

**groupe  
d'intelligence  
artificielle**

**laboratoire scientifique  
de l'informatique  
70 route léon-lachamp  
13009 marseille**



**université d'aix-marseille II**

RAPPORT DE RECHERCHE INTERMEDIAIRE

sur le CONTRAT SESORI N° 73 047

période du 16 Octobre 1973

au 16 Octobre 1974

UN SYSTEME DE COMMUNICATION

HOMME-MACHINE EN FRANCAIS

PASERO R.

Octobre 1974

Responsable scientifique :

A. COLMERAUER

Les travaux ont porté sur :

- (1) l'amélioration de notre outil informatique de base : PROLOG. PROLOG est maintenant vraiment devenu un langage de programmation conversationnel avec toutes les facilités.

G. BATTANI et H. MELONI en ont écrit un interpréteur : FORTRAN qui, couplé avec le superviseur général écrit en PROLOG, par P. ROUSSEL, en a fait un langage de programmation relativement efficace et très transportable d'une machine à l'autre.

PROLOG tourne d'ailleurs maintenant sur des machines aussi différentes que ISM, CII, PDP, UNIVAC, DC et HONEYWELL-BULL et ceci à GRENOBLE, PARIS, LANNION, MARSEILLE, MONTREAL, EDIMBOURG, LONDRES, MILAN et VARSOVIE.

- (2) des études purement linguistiques, principalement sur un meilleur choix de structure profonde et sur un traitement très élaboré du problème de la résolution des pronoms avec leurs antécédents. Ce traitement fait intervenir des considérations sémantiques en plus des considérations syntaxiques.
- (3) la construction effective d'un système mettant en oeuvre les points décrits en (1) et (2) : la partie la plus intéressante de ce système fait l'objet d'une description détaillée dans ce qui suit.

.../...

## UNE PARTIE INTERESSANTE DE NOTRE SYSTEME

par R. PASERO

---

### INTRODUCTION

Notre travail s'inscrit dans le cadre de la réalisation d'un système permettant l'utilisation d'une langue naturelle pour communiquer avec l'ordinateur. La base du système est la démonstration automatique. Nous utilisons un langage de programmation appelé PROLOG où chaque instruction n'est rien d'autre qu'un énoncé de logique du 1er ordre et où exécuter un programme revient à démontrer un théorème.

Je me suis occupé personnellement de l'étude sémantique du français. De ce fait, mes recherches ont un aspect linguistique, mais dans un but qui est de mettre en évidence des mécanismes généraux permettant à l'ordinateur de traduire un texte français dans le langage de la logique du premier ordre.

Mes principaux objectifs à cet effet, cette année, ont été :

- . de définir une structure profonde du français adaptée à ce type de problème.
- . l'étude des quantificateurs dans la phrase, la traduction d'une phrase dans le langage de la logique du 1er ordre étant principalement basée sur le type des quantificateurs.
- . l'étude d'un mécanisme de résolutions des pronoms basé sur des critères sémantiques, en plus des critères syntaxiques.

1 - Définition de la structure profonde du français

La structure profonde que nous avons adoptée offre une représentation des phrases assez proche de la structure de surface, tout en faisant apparaître les relations sémantiques entre les éléments des phrases. De plus, les mécanismes de traduction de cette structure en un ensemble de formules logiques du 1er ordre sont alors assez simples.

Dans cette structure, une phrase aura la forme suivante :

< phrase > ::= < quantificateur > . (< trait prep. >). < phrase >. < phrase > |  
 < connecteur >. < phrase >. < phrase > | NON. < phrase > |  
 < prédicat >

< quantificateur > ::= CHAQUE | UN | LE | AUCUN | PRON | PROP

< trait prep > ::= < prep >. < genre >. < nombre >. < indice >

< prep > ::= SUJ | C(DIR) | C(A) | C(DE) | ...

< genre > ::= F | M | \* G

< nombre > ::= S | P | \* N

< indice > ::= \* I | < nom propre >

< connecteur > ::= ET | OU

< prédicat > ::= PR (< arguments >. < verbe >) | PR (< argument >. REPONSE) | PR (VRAI

< arguments > ::= < argument > | < argument >. < arguments >

< argument > ::= (< nombre >. < indice >)

< verbe > ::= CRIER | AIMER | CHIEN | HOMME | ...

< nom propre > ::= HORTENSE | EMILE | ...

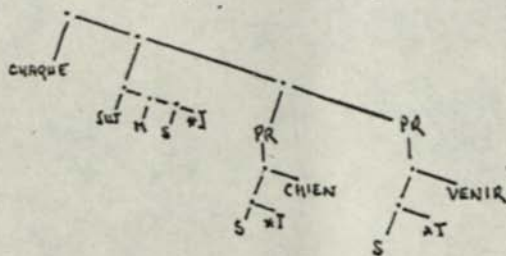
Exemple : La phrase

1) chaque chien vient

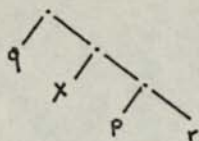
sera représentée par :

CHAQUE . (SUJ, M, S, \*I). PR((S, \*I). CHIEN). PR((S, \*I). VENIR)

qui si l'on définit " . " comme opérateur droite-gauche se représente par le peigne :



A chaque syntagme nominal quantifié par q est donc associée une structure de la forme :



que nous écrirons : q.x.p.r

- où
- q est un quantificateur
  - x un indice avec les traits du syntagme nominal
  - p la représentation de la proposition relative
  - r la représentation du reste de la proposition principale.

Ainsi l'exemple 1) peut se traduire schématiquement dans cette structure par :

CHAQUE. X. CHIEN (X) . VIENT (X)

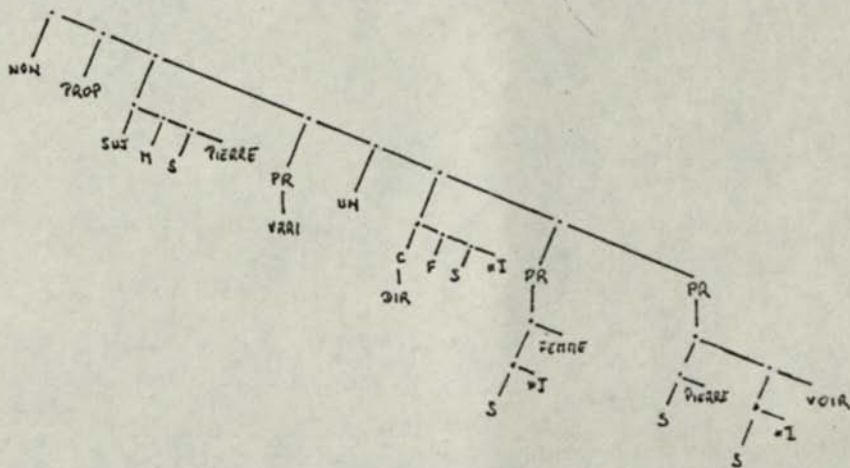
qui signifie "chaque x qui est chien vient".

La négation liée au verbe est ramenée tout en haut de l'arbre représentant la phrase :

exemple : la phrase

2) Pierre ne voit pas une femme

sera représentée par :



## 2 - Mécanismes généraux de traduction

- Principes élémentaires :

La traduction d'une phrase en un ensemble d'énoncés logiques du 1er ordre repose principalement sur le type des quantificateurs. Ainsi nous associerons à une phrase dont la structure est :

CHAQUE . x . p . r

une formule logique quantifiée universellement de la forme :

$$\forall x p_1 \supset r_1$$

où  $p_1$  et  $r_1$  sont les formules associées à p et r,

et nous associerons à une phrase de la forme :

UN . x . p . r

la formule logique

$$\exists x p_1 \wedge r_1$$

Ainsi la phrase de l'exemple 1) se traduira par :

$$\forall x \text{ chien } (x) \supset \text{ vient } (x)$$

soit sous forme clausale par :

$$+ \text{ vient } (x) - \text{ chien } (x).$$

Une phrase quantifiée par LE donnera lieu à une forme plus complexe à cause de la présupposition d'unicité :

exemple : la phrase :

3) Le chien court

qui est représentée par :

LE, X . chien (x) . Court (x)

se traduira par :

$$\exists x [\text{chien } (x) \wedge \text{court}(x) \wedge \forall y (\text{chien } (y) \supset x = y)]$$

soit sous forme clausale :

$$+ \text{ chien } (A)$$

$$+ \text{ vient } (A)$$

$$+ \text{ EG}(Y, A) - \text{ chien } (Y).$$

- Axiomatisation :

L'algorithme utilisé pour engendrer les énoncés logiques repose en fait principalement sur l'algorithme de Skolemisation que l'on peut axiomatiser assez simplement si l'on considère que les seuls quantificateurs sont CHAQUE et UN pour simplifier.

On va utiliser pour cela un prédicat spécial PAS, qui signifie "on n'a pas", à deux arguments :

$$\text{PAS} ( X, U )$$

où - X est en fait l'arbre, considéré comme un terme, représentant la phrase.  
 - U un terme construit à partir de deux fonctions d et g et des variables universelles permettant de créer les fonctions de Skolem.

Les axiomes sont alors :

- 1 + PAS (chaque. x. p. q, u) - PAS (ou, non, p, q, x, u)
- 2 + PAS (un, u, p, q, u) - PAS(et, p, q, u)
- 3 + PAS (et, p, q, u) - PAS(p, g(u))
- 4 + PAS (et, p, q, u) - PAS(q, d(u))
- 5 + PAS (ou, p, q, u) - PAS(p, g(u)) - PAS (q, d(u))
- 6 + PAS (non, non, p, u) - PAS (p, u)
- 7 + PAS (Pr(x), u) + Pr(x)
- 8 + PAS (non, Pr(x), u) - Pr(x)
- 9 + PAS (non, chaque, x, p, q, u) - PAS (un, x, p, non, q, u)
- 10 + PAS (non, un, x, p, q, u) - PAS(chaque, x, p, non, q, u)
- 11 + PAS (non, et, p, q, u) - PAS(ou, (non, p), non, q, u)
- 12 + PAS (non, ou, p, q, u) - PAS(et, (non, p), non, q, u)

( les caractères soulignés sont des variables quantifiées universellement).

Etant donné une phrase représentée par un arbre A, il suffira, pour obtenir la forme clausale constituant la traduction de la phrase, de canceler sur ces axiomes le littéral :

- PAS (A, i)                    i étant un entier permettant de numéroter les phrases

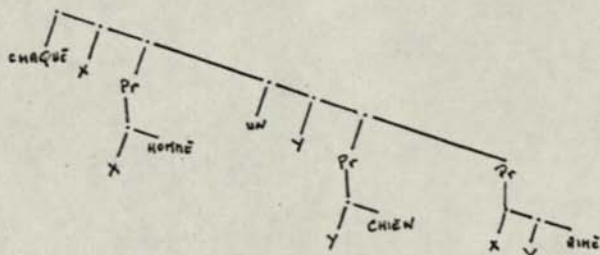
jusqu'à ce qu'on obtienne une résolvente ne contenant plus de littéral -PAS (X, u)



exemple : considérons la phrase :

4) chaque homme aime un chien

représentée schématiquement par l'arbre :



considérons la clause :

13  $\neg$ PAS (chaque, x, Pr(x, homme), un, y, Pr(y, chien), Pr(x, y, aime), o)  
on déduit de 1) la résolvente

14  $\neg$ PAS (ou, (non, Pr(x, homme)), un, y, Pr(y, chien), Pr(x, y, aime), x, o)  
de 5)

15  $\neg$ PAS (non, Pr(x, homme), g(x, o))  $\neg$ PAS (un, y, Pr(y, chien), Pr(x, y, aime), d(x, o))  
de 8)

16  $\neg$ Pr(x, homme)  $\neg$ PAS(un, y, Pr(y, chien), Pr(x, y, aime), d(x, o))  
de 2)

17  $\neg$ Pr(x, homme)  $\neg$ PAS(et, Pr(d(x, o), chien), Pr(x, d(x, o), aime), d(x, o))  
de 3)

18  $\neg$ Pr(x, homme)  $\neg$ PAS(Pr(d(x, o), chien), g(d(x, o)))  
de 7)

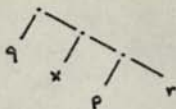
19  $\neg$ Pr(x, homme) + Pr(d(x, o), chien)  
de 17 et 4 on déduit également

20  $\neg$ Pr(x, homme)  $\neg$ PAS(Pr(x, d(x, o), aime), d(d(x, o)))  
de 7)

21  $\neg$ Pr(x, homme) + Pr(x, d(x, o), aime)

Les deux clauses 19 et 21 représentent alors les énoncés logiques correspondant à la phrase 4).

La structure à deux termes p et r que nous avons adoptée,



est nécessaire pour bien traiter les syntagmes nominaux de type définis. Au niveau sémantique, pour définir des interprétations, il semble qu'il faille se placer dans une logique à 3 valeurs, où une phrase pourra avoir comme valeur de vérité :

- vrai
- faux
- indéterminé

### 3 - Résolution des pronoms

Notre objectif consiste à faire appel, pour résoudre les pronoms, à des critères sémantiques en plus des critères syntaxiques qui sont parfois insuffisants.

Pour cela les pronoms ne devront pas être résolus au niveau de l'analyse syntaxique, mais par une déduction logique.

Le principe pour réaliser ceci va être d'associer, au niveau de la traduction de la phrase, à chaque syntagme nominal un "numéro" différent et de produire pour chacun d'eux une clause de la forme :

+ sn(i, x) signifiant "x a les propriétés énoncées par le syntagme n° i"

Exemple : la phrase

5) un homme vient

se traduira par :

- (a) + sn(1, A)
- (b) + vient (A)
- (c) + homme (A)

si 1 est le numéro associé au syntagme "chaque homme".

La clause (a) signifie alors :

"A a les propriétés énoncées par le syntagme n° 1".

Considérons maintenant une phrase contenant un pronom :

Exemple : la phrase

6) il chante.

Si le "numéro" associé au pronom "il" est 2, nous traduirons cette phrase par la clause :

(d) + chante (x) - ant (2, i) - sn(i, x) où x et i sont des variables

qui signifie :

" Si l'antécédent de 2 est i et  
si x a les propriétés énoncées par le syntagme n° 1 alors  
x chante".

Si on suppose maintenant que l'on a affirmé :

(e) + ant (2, 1)

en unifiant (e) avec (d) on obtient la résolvente :

(f) + chante (x) - sn(1, x)

et en unifiant (f) avec (a) on obtient la résolvente :

(g) + chante (A).

Ce mécanisme peut s'appliquer également dans le cas de pronom faisant référence à un syntagme nominal quantifié par CHAQUE.

Considérons par exemple les deux phrases :

6) - Chaque psychiatre est une personne.

7) - Chaque personne qu'il analyse est malade.

la phrase 6) se traduira par les clauses :

a) + sn(1, x) - psychiatre (x) où x est une variable

b) + personne (x) - psychiatre (x)

si 1 est le numéro associé au syntagme nominal "chaque psychiatre".

La clause a) signifie alors :

" x a les propriétés énoncées par le syntagme n° 1 si x est  
psychiatre".

La phrase 7) produira une clause de la forme :

c) + malade (y) - personne (y) - ant(2, i) - sn(i, z) - analyse(z, y)  
où y, i, z sont des variables,

si 2 est le numéro associé au syntagme nominal correspondant au pronom "il".

Si on suppose également que l'on a affirmé :

d) + ant (2, 1)

en unifiant d) avec c) on obtient la résolvante :

e) + malade (y) - personne(y) - sn(1, 2) - analyse (z, y)

et en unifiant e) avec a) :

f) + malade (y) - personne (y) - psychiatre (z) - analyse (z, y).

A l'heure actuelle, la recherche de l'antécédent consiste à essayer tous les syntagmes nominaux qui précèdent le pronom jusqu'à une certaine "distance". Cette recherche utilise des critères syntaxiques et sémantiques pour éliminer certaines solutions. Une possibilité ayant été obtenue, elle ne sera définitivement conservée que si l'énoncé logique obtenu ne rend pas le système inconsistant. Dans ce cas on essaye une nouvelle solution.

## 4 - Exemples de conversation :

Voici un exemple de conversation que nous avons eue. Le texte tapé par l'interlocuteur au télétype est précédé d'une \*.

\* -ENQUETE!

\* (CHACUE PERSONNE) QUI AIME UN CHIEN CHIE.

CHACUE PERSONNE QUI AIME UN CHIEN CHIE POINT NIL

CHACUE (SUJ. F. S. \*XO) (ET. FR(CS. \*XO). PERSONNE) UN. (C(DI) M. S. \*X1) (ET. FR(CS. \*X1) (CHIEN). FR(VHAI)). FR(CS. \*XO) (S. \*A1) AIMER). FR(CS. \*XO) CHIE)  
OK. CONTINUEZ..

\* EMILE AIME CHACUE CHIEN.

EMILE AIME CHACUE CHIEN POINT NIL

PROP. (SUJ. M. S. EMILE) FR(CS. EMILE) PERSONNE CHACUE (C(DI) M. S. \*XO) (ET. FR(CS. \*XO) (CHIEN). FR(VHAI)). FR(CS. EMILE) (S. \*XO) AIMER)  
OK. CONTINUEZ..

\* EMILE NE CHIE PAS.

EMILE CHIE PAS POINT NIL

NON PROP. (SUJ. M. S. EMILE) FR(CS. EMILE) PERSONNE FR(CS. EMILE) CHIE)  
HYPOTHESE INCONSISTANTE. ESSAYONS UNE AUTRE ANALYSE...  
LA REPONSE:  
EMILE CHIE PAS POINT NIL  
EST REPUEE CAR INCONSISTANTE.

\* QUI EST-CE QUI CHIE?

QUI EST-- CE QUI CHIE ? NIL

CHACUE (SUJ. \*XO. S. \*X1) (ET. FR(CS. \*X1) PERSONNE) FR(CS. \*X1) CHIE) FR(CS. \*X1) REPONSE  
QUESTION

Reponse: PROP. (SUJ. \*XO. S. EMILE) FR(CS. EMILE) PERSONNE) FR(CS. EMILE) CHIE)  
OK. CONTINUEZ.. (i.e: EMILE)

\* QUI EST-CE QUI AIME EMILE?

QUI EST-- CE QUI AIME EMILE ? NIL

CHACUE (SUJ. \*XO. S. \*X1) (ET. FR(CS. \*X1) PERSONNE) PROP. (C(DI) M. S. EMILE) FR(CS. EMILE) PERSONNE) FR(CS. \*X1) (S. EMILE) AIMER). FR(CS. \*X1) REPONSE  
QUESTION

Reponse: JE NE SAIS PAS INTERO. (SUJ. \*XO. S. \*X1) FR(CS. \*X1) PERSONNE) PROP. (C(DI) M. S. EMILE) FR(CS. EMILE) PERSONNE) FR(CS. \*X1) (S. EMILE) AIMER)  
OK. CONTINUEZ.. (i.e: JE NE SAIS PAS)

\* CHACUE PERSONNE CHIE.

CHACUE PERSONNE CHIE POINT NIL

CHACUE (SUJ. F. S. \*XO) (ET. FR(CS. \*XO) PERSONNE) FR(VHAI)). FR(CS. \*XO) CHIE)  
OK. CONTINUEZ..

\* QUI EST-CE QUI CRIE?

GUI. EST. -- CE. QUI. CRIE. ? .NIL

CHACUE. (SUJ. \*XO. S. \*X1) . (ET. PR((S. \*X1) . PERSONNE) . PR((S. \*X1) . CRIER)) . PR  
((S. \*X1) . REFONSE)  
QUESTION

Reponse: CHACUE. (SUJ. \*XO. S. \*X1) . PR((S. \*X1) . PERSONNE) . PR((S. \*X1) . CRIER)  
OK. CONTINUEZ.. (ie: chaque personne)

\* AUCUNE PERSONNE NE CRIE.

AUCUNE. PERSONNE. CRIE. POINT. NIL

AUCUN. (SUJ. F. S. \*XO) . (ET. PR((S. \*XO) . PERSONNE) . PR(VRAI)) . PR((S. \*XO) . CRI  
ER)

PHRASE INCONSISTANTE. ESSAYONS UNE AUTRE ANALYSE...

LA PHRASE:

AUCUNE. PERSONNE. CRIE. POINT. NIL

EST REFUSEE CAR INCONSISTANTE.

\* AUCUNE PERSONNE N'AI ME HORTENSE.

AUCUNE. PERSONNE. AIME. HORTENSE. POINT. NIL

AUCUN. (SUJ. F. S. \*XO) . (ET. PR((S. \*XO) . PERSONNE) . PR(VRAI)) . PROP. (C(LI)R. F  
. S. HORTENSE) . PR((S. HORTENSE) . PERSONNE) . PR((S. \*XO) . (S. HORTENSE) . AIMER)

OK. CONTINUEZ..

\* QUI EST-CE QUI AIME HORTENSE?

GUI. EST. -- CE. QUI. AIME. HORTENSE. ? .NIL

CHACUE. (SUJ. \*XO. S. \*X1) . (ET. PR((S. \*X1) . PERSONNE) . PROP. (C(DI)R. F. S. HORT  
ENSE) . PR((S. HORTENSE) . PERSONNE) . PR((S. \*X1) . (S. HORTENSE) . AIMER)) . PR((S  
\*X1) . REFONSE)

QUESTION

Reponse: AUCUN. (SUJ. \*XO. S. \*X1) . PR((S. \*X1) . PERSONNE) . PROP. (C(DI)R. F. S. HORTENSE)  
. PR((S. HORTENSE) . PERSONNE) . PR((S. \*X1) . (S. HORTENSE) . AIMER)  
OK. CONTINUEZ.. (ie: Aucune personne)

\* CHACUE CHIEN CRIE.

CHACUE. CHIEN. CRIE. POINT. NIL

CHACUE. (SUJ. M. S. \*XO) . (ET. PR((S. \*XO) . CHIEN) . PR(VRAI)) . PR((S. \*XO) . CRIER  
)

OK. CONTINUEZ..

\* CHACUE PERSONNE QUI AIME UN CHIEN LE DONNE A HORTENSE.

CHACUE. PERSONNE. QUI. AIME. UN. CHIEN. LE. DONNE. A. HORTENSE. POINT. NIL

CHACUE. (SUJ. F. S. \*XO) . (ET. PR((S. \*XO) . PERSONNE) . UN. (C(LI)R. M. S. \*X1) . (ET  
. PR((S. \*X1) . CHIEN) . PR(VRAI)) . PR((S. \*XO) . (S. \*X1) . AIMER)) . PRON. (C(LI)R. M.  
S. \*X2) . PR((S. \*X2) . ETRE) . PROP. (C(A)R. F. S. HORTENSE) . PR((S. HORTENSE) . PE  
RSONNE) . PR((S. \*XO) . (S. \*X2) . (S. HORTENSE) . DONNER)

OK. CONTINUEZ..

\* QUI EST-CE QUI DONNE UN CHIEN A HORTENSE?

QUI. EST. CE. QUI. DONNE. UN. CHIEN. A. HORTENSE. ? . NIL

CHIEN. (SUJ. \*X0. S. \*X1). (ET. PR(CS. \*X1). PERSONNE). UN. (CC(DI)D. M. S. \*X2). (ET. PR(CS. \*X3). CHIEN). PR(VRAI)). PROP. (CC(A). F. S. HORTENSE). PR(CS. HORTENSE). PERSONNE). PR(CS. \*X1). (S. \*X3). (S. HORTENSE). DONNER). PR(CS. \*X1). REFUSÉ

QUESTION

Réponse: PROP. (SUJ. \*X0. S. EMILE). PR(CS. EMILE). PERSONNE). UN. (CC(DI)D. M. S. \*X1). (ET. PR(CS. \*X1). CHIEN). PR(VRAI)). PROP. (CC(A). F. S. HORTENSE). PR(CS. HORTENSE). PERSONNE). PR(CS. EMILE). (S. \*X1). (S. HORTENSE). DONNER)

OK. CONTINUEZ..

(i.e: EMILE)

\* TOUTE PERSONNE QUI DONNE UN CHAT A HORTENSE L'AIME.

TOUTE. PERSONNE. QUI. DONNE. UN. CHAT. A. HORTENSE. L. AIME. POINT. NIL

CHACUN. (SUJ. F. S. \*X0). (ET. PR(CS. \*X0). PERSONNE). UN. (CC(DI)D. M. S. \*X1). (ET. PR(CS. \*X1). CHAT). PR(VRAI)). PROP. (CC(A). F. S. HORTENSE). PR(CS. HORTENSE). PERSONNE). PR(CS. \*X0). (S. \*X1). (S. HORTENSE). DONNER). PRON. (CC(DI)D. \*X2. S. \*X3). PR(CS. \*X3). ETRE). PR(CS. \*X0). (S. \*X3). AIMER)

PHRASE INCONSISTANTE. ESSAYONS UNE AUTRE ANALYSE...

LA PHRASE:

TOUTE. PERSONNE. QUI. DONNE. UN. CHAT. A. HORTENSE. L. AIME. POINT. NIL  
ET REFUSE CAR INCONSISTANTE.

\* TOUTE PERSONNE QUI AIME HORTENSE LUI DONNE UN CHAT.

TOUTE. PERSONNE. QUI. AIME. HORTENSE. LUI. DONNE. UN. CHAT. POINT. NIL

CHACUN. (SUJ. F. S. \*X0). (ET. PR(CS. \*X0). PERSONNE). PROP. (CC(DI)D. F. S. HORTENSE). PR(CS. HORTENSE). PERSONNE). PR(CS. \*X0). (S. HORTENSE). AIMER). PRON. (CC(DI)D. M. S. \*X1). PR(CS. \*X1). ETRE). UN. (CC(A). M. S. \*X2). (ET. PR(CS. \*X2). CHAT). PR(VRAI)). PR(CS. \*X0). (S. \*X1). (S. \*X2). DONNER)

PHRASE INCONSISTANTE. ESSAYONS UNE AUTRE ANALYSE...

CHACUN. (SUJ. F. S. \*X0). (ET. PR(CS. \*X0). PERSONNE). PROP. (CC(DI)D. F. S. HORTENSE). PR(CS. HORTENSE). PERSONNE). PR(CS. \*X0). (S. HORTENSE). AIMER). PRON. (CC(DI)D. \*X1. S. \*X2). PR(CS. \*X2). ETRE). UN. (CC(DI)D. M. S. \*X3). (ET. PR(CS. \*X3). CHAT). PR(VRAI)). PR(CS. \*X0). (S. \*X3). (S. \*X2). DONNER)

PHRASE INCONSISTANTE. ESSAYONS UNE AUTRE ANALYSE...

LA PHRASE:

TOUTE. PERSONNE. QUI. AIME. HORTENSE. LUI. DONNE. UN. CHAT. POINT. NIL  
ET REFUSE CAR INCONSISTANTE.

\* TOUTE PERSONNE QUI AIME UN CHAT LE DONNE A HORTENSE.

TOUTE. PERSONNE. QUI. AIME. UN. CHAT. LE. DONNE. A. HORTENSE. POINT. NIL

CHACUN. (SUJ. F. S. \*X0). (ET. PR(CS. \*X0). PERSONNE). UN. (CC(DI)D. M. S. \*X1). (ET. PR(CS. \*X1). CHAT). PR(VRAI)). PR(CS. \*X0). (S. \*X1). AIMER). PRON. (CC(DI)D. M. S. \*X2). PR(CS. \*X2). ETRE). PROP. (CC(A). F. S. HORTENSE). PR(CS. HORTENSE). PERSONNE). PR(CS. \*X0). (S. \*X2). (S. HORTENSE). DONNER)

OK. CONTINUEZ..

## CONCLUSION

Le système que nous expérimentons pour l'instant n'est encore que rudimentaire, mais les résultats obtenus encouragent, me semble-t-il, à poursuivre dans cette voie.

Le fait qu'il soit conversationnel, du moins en partie, permet de supprimer certaines ambiguïtés au niveau de l'analyse (si plusieurs analyses sont possibles, on ne conservera que celle qui ne rend pas le système inconsistant) et également au niveau de la résolution des pronoms où un test d'inconsistance élimine certaines résolutions de pronoms qui pourraient paraître correctes d'un point de vue purement syntaxique.