS
   21  21
 RECHERCHE LEXICALE A PARTIR DU PHONEME  1
+                                          HYPOTHESES DROITES
    1   16   17   18    2    5    6    7    8   11
 HYPOTHESES CONFIRMEES
   1.1.000   3.   1.
  11. .500   2.   2.

 TERMINAUX RECONNUS
   21  21   1
 RECHERCHE LEXICALE A PARTIR DU PHONEME  3
 RECHERCHE LEXICALE A PARTIR DU PHONEME  3
+                                          HYPOTHESES DROITES
    2    5    6    7   11   16   17   18
 HYPOTHESES CONFIRMEES
    2. .800   7.   4.
    7. .500   9.   6.

 TERMINAUX RECONNUS
   21  21   1   2
 RECHERCHE LEXICALE A PARTIR DU PHONEME  7
 RECHERCHE LEXICALE A PARTIR DU PHONEME  7
+                                          HYPOTHESES DROITES
   16   17   18    5    6    7   11
 HYPOTHESES CONFIRMEES
  17. .800  11.   8.
  17. .800  11.   8.

 TERMINAUX RECONNUS
   21  21   1   2  17
 RECHERCHE LEXICALE A PARTIR DU PHONEME 11
+                                          HYPOTHESES DROITES
    5    6    7   11   16   17   18
 HYPOTHESES CONFIRMEES
    7. .667  15.  12.

 TERMINAUX RECONNUS
   21  21   1   2  17
 RECHERCHE LEXICALE A PARTIR DU PHONEME 11
+                                          HYPOTHESES DROITES
   19
 RECHERCHE LEXICALE A PARTIR DU PHONEME 11
+                                          HYPOTHESES DROITES
   19
 HYPOTHESES CONFIRMEES
  220. .667  19.  18.

 TERMINAUX RECONNUS
   21  21   1   2  17   7
 RECHERCHE LEXICALE A PARTIR DU PHONEME 15
+                                          HYPOTHESES DROITES
    9   10
 HYPOTHESES CONFIRMEES
   10. .800  19.  16.                                        .....
```

```
 TERMINAUX RECONNUS
   21  21   1   2  17   7  10
 RECHERCHE LEXICALE A PARTIR DU PHONEME 19


+                                          HYPOTHESES DROITES
   11   16   17   18
 HYPOTHESES CONFIRMEES
   11. .500  20.  19.

 TERMINAUX RECONNUS
   21  21   1   2  17 220
 RECHERCHE LEXICALE A PARTIR DU PHONEME 19
+                                          HYPOTHESES DROITES
   22
 HYPOTHESES CONFIRMEES
   22. .000  31.  20.

 TERMINAUX RECONNUS
   21  21   1   2  17   7  10  11
 RECHERCHE LEXICALE A PARTIR DU PHONEME 20
+                                          HYPOTHESES DROITES
   13   20   14   15
 HYPOTHESES CONFIRMEES
  240. .875  29.  23.
   14. .750  22.  20.

 TERMINAUX RECONNUS
   21  21   1   2  17   7  10  11 240
 RECHERCHE LEXICALE A PARTIR DU PHONEME 29
+                                          HYPOTHESES DROITES
   22
 HYPOTHESES CONFIRMEES
   22. .800  31.  30.

 TERMINAUX RECONNUS
   21  21   1   2  17   7  10  11 240  22

 PHRASE RECONNUE AVEC UNE PLAUSIBILITE DE  .63
*STOP* 0
************************************************************
STPTIME VIRT REL ACT TRA CRM  FAULTS     INS    OUTS     RATE-P/S
  0.030   60  44  38  03  31       37      00      00         20.

T545     STEP  WAIT  CORE     END    DISC     TIME    START  CO
          01   00.01   33   02*05*54   00      00.03 02*05*47 TI
************************************************************
************************************************************
     TIME       TIME*CORE       CORE-USE    TIME*DISC     DISC
     00.03         01.39            70%        00.93
       DS          DS              SHR         SHR          CR
  I/O-BYTES I/O-CALLS      I/O-BYTES I/O-CALLS     CARDS
     90132        44             00          00         00
    JOB TERMINATED      RECON...X542 HATO  02*05*54*
```

```
  21  21
  21  21   1
  21  21   1   2
  21  21   1   2  17
  21  21   1   2  17
  21  21   1   2  17   7
  21  21   1   2  17   7  10
  21  21   1   2  17 220
  21  21   1   2  17   7  10  11
  21  21   1   2  17   7  10  11 240
  21  21   1   2  17   7  10  11 240  22
```


Exemple 3

Il s'agit du même treillis que celui de l'exemple 2 mais 10 phonèmes "de bruit" ont été ajoutés à la fin.

La bonne requête est trouvée mais au prix d'un nombre considérablement accru de retour-arrière.

A la 9è étape la phrase correcte est trouvée mais l'échec de la reconnaissance de la fin de phrase provoque un retour arrière et la prise en compte de faux contextes, qui n'incorporent plus bonjour (2). Cette remise en question de bonjour vient du fait que le système a reconnu simultanément bonjour et je voudrais à cause des limites trop imprécises accordées aux réalisations phonétiques dans le treillis (voir § IV.2.3.)

```
21  21
21  21   1
21  21   1   2
21  21   1   2   7
21  21   1   2   7  10
21  21   1   2   7  10  11
21  21   1   2   7  10  11 240
21  21   1   2   7  10  11  14
21  21   1   2   7  10  11  14 240
21  21   1   2   7  10  11  14  12
21  21   1   7
21  21   1   7  10
21  21   1   7  10  11
21  21   1   7  10  11 230
21  21   1   7  10  11  14
21  21   1   7  10  11  14 240
21  21   1   7  10  11  14  12
21  21  11
21  21   1   2   7  10  11  14  12 -17
21  21   1   2   7  10  11  14  12 -17 200
21  21   1   2   7  10  11  14  12 -16
21  21   1   2   7  10  11  14  12 -16 200
21  21   1   2   7  10  11  14 240  22
```

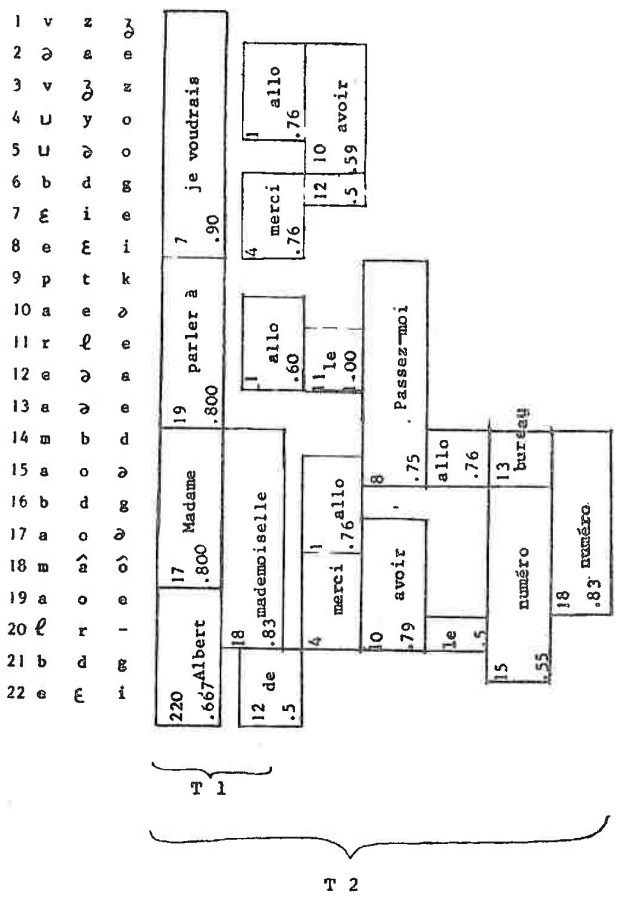
Schéma de la progression de la reconnaissance.

.....

Exemple 4

"j'voudrais parler à madame Albert"

7 9 17 220



T1 = Treillis de mots construit par le système

T2 = Treillis de mots construit sans mettre en oeuvre la source d'information syntaxique.

.....

Remarque

La procédure INITIA localise le groupe je voudrais en début de communication. La reconnaissance s'effectue de gauche à droite.

Le temps de reconnaissance est de 1,2 seconde. Il n'y a qu'un seul retour-arrière après l'échec de la reconnaissance de tous les noms de personne à la suite de mademoiselle. Le faible nombre de retour-arrière valide le choix de la stratégie "meilleur d'abord". Il n'en serait plus de même, vraisemblablement, avec un langage plus vaste et une grammaire ayant un facteur de branchement supérieur.

```
 C10C04      :MON2              16*05*79
RIEN JOB ACTIVATED       MARI....X512 HATO  01*19*27*
]JOB,T MARI....X512,HATO,GEHAAAJM
!LIMIT (CORE,60)
!ASSIGN 100,FIL,(STS,OLD),(NAM,MHATON11-GRAMM)
!ASSIGN 101,FIL,(STS,OLD),(NAM,MHATON11-DONNE)
!ASSIGN 102,FIL,(NAM,ESSAI13-JM),(STS,OLD)
!RUN (LMN,MHATON11-OBJET2)
EXECUTION F4LIB VERSION V10-05 01*19*29*

 3 EST ABSENT

 TERMINAUX RECONNUS
    7   7
 RECHERCHE LEXICALE A PARTIR DU PHONEME  7
+
    9   10                                    HYPOTHESES DROITES
 HYPOTHESES CONFIRMEES
    9. .800  12.   9.

 TERMINAUX RECONNUS
    7   7   9
 RECHERCHE LEXICALE A PARTIR DU PHONEME 12
+
   16   17   18                               HYPOTHESES DROITES
 HYPOTHESES CONFIRMEES
   17. .800  17.  14.
   18. .750  20.  14.

 TERMINAUX RECONNUS
    7   7   9  17
 RECHERCHE LEXICALE A PARTIR DU PHONEME 17
+
   19                                        HYPOTHESES DROITES
 HYPOTHESES CONFIRMEES
  220. .667  20.  19.
```

.....

```
 TERMINAUX RECONNUS
    7   7   9  18
 RECHERCHE LEXICALE A PARTIR DU PHONEME 20
+                                          HYPOTHESES DROITES
   19
 RECHERCHE LEXICALE A PARTIR DU PHONEME 20
+                                          HYPOTHESES DROITES
   19
 HYPOTHESES CONFIRMEES
  -19. .000  20.  20.

 TERMINAUX RECONNUS
    7   7   9  17 220
 RECHERCHE LEXICALE A PARTIR DU PHONEME 20
+                                          HYPOTHESES DROITES
   22
 HYPOTHESES CONFIRMEES
   22. .400  24.  21.

 TERMINAUX RECONNUS
    7   7   9  17 220  22
 RECHERCHE LEXICALE A PARTIR DU PHONEME  1
+                                          HYPOTHESES GAUCHES
   21   16    1   17   18    2
 HYPOTHESES CONFIRMEES
   21. .800   1.   1.

 TERMINAUX RECONNUS
   21   7   7   9  17 220  22

 PHRASE RECONNUE AVEC UNE PLAUSIBILITE DE  .78
*STOP* 0
*************************************************************************
STPTIME VIRT REL ACT TRA CRM  FAULTS    INS   OUTS    RATE-P/S UC-E RL-E
  0.022   60  44  38  03  31      40     00     00        30.9 1.77 1.80

T540    STEP  WAIT  CORE     END    DISC     TIME    START  CODE  PROGRAM
         01   00.01   34   01*19*32   00     00.02 01*19*27 TERM MHATON11-OBJE
*************************************************************************
*************************************************************************
     TIME        TIME*CORE        CORE-USE     TIME*DISC      DISC-USE
     00.02          01.03             70%        00.69           00%
       DS          DS              SHR        SHR          CR          CP          LP
    I/O-BYTES I/O-CALLS      I/O-BYTES I/O-CALLS      CARDS       CARDS       PAGES
       89804        44              00         00         00          00          03
RIEN JOB TERMINATED      MARI....X512 HATO  01*19*32*


    7   7
    7   7   9
    7   7   9  17
    7   7   9  18
    7   7   9  17 220
    7   7   9  17 220  22
   21   7   7   9  17 220  22
```

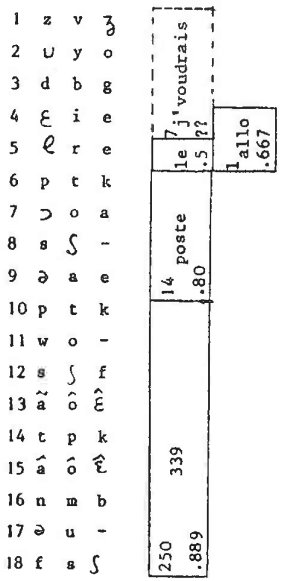
.....

Exemple 5

Phrase prononcée

j'voudrais le poste 339

7 11 14 250



Treillis de mots construit par le système

(Allo est trouvé dans la transcription phonétique de j'voudrais qui n'est pas recherché)

Phrase reconnue

allo le poste 339

1 11 14 250

Remarques

La phrase prononcée n'est pas correcte du point de vue grammatical, la phrase correcte aurait été :

je voudrais avoir le poste 339

.....

Après avoir reconnu le poste 339, le système tente la reconnaissance de avoir (10) et passez-moi (8). Ces deux terminaux sont supposés élidés. Mais il est possible de reconnaître allo, comme le montre le treillis ci-dessus. Par contre, cette requête n'est pas reconnaissable de gauche à droite car le taux de reconnaissance de j'voudrais est trop faible pour admettre une élision du terminal suivant.

Schéma de la reconnaissance

```
 3 EST ABSENT

 TERMINAUX RECONNUS
   14  14
 RECHERCHE LEXICALE A PARTIR DU PHONEME  9
+                                          HYPOTHESES DROITES
   20   12
 HYPOTHESES CONFIRMEES
  250. .889  18.  10.

 TERMINAUX RECONNUS
   14  14 250
 RECHERCHE LEXICALE A PARTIR DU PHONEME  6
+                                          HYPOTHESES GAUCHES
   11
 HYPOTHESES CONFIRMEES
   11. .500   5.   5.

 TERMINAUX RECONNUS
   11  14  14 250
 RECHERCHE LEXICALE A PARTIR DU PHONEME  5
+                                          HYPOTHESES GAUCHES
   10
 RECHERCHE LEXICALE A PARTIR DU PHONEME  5
+                                          HYPOTHESES GAUCHES
   10
 HYPOTHESES CONFIRMEES
  -10. .000   5.   5.

 TERMINAUX RECONNUS
   11  14  14 250
 RECHERCHE LEXICALE A PARTIR DU PHONEME  5
+                                          HYPOTHESES GAUCHES
    8
 RECHERCHE LEXICALE A PARTIR DU PHONEME  5
+                                          HYPOTHESES GAUCHES
    8
 HYPOTHESES CONFIRMEES
   -8. .000   5.   5.
 RECHERCHE LEXICALE A PARTIR DU PHONEME 18
 RECHERCHE LEXICALE A PARTIR DU PHONEME 18
+                                          HYPOTHESES DROITES
    4    3   22
 HYPOTHESES CONFIRMEES
   22. .800  20.  19.
```

....


```
RMINAUX RECONNUS
11  14  14 250  22
CHERCHE LEXICALE A PARTIR DU PHONEME  5
                                          HYPOTHESES GAUCHES
21   16    1   17   18    2
POTHESES CONFIRMEES
21. .200   1.   4.
 1. .667   4.   5.

RMINAUX RECONNUS
 1  11  14  14 250  22
CHERCHE LEXICALE A PARTIR DU PHONEME  4
                                          HYPOTHESES GAUCHES
21


POTHESES CONFIRMEES
21. .400   1.   3.

RMINAUX RECONNUS
21   1  11  14  14 250  22

RASE RECONNUE AVEC UNE PLAUSIBILITE DE  .75
OP* 0
*********************************************************************
TIME VIRT REL ACT TRA CRM  FAULTS    INS   OUTS    RATE-P/S UC-E RL-E
.023   60  44  38  03  31      38     00     00        27.9 1.78 1.80

    STEP  WAIT  CORE     END    DISC     TIME    START  CODE  PROGRAM
      01   00.01   34   01*15*24   00     00.02 01*15*18 TERM MHATON11-OBJE
*********************************************************************
*********************************************************************
 TIME       TIME*CORE       CORE-USE    TIME*DISC     DISC-USE
 00.02         01.08            70%       00.72          00%
   DS         DS            SHR       SHR         CR        CP        LP
I/O-BYTES I/O-CALLS     I/O-BYTES I/O-CALLS     CARDS     CARDS     PAGES
   89804        44            00        00        00        00        03
 JOB TERMINATED     MARI....X512 HATO  01*15*24*


14  14
14  14 250
11  14  14 250
11  14  14 250
11  14  14 250  22
 1  11  14  14 250  22
21   1  11  14  14 250  22
```

.....

5) CONCLUSION

La reconnaissance d'une phrase par îlots de confiance est une stratégie intéressante. Les exemples présentés dans le cas d'un langage artificiel simple donnent une vue partielle de ses qualités et défauts.

Comme qualités :

- assez bonne insensibilité aux bruits dès que l'on a reconnu un îlot consistant de terminaux (exemple 5)
- grande vitesse de reconnaissance (comparaison entre les exemples 1 et 2)
- souplesse d'utilisation puisqu'elle permet aussi une reconnaissance de gauche vers la droite (exemple 4)

Comme défauts :

- les heuristiques de calcul de taux de reconnaissance des îlots sont difficiles à déterminer. Il est en effet difficile de savoir quel est le meilleur îlot
  - . l'un est formé de bons terminaux séparés par des trous,
  - . l'autre est formé de terminaux contigus médiocrement reconnus.
- une place mémoire importante due à l'utilisation de deux réseaux récursifs.

Nous avons été surpris par les temps d'éxécution, ils sont trois fois plus faibles que ceux observés dans la première version de MYRTILLE I (3).

Nous pouvons avancer plusieurs explications :

- les exemples présentés s'exécutent sur un CII IRIS 80, alors que les précédents s'exécutaient sur un CII 10 070 plus ancien et moins rapide.
- la structure de données de l'automate contient les ensembles d'hypothèses correspondants aux différentes étapes de l'analyse. Elle est plus intéressante qu'une

.....

représentation par règles génératives d'une grammaire. Alors que le précédent système MYRTILLE I construisait l'ensemble d'hypothèses à émettre à partir des règles de la grammaire, dans notre version ascendante, ces ensembles sont précompilés. Il en résulte un gain de temps appréciable que l'on retrouve dans des systèmes entièrement précompilés tel que HARPY (33) et qui les rend plus rapides que les autres systèmes.

- L'algorithme de reconnaissance de mots donne de meilleurs résultats lors de la localisation d'un terminal dans une portion de treillis mal délimitée.

- BIBLIOGRAPHIE -

REFERENCES CONCERNANT LES SYSTEMES DE RECONNAISSANCE

DE LA PAROLE

1 D. RAJ REDDY "Speech Recognition by Machine"
Proceeding of the IEEE, April 1976.

2 J.P. HATON "Contribution à l'analyse, la paramétrisation et la reconnaissance automatique de la parole" Thèse d'état, Univ. de Nancy I, janv. 1974.

3 J.M. PIERREL "Contribution à la compréhention automatique du discours continu", thèse de 3ème cycle, Univ. de Nancy I, nov. 1975.

4 D.E. HALL "the linguistic Module, final Report of Speech understanding Systems", MIT Lincoln lab., Lexington.

5 F. JELINEK, L.R. BAHL, and R.L. MERCER
"Design of a linguistic, statistical Decoder for Recognition of Continuous Speech"
IEEE Trans. Inform. Théory, Vol IT-21, PP 250-256, May 1975.

6 C.C. TAPPERT, "Experiments with a Tree-Seard Method converting noisy phonetic Representation into standard orthography"
IEEE Trans. acoust., Speech, signal, Processing, Vol. ASSP-23, PP 129-135, Feb. 1975.

7 L.R. BAHL and F. JELINEK
"Decoding for Channels with Insertions, Deletions, and Substitutions with Applications to Speech Recognition", IEEE Trans. Inform. Theory. Col. IT 21, PP 404-411, 1975.

8 D. FORNEY "the Viterbi Algorithm", Proc. IEEE, Vol 61, n°3, PP 268-278, Mars 1973.

G.M. WHITE "Dynamic Programming, the Viterbi Algorithm and low Cost Speech Recognition", ITT Defense Comm. Division, Proc. IEEE.

9 D.H. KLATT "Word Vérification in a Speech Understanding System", Ed., Speech Recognition : Invited Papers of the IEEE Symp. New-York : Academic Press, 1975.

.....

10 G. GOUARDERES "Organisation d'un lexique en vue de la reconnaissance de la parole en continu" mémoire de doct. Ing. CNAM, Univ. Paul Sabatier, Toulouse, 1977.

" " "Les lexiques automatiques pour le traitement de la parole : version 3 du lexique du projet A.R.I.A.L.", lab. CERFIA, Univ. Paul Sabatier, Toulouse, 1977.

11 G. MALMBERG "La Phonétique", Que sais-je ? N° 637, P.U.E.

12 J.P. HATON, J.M. PIERREL "Data Structures and Architecture of MYRTILLE II Speech Understanding System", 4 th I.J.C.P.R., nov. 7-10, 1978.

13 J.S. BRIDLE "An Efficient Elastic-Template Method for Detecting Given Words in Running Speech", Brit. acoust. Soc. Proc. Tome 2, 1973.

14 J.P. HATON, C. SANCHEZ "the Acoustic Level of MYRTILLE II Speech Understanding System", I.C.A.S.S.P. 1979.

15 R.W. CHRISTIANSEN an C.K. RUSHFORTH "Detecting and Locating Key-Words in continuous Speech using linear predictive Coding".IEEE Trans. on Acoustics, Speech and Signal Processing, Vol ASSP-25, n° 5, oct. 1977.

16 A.S. RABINOWITZ "On the value of the Fano Algorithm in établisching the graphemic Form of Machine derived phonetic Strings", Th D. Thésis, Univ. of North Carolina, Roleigh, March 1972.

17 R. VIVES "Utilisation de l'information phonémique et syllabique pour la reconnaissance de mots prononcés isolement ou dans des phrases" 10e J.E.P., Grenoble, 1979.

18 J.Y. GRESSER et R. VIVES "A Similarity Index Between Strings of Symbols, Application to Automatic Word and Language Recognition", Proc. of the first Intern. Joint conf. on Pattern Recognition, Oct.30 - Nov. 1, 1973, Washington.

19 B. CAUSSE, D. DOURS, R. FACCA, G. PERENNOU "Evaluation d'une méthode ascendante d'analyse lexicale dans le discours continu". 7 J.E.P., Nancy 1976.

.....

20 V.M. VELICHKO and N.G. ZAGORYKO "Automatic Recognition of 200 Words", Int. Journal of Man-Machine Studies, (1970), 2, 223, 235.

21 H. SAKOE and S. CHIBA "A dynamic-programming Approach to continuous Speech Recognition", Proc. IEEE Symposium on Speech Recognition, PP 101, April 1974.

22 G. CAELEN, G. MAURAND "La durée des phonèmes et la reconnaissance des débuts et fins de mots : étude acoustique des structures homophoniques chez deux locuteurs" 9e J.E.P. Lannion 1978.

23 D. SANKOFF "Shorcuts, Diversions, and maximal chains in Partially Ordered Sets", Discrete Math. Vol 4, n°3, March 1973, PP 287-297, North Holland Publisching Company.

24 D. SANKOFF "Matching Sequences under Deletion/Insertion Constraints", Proc. Nat. Acad. Sci., U.S.A., 69 (1972), PP. 4-6.

25 J.H. ELLIS Electronics Letters, vol 5, 335-6, July 1969.

26 J.H. WARREN "A Dynamic Pattern Matching Algorithm with Applications to Automatic Speech Recognition" in Machine Perception of Patterns and Pictures, Institute of Physics, London, 1972.

27 M.F. MEDRESS and all " Word spotting in conversational Speech", IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing, IEEE, CH 1379-7 ASSP, PP 599-602, 1979.

28 H. MATSUMOTO and H. WAKITA "Frequency warping for nonuniform Talker Normalisation", IEEE ASSP, CH 1379-7, PP 566,569, 1979.

29 J.P. HATON et G. PERENNOU "Reconnaissance automatique de la parole", 8e école d'été d'informatique de l'A.F.C.E.T., Namur 10-22 juillet 1978.

30 J.M. PIERREL "MYRTILLE II, un système de compréhension du discours continu"; Publication du C.R.I.N. 78-R-019 Univ. de Nancy I, 1978.

.....

31 N.R. DIXON an H.F. SILVERMAN,
"A General Language-operated Decision Implemention System (GLODIS) : Its Application to continuous Speech Segmentation", IEEE Trans. Acoust. Speech, Signal Processing, vol. ASSP-24, 1976.

32 M. BATES "the Use of Syntax in a Speech Understanding System", IEEE Trans. Acoust. Speech, Signal Processing, vol. ASSU-23, PP 112-117, Feb. 1975.

33 B. LOWERRE "A comparative performance analysis of speech understanding systems" Ph. D. dissertation, Comput. Sci. Dep., Carnegie-Mellon Univ., Pittsburgh, PA, 1976.

REFERENCES CONCERNANT L'ANALYSE SYNTAXIQUE

AS 1 M. GRIFFITHS "LL (1) Grammars and Analysers" Compiler Construction : an advanced Course, Lecture Notes in Computer Science. Springer Verlag

AS 2 J.J. HORNING "LR Grammars and Analysers" Compiler Construction : An advanced Course lecture Notes in Computer Science. Springer Verlag

AS 3 C. PAIR "Structures de données et compilation" Cours de maîtrise d'informatique U.E.R. de mathématiques appliquées, Univ. de Nancy I.

AS 4 W.A. WOODS "Transition Network Grammars for natural Langage Analysis", Commun. Ass. Comput. Mach., vol 13, n° 10, PP 591-602, 1970.

AS 5 P.L. MILLER "A locally organized Parser for spoken Input", Commun. of the ACM, Vol 17, n° 11, 1975.

AS 6 V.A. AHO et J.D. ULLMAN "the Theory of parsing, Translation and compiling". Vol 1 Princite hall 1972
Vol 2 Princite hall 1973.

AS 7 J.P. HATON et R. MOHR "A parsing Algorithm for imperfect Patterns and its Application to Speech and Image Recognition", 3d I.J.C.P.R., San Diego, U.S.A., nov. 1976.

AS 8 P. QUINTON "Utilisation d'un analyseur syntaxique pour la reconnaissance de la parole continue", Annales des télécommunications, tome 32, N° 9-10, septembre-octobre 1977

AS 9 G. VEILLON "Traitement automatique des langues naturelles et communication homme-machine" Support de cours de l'école I.R.I.A. 21-25 nov. 1977.

AS 10 J.P. HATON et R. MOHR "Rapport du contrat sesori 76-039, utilisation des méthodes syntaxiques en reconnaissance de formes" Publication du CRIN n° 77-R-033, 1977.

NOM DE L'ETUDIANT : Monsieur MARI Jean François

NATURE DE LA THESE : Doctorat de Troisième cycle d'INFORMATIQUE

VU, APPROUVE

et PERMIS D'IMPRIMER

NANCY LE 1 [illegible]1979 5087

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

M. BOULANGE