

1125



UN SYSTEME D'ANALYSE DES LANGUES NATURELLES :  
APPLICATION A LA CORRECTION INTERACTIVE DE TEXTES

-----

J. COURTIN

18 Avril 1975

## 1 - PRESENTATION DU SYSTEME :

Ce système, dont le but est l'analyse syntaxique d'une langue naturelle, comprend:

- un système d'analyse morphologique
- un système d'analyse syntaxique

L'articulation entre les différents composants du système est illustrée par le schéma ci-après.

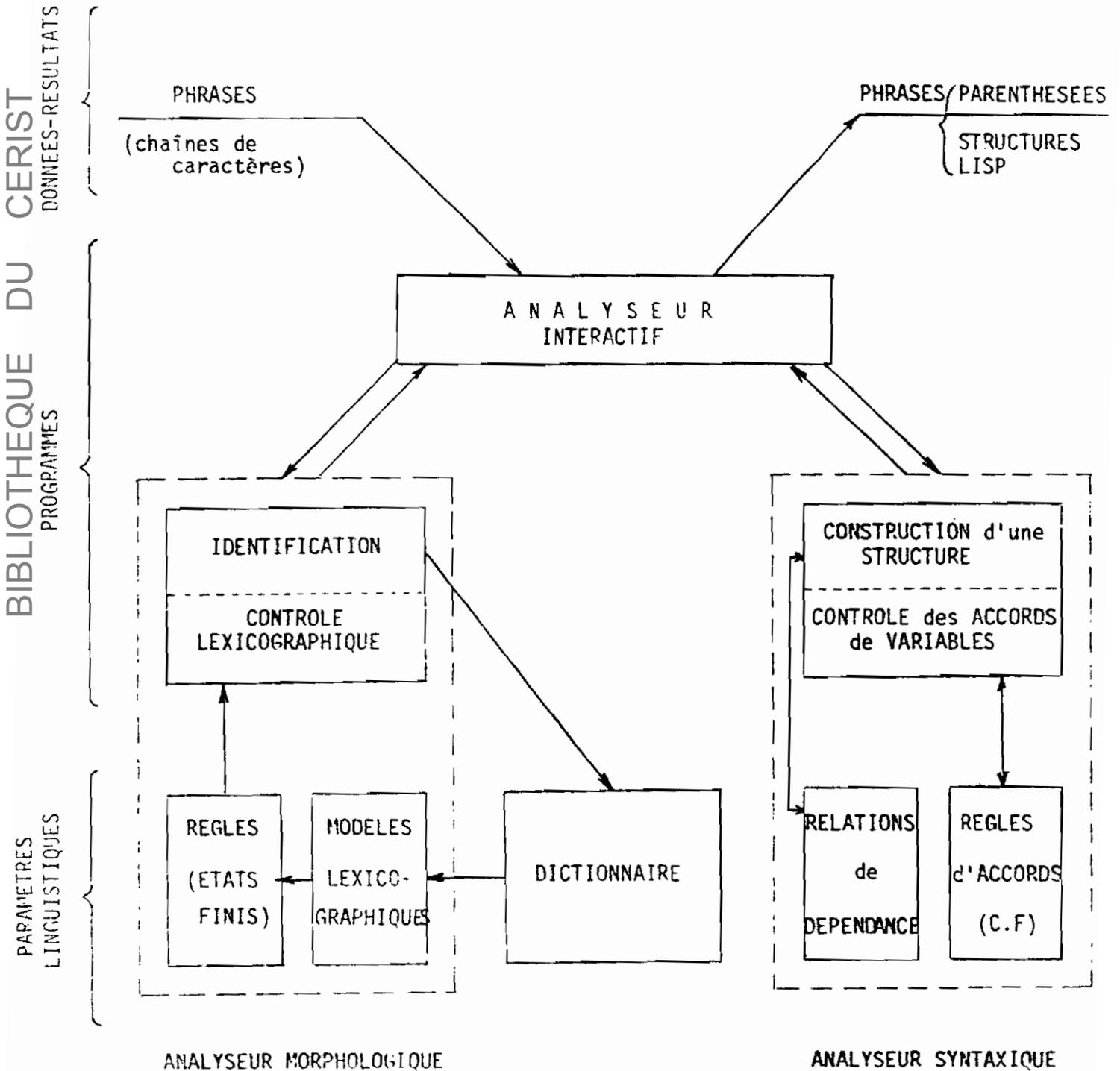
### 1.1 Le système d'analyse morphologique :

La description de ce système se trouvant dans [1,2,3], nous ne rappelons seulement ici que les points principaux. Il est composé :

- d'un transducteur général d'états finis qui permet grâce à une grammaire, un dictionnaire, et un ensemble de modèles de réaliser l'analyse morphologique d'une langue naturelle.
- d'un éditeur morphologique permettant une utilisation conversationnelle et interactive.

Ce système a nécessité la définition de langages mis à la disposition du linguiste pour l'écriture de règles de grammaire, des modèles et du dictionnaire. Leur emploi est extrêmement simple et ils ont donné lieu à l'écriture de compilateurs dont certains sont incrémentiels.

### SCHEMA FONCTIONNEL DE L'ANALYSEUR



Ce système est utilisé par l'équipe de traitement automatique des langues de l'Université de Grenoble II qui a ainsi réalisé la partie linguiste (mise au point des règles de grammaire, des modèles et du dictionnaire).

### 1.2 Le système d'analyse syntaxique :

Nous avons abandonné la méthode classique basée sur l'utilisation d'un analyseur syntaxique de type hors contexte plus ou moins modifié pour permettre certaines optimisations. Le traitement de la syntaxe d'une langue naturelle au moyen d'un tel modèle se heurte à la difficulté de construire un modèle valide. Le procédé algorithmique consiste à vérifier que la phrase proposée est acceptée par la grammaire. Il est donc nécessaire de disposer d'un modèle de la langue extrêmement développé, et sa mise au point nécessite de nombreuses retouches. Par ailleurs, toute phrase non conforme au modèle prévu est refusée, même si elle est compréhensible pour le lecteur. De plus, les structures hors contexte étant peu interprétables, il est nécessaire d'utiliser un module supplémentaire de transformation de structures.

Nous nous sommes orientés sur une autre voie qui consiste à essayer de construire directement une structure interprétable en n'utilisant uniquement que le graphe de toutes les relations possibles entre les catégories sans se préoccuper de la vérification grammaticale.

Nous avons donc défini un analyseur de dépendances dont la description algorithmique est donnée en [4]. Le procédé est simple : A partir du graphe complet de toutes les relations de dépendances possibles entre les catégories, on élimine celles qui sont incohérentes pour des raisons topologiques.

Nous avons donc mis à la disposition du linguiste un langage extrêmement simple pour lui permettre de définir les relations de dépendances et de les mettre à jour, ainsi que d'un éditeur de syntaxe pour l'utilisation conversationnelle du système.

L'avantage de cette méthode réside dans les faits suivants :

1°) les structures de dépendances étant plus naturelles que les structures hors contexte, la maîtrise de l'outil informatique par un

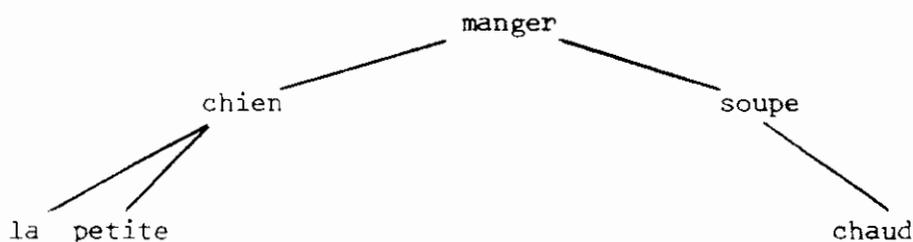
linguiste en est facilitée grâce à la transparence du système.

2°) La rapidité de l'algorithme rend l'utilisation interactive et conversationnelle possibles.

3°) L'obtention d'une structure interprétable même si la phrase n'est pas correcte syntaxiquement.

Exemple :

avec la phrase : "la petite chien manger soupe chaud", on obtient la structure suivante :



Nous avons été amenés à définir un deuxième module d'analyse dont l'objet était la vérification de certains traits syntaxiques tels que : les accords grammaticaux, la présence d'un article pour un substantif, etc ...

Le linguiste écrit ces règles d'accords au moyen du langage qui existait au C.E.T.A. et qui servait à définir les règles binaires hors contexte nécessaires à l'algorithme de COCKE. Ici, nous nous servons uniquement de la partie réécriture. L'algorithme consiste à réeffectuer un parcours post-fixé de l'arbre de dépendances (ou des sous-arbres) et à appliquer les règles hors contexte sur les catégories rencontrées.

Le système d'analyse syntaxique est donc composé de deux modules :

- un système de dépendances qui est utilisé comme filtre relationnel.
- un système hors contexte qui est utilisé comme filtre grammatical.

Ces deux modules peuvent fonctionner : soit en parallèle et leur action est complémentaire pour déterminer le plus rapidement possible la ou les solutions (structure(s) de dépendances syntaxiquement correcte(s)), soit en série le système hors contexte s'appliquant alors à la (ou aux) structure(s) de dépendances déterminée(s) par le filtre relationnel.

L'utilisation normale est l'utilisation des deux modules en parallèle .  
Elle correspond à l'obtention de solution(s) le plus rapidement possible.  
Son fonctionnement est du type "coroutine" et presque "hétérarchique".

Nous donnons quelques exemples avant de passer à une application possible  
du système P.I.A.F. qui va montrer toute sa puissance et sa généralité et  
qui va illustrer l'utilisation des deux modules syntaxiques en série.

### 1.3 Exemples :

```
load synt2250
R; T=1.51/2.25 09:14:19
```

```
start
EXECUTION BEGINS...
DONNEES ? ( T OU NOM TYPE)
> t
RESULTATS ? ( T OU I OU LISP OU 2250)
> t
MODE ? ( PAS A PAS OU CONTINU )
> pas a pas
ALGORITHMES ? ( DEP OR DEPCF OU DEP+CF )
> dep
>
  le beau petit chien noir mange un os .
```

```

|LE -(artd)
|BEAU -(adjq)
|PETIT -(adjq)
|CHIEN -(subc)
|      |NOIR -(adjq)
|MANGE -(verb)
|      |UN -(apep)
|      |OS -(subc)
```

\*

```
> l'oiseau qui se trouve sur le toit rouge n'a pas de nids .
```

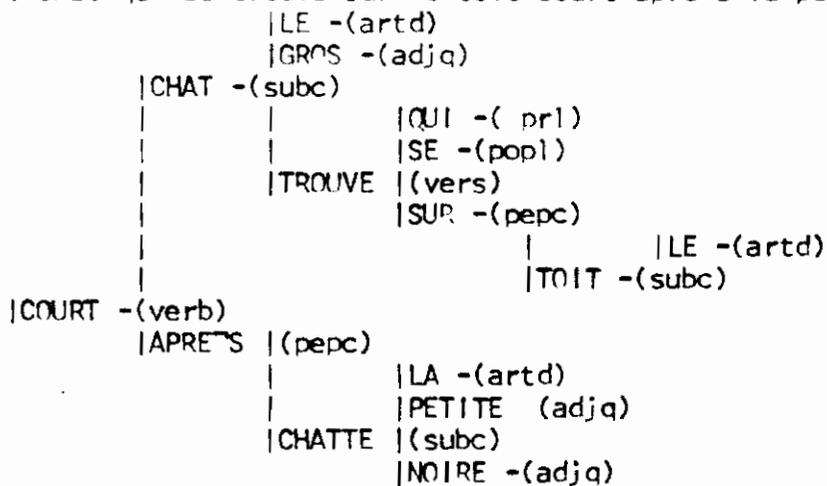
```

|L'-(artd)
|OISEAU |(subc)
|      |      |QUI -( pr1)
|      |      |SE -(pop1)
|      |TROUVE |(vers)
|      |      |SUR -(pepc)
|      |      |      |LE -(artd)
|      |      |      |TOIT -(subc)
|      |      |      |ROUGE -(adjq)
|      |N'-( ne)
|A -(verb)
|      |PAS -( pas)
|      |DE -(pepc)
|      |      |NIDS -(subc)
```

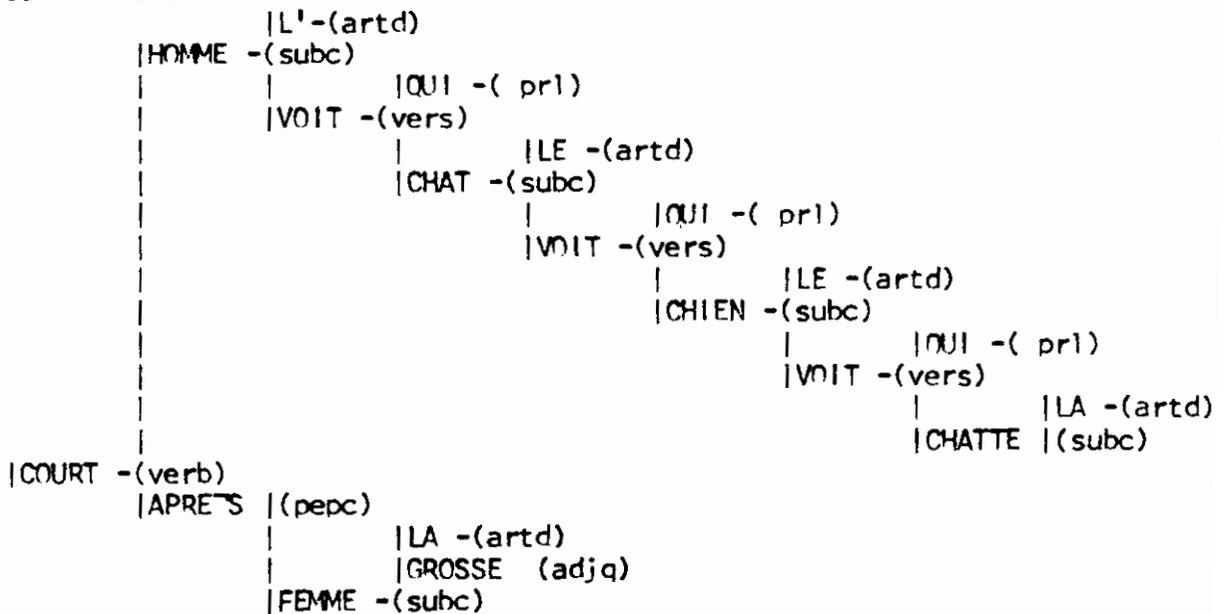
\*

```
>
```

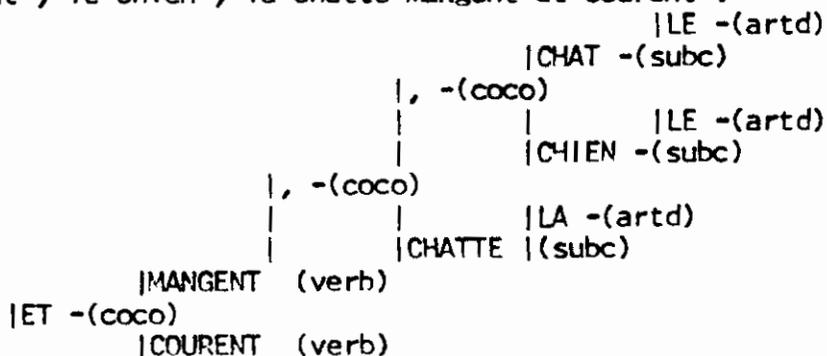
> le gros chat qui se trouve sur le toit court après la petite chatte noire .



\*  
> l'homme qui voit le chat qui voit le chien qui voit la chatte court après  
> la grosse femme .



\*  
> le chat , le chien , la chatte mangent et courent .



\*  
>