

**groupe
d'intelligence
artificielle**

**laboratoire scientifique
de linguistique
70 route léon-lachamp
13009 marseille**

UN ESSAI DE COMMUNICATION SENSÉE
EN LANGUE NATURELLE

PASERO R.

SEPTEMBRE 1976

université d'aix-marseille II

UN ESSAI DE COMMUNICATION SENSEE
EN LANGUE NATURELLE

PASERO R.

SEPTEMBRE 1976

Recherche subventionnée par le
contrat IRIA-SESORI n° 73 047.

AVANT PROPOS

CHAPITRE I : PRESENTATION GENERALE DU SYSTEME

CHAPITRE II : STRUCTURE PROFONDE DU FRANCAIS

CHAPITRE III : TRADUCTION D'UNE PHRASE

- 1 Approche empirique
- 2 Énoncés élémentaires traduisant une phrase déclarative
- 3 Énoncés élémentaires associés aux phrases interrogatives
- 4 Structure des énoncés élémentaires

CHAPITRE IV : PRODUCTION DES ÉNONCÉS ÉLÉMENTAIRES

- 1 Production des énoncés exprimant la relation décrite par le verbe
- 2 Production des énoncés élémentaires "définissant les domaines"
- 3 Présupposition

CHAPITRE V : TRAITEMENT DES PRÉSUPPOSITIONS

- 1 Résolution des pronoms
- 2 Traitement des présuppositions associées à l'emploi du défini

CHAPITRE VI : DÉDUCTIONS

- 1 Contrôle des déductions
- 2 Règles de raisonnement
- 3 Contrôle des échanges

EXEMPLES DE CONVERSATIONS

BIBLIOGRAPHIE

AVANT-PROPOS

Le travail que nous nous proposons de décrire ici s'inscrit dans le cadre d'une recherche de communication homme-machine en langage naturel et représente la base déterminante à partir de laquelle un système intelligent pourrait être développé.

Il ne s'agit pas pour nous d'aborder tous les problèmes concernant l'élaboration de tels systèmes mais de décrire essentiellement, à travers un "système qui tourne", une structure et un algorithme de représentation du Français en logique permettant la résolution retardée des pronoms et la mise en évidence des liens qu'ils créent entre les phrases.

Nos principaux objectifs sont :

- de définir une structure profonde du Français adaptée à notre problème. La structure profonde choisie, tout en ne s'écartant pas trop de la structure de surface s'apparente à la structure logique des λ -expressions, ce qui facilite la traduction en énoncés élémentaires (chapitre II).
Nous ne nous occupons pas des problèmes d'analyse du Français qui intéressent plus particulièrement A. COLMERAUER. L'entrée du système est donc la structure profonde d'une phrase;
- de donner une représentation du Français dans un sous-ensemble de la logique, cf (1), permettant de traiter d'une façon analogue les phrases déclaratives et interrogatives, et de retarder le plus possible la résolution des pronoms de manière à faire intervenir dans la recherche de l'antécédent des critères sémantiques (chapitre III).
Nous avons voulu également, dans cette traduction, "limiter" la portée des quantificateurs universels de manière à ce que les existentiels ne dépendent que d'un minimum de variables universelles.
Dans :

Chaque homme voit le chien

"le chien" ne doit pas dépendre de "l'homme" choisi. Les pronoms peuvent modifier la portée des quantificateurs dans une même phrase,

Chaque homme épouse la femme qu'il aime,

et également d'une phrase à l'autre,

Chaque homme possède un nez.

Il possède aussi une bouche;

- de dégager des mécanismes de traduction basés essentiellement sur le rôle des quantificateurs et de la négation (chapitre IV).

Pour pouvoir traiter correctement les présuppositions associées à l'emploi du défini et de pronoms du type :

cet animal qui vient ...

nous avons été amenés à nous placer dans une logique à trois valeurs.

Nous donnons, dans le chapitre V, quelques précisions sur le traitement des présuppositions liées à l'emploi du défini (existence et unicité) et de pronoms en décrivant les critères intervenant dans leur résolution et, dans le chapitre VI, quelques informations sur les contrôles que nous exerçons au niveau des déductions pour limiter l'arbre de recherche, sur les règles de raisonnement dont dispose la machine, et sur la façon dont sont gérés les échanges.

Nous donnons, à la fin, quelques exemples simples de conversations que nous avons eues avec la machine.

I - PRESENTATION GENERALE DU SYSTEME

C'est un système qui permet de "dialoguer" avec la machine en lui décrivant un "monde" à l'aide de phrases affirmatives ou négatives, et de lui poser des questions sur ce monde.

L'utilisateur a les possibilités suivantes :

- communiquer de l'information, c'est à dire énoncer des phrases affirmatives ou négatives.
- supprimer de l'information en énonçant volontairement ou involontairement des phrases qui contredisent des faits déjà énoncés.
- poser des questions du type "pourquoi ..." et "qui ..." auxquelles le système doit répondre (-je ne sais pas - éventuellement).
- aider la machine dans sa compréhension du monde en répondant à ses questions (par -oui- ou -non-),
- demander à la machine de lui communiquer tous les résultats intermédiaires obtenus dans sa compréhension du monde.
- demander à la machine de lui décrire le "monde".
- arrêter le système en disant "au revoir".

La machine contrôle constamment si ce qui est énoncé par l'utilisateur a un "sens". Pour cela le système vérifie que les présuppositions attachées à un énoncé ne contredisent pas les faits antérieurs.

Exemple 1 : Qui voit le chien ?

Cette question n'a de sens et n'admet donc de réponse que si "il existe un et un seul chien" ne contredit pas les faits déjà énoncés.

C'est également au niveau du traitement des présuppositions que sont résolus les pronoms :

Exemple 2 : Il vient.

Cette phrase n'a de sens (vrai ou faux) que si le pronom il est résolu.

II - STRUCTURE PROFONDE DU FRANCAIS

La structure profonde des phrases énoncées par l'interlocuteur est la suivante :

```

<phrase> ::= <phrase interrogative> | <phrase déclarative>

<phrase interrogative> ::= pourquoi.<formule> |
                           intero.<genre-nb>.<propriété>.<propriété>

<phrase déclarative> ::= <formule> | oui | non | aurovoir

<formule> ::= <quantificateur>.<genre-nb>.<propriété>.<propriété> |
              non.<formule> | et.<formule>.<formule> |
              ou.<formule>.<formule> | <formule élémentaire>

<formule élémentaire> ::= pr(<groupe verbal>) | vrai

<propriété> ::= 1.<variable>.<formule>

<quantificateur> ::= chaque | un | le | pron | aucun

<genre-nb> ::= <genre>.<nb> | <variable>

<genre> ::= m | f | <variable>

<nb> ::= s | p | <variable>

<groupe verbal> ::= <variable>.<variable>.<variable>.<verbe 3>.nil |
                   <variable>.<variable>.<verbe 2>.nil |
                   <variable>.<nom propre>.nom.nil |
                   <variable>.<verbe 1>.nil |
                   <verbe 0>.nil

<variable> ::= "toute variable PROLOG"

<verbe 3> ::= "tout verbe à 3 arguments"

<verbe 2> ::= "tout verbe à 2 arguments"

<verbe 1> ::= "tout verbe à 1 argument" | <nom commun>

<verbe 0> ::= détail

<nom propre> ::= "tout nom propre"

<nom commun> ::= "tout nom commun"

```

Exemple 3 : chaque enfant crie

est représenté par :

chaque.(m.s).(1.x.pr(x.enfant.nil)).1.y.pr(y.crier.nil)

que l'on traduira de façon schématique dans la suite par :

chaque(x.enfant(x),x.cris(x))

Pour permettre le traitement du pluriel, les objets du monde sont considérés comme des ensembles d'individus et les relations comme des relations entre ensembles (à partir desquelles on peut déduire des relations au niveau des individus).

Ainsi l'individu "le chien" de l'exemple 1 est représenté par un ensemble à un seul élément.

Le fait qu'il n'y ait qu'un et un seul chien, dans cet exemple, s'exprime par une "règle d'inclusion" signifiant que tout autre "ensemble de chien" est inclus dans celui-ci.

Les noms propres sont traités comme des arguments définis, en ce sens qu'un nom propre, Pierre par exemple, est traduit par :

le x qui a pour nom Pierre ...

ce qui est représenté par :

le(x.Pr(x.Pierre.nom.nil),...)

La structure des réponses produites par la machine est décrite par les règles suivantes :

<réponse> ::= <réponse à qui> | <réponse à pourquoi> |
 <demande d'information> | <rejet d'information>

<réponse à qui> ::= <défini>.<nom commun> | <indéfini>.<nom commun> |
 <nom propre> | je ne sais pas !

<réponse à pourquoi> ::= parceque je sais raisonner et que vous m'avez
 dit : <liste de formules> | je ne sais pas !

<demande d'information> ::= l'antécédent du pronom : sn(pron.<genre-nb>.
 <propriété>) est bien le syntagme : sn(<syntagme>) |
 oui | ou non !

<syntagme> ::= le.<genre-nb>.<propriété> |
 un.<genre-nb>.<propriété> |
 chaque.<genre-nb>.<propriété>

<rejet d'information> ::= vous supposez l'existence de : sn(le.<genre-nb>.
 <propriété>). cela est absurde. j'oublie cette phrase. |
 c'est trop complexe ! j'oublie cette phrase. |
 vous vous contredisez. j'oublie que vous m'avez dit :
 ph(<formule>) |
 vous vous contredisez. j'oublie que vous m'avez dit :
 sn(pron.<genre-nb>.<propriété>) en parlant de :
 sn(<syntagme>) |
 vous vous contredisez. j'oublie que : sn(le.<genre-nb>.
 <propriété>) existe. |
 je ne comprends pas à qui se réfère le pronom :
 sn(pron.<genre-nb>.<propriété>). j'oublie cette phrase.

<défini> ::= le | la | les

<indéfini> ::= un | une | des

<liste de formules> ::= <vide> | ph(<formule>).<liste de formules>

<vide> ::=

III - TRADUCTION D'UNE PHRASE

1°/ APPROCHE EMPIRIQUE

La structure profonde est traduite en un ensemble d'énoncés élémentaires qui sont eux-mêmes transformés en clauses régulières, cf (1).

Les énoncés élémentaires associés à une phrase consistent en :

- a) l'énoncé de la relation V décrite par le verbe à l'aide d'une formule de la forme suivante :

$$V(\underline{x}, \underline{y}) + (\underline{x}, \underline{y}) \in E$$

avec ici V un prédicat binaire, et E son domaine de définition;

- b) la définition de E, c'est-à-dire de la propriété :

$$(\underline{x}, \underline{y}) \in E$$

La relation $(\underline{x}, \underline{y}) \in E$ se définit à l'aide de relations élémentaires

$$\underline{x} \in E'$$

de la façon suivante :

$$(\underline{x}, \underline{y}) \in E + \underline{x} \in E_1 \wedge \underline{y} \in E_2(\underline{x})$$

le problème étant alors de définir les deux propriétés $\underline{x} \in E_1$ et $\underline{y} \in E_2(\underline{x})$

Exemple 4 : a) chaque garçon possède une voiture

b) chaque $(x, \text{garçon}(x), x, \text{un}(y, \text{voiture}(y), y, \text{possède}(x, y)))$

La phrase a) représentée par la structure profonde b) se traduit par :

i possède $(\underline{x}, \underline{y}) + (\underline{x}, \underline{y}) \in E$

ii $(\underline{x}, \underline{y}) \in E + \underline{x} \in E_1 \wedge \underline{y} \in E_2(\underline{x})$

iii $\underline{x} \in E_1 + \text{garçon}(\underline{x})$

iv $f(\underline{x}) \in E_2(\underline{x}) + \underline{x} \in E_1$ (où $f(\underline{x})$ représente "la voiture que \underline{x} possède")

a) Traitement des pronoms

Dans le cas de phrases où figure un pronom comme argument du verbe, nous dirons que le domaine de définition correspondant au pronom est le même que celui de son antécédent.

Exemple 5 : a) un homme est venu ce matin,
il a apporté cette lettre

b) un(x.homme(x), x.est venu ce matin(x))
pron(x.vrai, x. a apporté cette lettre(x)).

Les phrases a) représentées par les structures profondes b) se traduisent par :

i est venu ce matin(x) + x ∈ E₁

a ∈ E₁

ii a apporté cette lettre(x) + x ∈ E₂

x ∈ E₂ + x ∈ E₁

b) Portée des quantificateurs

Une relation du type (x,y) ∈ E peut se définir à partir de deux propriétés élémentaires par :

$$(\underline{x}, \underline{y}) \in E + \underline{x} \in E_1 \wedge \underline{y} \in E_2(\underline{x})$$

Ceci suppose que le deuxième ensemble ne peut être défini qu'en fonction du premier, ce qui n'a pas nécessairement toujours lieu :

Exemple 6 : a) aucun étudiant n'a résolu le problème

b) chaque(x.étudiant(x), x.le(y.problème(y),
y.non.résolu(x,y)))

Dans ce cas les énoncés élémentaires doivent être :

i non.résolu(x,y) + (x,y) ∈ E

ii (x,y) ∈ E + x ∈ E₁ ∧ y ∈ E₂

iii x ∈ E₁ + étudiant(x)

iv a ∈ E₂ (a représente "le problème")

les ensembles E₁ et E₂ étant alors totalement indépendants.
C'est également le cas pour des phrases du type

chaque x voit chaque y.

Mais l'occurrence d'un pronom peut modifier la portée des quantificateurs.

Exemple 7 :

a) chaque homme épouse la femme qu'il aime

b) chaque $(x, \text{homme}(x), x, \text{le}(y, \text{et}(\text{femme}(y), \text{pron}(z, \text{vrai}, z, \text{aime}(z, y))), y, \text{épouse}(x, y)))$

i $\text{épouse}(\underline{x}, \underline{y}) \leftarrow (\underline{x}, \underline{y}) \in E$

ii $(\underline{x}, \underline{y}) \in E \leftarrow \underline{x} \in E_1 \wedge \underline{y} \in E_2$

iii $\underline{x} \in E_1 \leftarrow \text{homme}(\underline{x})$

ivi $f(\underline{z}) \in E_2 \leftarrow \underline{z} \in E_3$

v $\underline{z} \in E_3 \leftarrow \underline{z} \in E_1$

La règle i se transforme donc en :

$\text{épouse}(\underline{x}, f(\underline{z})) \leftarrow \underline{x} \in E_1 \wedge \underline{z} \in E_1$

qui n'est correcte que si on "force la fusion (merge) entre $\underline{x} \in E_1$ et $\underline{z} \in E_1$ ", ce qui doit être fait par une définition correcte de ii.

Dans les mêmes conditions l'exemple 4 peut être traduit par :

i $\text{possède}(\underline{x}, \underline{y}) \leftarrow (\underline{x}, \underline{y}) \in E$

ii $(\underline{x}, \underline{y}) \in E \leftarrow \underline{x} \in E_1 \wedge \underline{y} \in E_2$

iii $\underline{x} \in E_1 \leftarrow \text{garçon}(\underline{x})$

iv $f(\underline{x}) \in E_2 \leftarrow \underline{x} \in E_1$

i donne alors : $\text{possède}(\underline{x}, f(\underline{z})) \leftarrow \underline{x} \in E_1 \wedge \underline{z} \in E_1$

soit : $\text{possède}(\underline{x}, f(\underline{x})) \leftarrow \underline{x} \in E_1$ si "l'on force la fusion"

2°/ ENONCES ELEMENTAIRES TRADUISANT UNE PHRASE DECLARATIVE

A chaque argument de la phrase, c'est-à-dire à chaque structure

$q(p, r)$

est associé un ensemble de définition E. Pour donner un nom différent à chacun de ces ensembles (ou propriétés) nécessaires pour exprimer une phrase, on associe à chaque noeud $q(p, r)$ de l'arbre un numéro différent.

La numérotation adoptée est la suivante :

$i - d(i)$	$g(i)$ pour le <u>fil</u> s d'un noeud i
↓	
$g(i)$	$d(i)$ pour le <u>frère</u> d'un noeud i

La racine de l'arbre représentant la phrase porte le numéro n de la phrase. Etant donné un noeud $q(p,r)$ portant le numéro i , nous appellerons i l'ensemble de définition associé à ce noeud.

Deux types d'énoncés élémentaires sont à considérer :

- les énoncés exprimant la relation décrite par le verbe de la phrase.

Cette relation qoi , comme dans l'exemple 7, a en général la structure suivante :

épouse $(x,y) \leftarrow (x,y) \in E$

$(x,y) \in E \leftarrow x \in E_1 \wedge y \in E_2$

se traduit par l'énoncé élémentaire suivant :

épouse $(x,y) \leftarrow$ résoud $((i,x).(j,y).nil,nil,V)$

si i et j sont les noms des ensembles E_1 et E_2 .

Dans le cas où le verbe s'exprime par un énoncé négatif (sans prédicat positif) un prédicat absurde est systématiquement ajouté :

Exemple 8 : Pierre ne vient pas

se traduit par :

absurde \leftarrow vient(Pierre).

- les "définitions des domaines" associées aux arguments.
Une formule générale de la forme :

$x \in E_1 \leftarrow P(x,y,z) \wedge y \in E_2 \wedge z \in E_3$

se traduit par l'énoncé élémentaire :

$S_n(1,x,(2,y).(3,z).nil) \leftarrow P(x,y,z)$

si 1, 2, 3 sont les noms donnés aux ensembles E_1, E_2, E_3 .

Il reste alors à définir le prédicat résoud par les règles suivantes :

$$\text{résoud}(\underline{\text{nil}}, \underline{\text{v}}, \underline{\text{v}})$$

$$\text{résoud}(\underline{\text{x}}, \underline{\text{y}}, \underline{\text{v}}, \underline{\text{w}}) \leftarrow \text{dans}(\underline{\text{x}}, \underline{\text{v}}, \text{oui}) \wedge \text{résoud}(\underline{\text{y}}, \underline{\text{v}}, \underline{\text{w}})$$

$$\text{résoud}(\underline{\text{x}}, \underline{\text{y}}, \underline{\text{v}}, \underline{\text{w}}) \leftarrow \text{dans}(\underline{\text{x}}, \underline{\text{v}}, \text{non}) \wedge \text{sn}(\underline{\text{x}}, \underline{\text{u}}) \wedge \\ \text{résoud}(\underline{\text{u}}, \underline{\text{v}}, \underline{\text{v}}_1) \wedge \text{résoud}(\underline{\text{y}}, \underline{\text{x}}, \underline{\text{v}}_1, \underline{\text{w}})$$

avec la définition suivante du prédicat dans

$$\text{dans}(\underline{\text{i}}, \underline{\text{x}}, (\underline{\text{i}}, \underline{\text{x}}). \underline{\text{v}}, \text{oui})$$

$$\text{dans}(\underline{\text{i}}, \underline{\text{x}}, (\underline{\text{j}}, \underline{\text{y}}). \underline{\text{v}}, \underline{\text{w}}) \leftarrow \text{dif}(\underline{\text{i}}, \underline{\text{j}}) \wedge \text{dans}(\underline{\text{i}}, \underline{\text{x}}, \underline{\text{v}}, \underline{\text{w}})$$

$$\text{dans}(\underline{\text{i}}, \underline{\text{x}}, \text{nil}, \text{non}).$$

Exemple 9 :

- a) chaque homme qui possède un bateau l'entretient
 b) chaque(x.et.(homme(x).un(y.bateau(y).y.possède(x,y))),
 x.pron(z.vrai,z.entretient(x,z)))

La phrase 9a) représentée par la structure profonde 9 b) se traduit par :

$$\text{entretient}(\underline{\text{x}}, \underline{\text{z}}) \leftarrow \text{résoud}(1, \underline{\text{x}}). (3, \underline{\text{z}}). \text{nil}, \text{nil}, \underline{\text{y}})$$

$$\text{sn}(1, \underline{\text{x}}, (2, \underline{\text{y}}). \text{nil}) \leftarrow \text{homme}(\underline{\text{x}}) \wedge \text{possède}(\underline{\text{x}}, \underline{\text{y}})$$

$$\text{sn}(2, \underline{\text{y}}, \text{nil}) \leftarrow \text{bateau}(\underline{\text{y}})$$

$$\text{sn}(3, \underline{\text{z}}, (\underline{\text{i}}, \underline{\text{z}}). \text{nil}) \leftarrow \text{ant}(3, \underline{\text{i}})$$

$$\text{ant}(3, \underline{\text{z}})$$

Le fait que le pronom "le" dans la phrase 9 a) a pour antécédent "un bateau" a pour effet de modifier la portée de ce quantificateur. La formule logique correspondante est :

$$\forall x \forall y \quad \text{homme}(x) \wedge \text{bateau}(y) \wedge \text{possède}(x,y) \rightarrow \text{entretient}(x,y)$$

Des phénomènes analogues de modification de portée des quantificateurs apparaissent lors d'occurrences de pronoms dans une phrase dont la référence se situe dans une autre phrase.

3°/ ENONCES ELEMENTAIRES ASSOCIES AUX PHRASES INTERROGATIVES

Nous pouvons maintenant compléter la structure des énoncés élémentaires pour permettre le traitement des questions du type "qui...".

Les questions de ce type sont représentées par des structures du type

$$\text{intero}(p,q).$$

Trouver la réponse à une telle question consiste à :

- affirmer que pour tout x on a

$$\text{non} \cdot \text{et} \cdot a(p,\underline{x}) \cdot a(q,\underline{x})$$

$a(p,\underline{x})$ représente le résultat de l'opération qui consiste à appliquer la λ -expression p sur la variable \underline{x} .

- faire la preuve de l'inconsistance du système, la réponse étant donnée par la valeur de la variable \underline{x} , c'est-à-dire que la réponse apparaît comme le résultat d'une unification.

On modifiera donc la structure des énoncés élémentaires dans ce sens, en ajoutant à chaque prédicat un argument supplémentaire r (appelé réponse) qui permet s'il y a inconsistance de calculer la réponse éventuelle.

Exemple 10 : a) un homme vient

b) qui vient ?

10 a) et 10 b) se traduisent par :

i $\text{sn}(1.a,\text{nil},\underline{r})$

$\text{vient}(\underline{x},\underline{r}) + \text{résoud}(\{1.\underline{x}\};\text{nil},\text{nil},\underline{v},\underline{r})$

ii $\text{sn}(2.\underline{x},\text{nil},\text{rep}(\text{qui}(\underline{x})))$

$\text{absurde}(\underline{r}) + \text{vient}(\underline{x},\underline{r}) \wedge \text{résoud}(\{2.\underline{x}\};\text{nil},\text{nil},\underline{v},\underline{r})$

De i et ii on peut inférer $\text{absurde}(\text{rep}(\text{qui}(a)))$, ce qui signifie que le système est inconsistant pour $\underline{r}=\text{rep}(\text{qui}(a))$, c'est-à-dire que la réponse à la question "qui..." est a .

Nous concluons à une contradiction dans le système si l'on peut inférer

$\text{absurde}(\underline{r})$ pour tout \underline{r} .

Remarque : Considérons une question négative :

Qui ne vient pas ?

La traduction est alors :

$\text{sn}(1.\underline{x},\text{nil},\text{rep}(\text{qui}(\underline{x})))$

$\text{vient}(\underline{x},\underline{r}) + \text{résoud}(\{1.\underline{x}\};\text{nil},\text{nil},\underline{v},\underline{r})$

Nous ne traiterons pas ici des questions du type "pourquoi..." dont la réponse consiste à donner le détail de la preuve d'inconsistance. Pour cela on mémorise l'ensemble des énoncés élémentaires utilisés dans la preuve.

4°/ STRUCTURE DES ENONCES ELEMENTAIRES

```

<énoncé élémentaire> ::= en(<couple>,<liste de couples>,<réponse>).
      <fin de clause> | <tête de clause>.<résoud>.<fin de clause>
<tête de clause> ::= absurde(<réponse>) | <prédicat de base>
<fin de clause> ::= nil | <prédicat de base>.<fin de clause>
<résoud> ::= résoud(<liste de couples>,<liste de couples>,
      <liste de couples>,<réponse>)
<prédicat de base> ::= pr(<arguments pr>,<réponse>) |
      ant(<numéro>,<numéro>)
<couple> ::= <numéro>.<individu> | <variable>
<numéro> ::= <variable> | <entier> | d(<numéro>) | g(<numéro>)
<individu> ::= <variable> | <genre-nb>.ch(<propriété>) |
      <genre-nb>.<variable>
<arguments pr> ::= <individu>.<individu>.<individu>.<verbe 3>.nil |
      <individu>.<individu>.<verbe 2>.nil |
      <individu>.<individu>.dans.nil |
      <individu>.<nom propre>.nom.nil |
      <individu>.<verbe 1>.nil |
      <verbe 0>.nil
<réponse> ::= <variable> | rep(qui(<individu>)) | rep(pourquoi) |
      rep(faux)
<liste de couples> ::= nil | <variable> | <couple>.<liste de couples>
<variable> ::= "toute variable PROLOG"

```

Un énoncé élémentaire est défini comme une liste de prédicats

$$P_1.P_2 \dots .P_n.nil$$

signifiant :

$$P_1 \wedge P_2 \wedge \dots \wedge P_n$$

Etant donné un ensemble S d'énoncés élémentaires, nous dirons que ce système est inconsistant si et seulement si

$\forall r$ absurde(r) + S

ce qui revient à annuler sur S le prédicat négatif

absurde(rep(faux))

IV - PRODUCTION DES ENONCES ELEMENTAIRES

Nous donnons maintenant les règles permettant d'engendrer l'ensemble des énoncés élémentaires traduisant une phrase.

1°/ PRODUCTION DES ENONCES EXPRIMANT LA RELATION DECRITE PAR LE VERBE

Cela peut être décrit en terme de règles de réécriture à l'aide du prédicat :

$$\text{not}(A, u, v, i, r, S)$$

- où - A est la structure profonde d'une phrase,
 - S un énoncé élémentaire,
 - i un numéro,
 - u et v deux listes de couples,
 - r une réponse,

signifiant : l'arbre A de numéro i se réécrit en S; v est alors la liste u augmentée des couples i, x associés à chaque variable libre occurant dans S, et r la réponse associée.

par les règles suivantes :

$$\text{not}(\text{pourquoi}, p, u, v, i, \text{rep}(\text{pourquoi}), s) \leftarrow \text{not}(\text{non}, p, u, v, i, \text{rep}(\text{pourquoi}), s)$$

$$\text{not}(\text{inter}, x, p, q, u, v, i, r, s) \leftarrow \text{not}(\text{non}, \text{et}, a(p, y), a(q, y), (i, y), u, v, i, r, s)$$

$$\text{not}(t, x, p, q, u, v, i, r, s) \leftarrow \text{not}(a(q, y), (i, y), u, v, d(i), r, s) \wedge \text{quant}(t)$$

$$\text{not}(\text{un}, x, p, q, u, v, i, r, s) \leftarrow \text{not}(a(p, y), (i, y), \text{nil}, v, g(i), r, s)$$

$$\text{not}(\text{pr}(x), u, u, i, r, \text{pr}(x, r), \text{résoud}(u, \text{nil}, v, r), \text{nil})$$

$$\text{not}(\text{non}, \text{pr}(x), u, u, i, r, \text{absurde}(r), \text{résoud}(u, \text{nil}, v, r), \text{pr}(x, r), \text{nil})$$

$$\text{not}(\text{non}, \text{vrai}, u, u, i, r, \text{absurde}(r), \text{résoud}(u, \text{nil}, v, r), \text{nil})$$

$$\text{not}(\text{et}, p, q, u, v, i, r, s) \leftarrow \text{not}(p, u, v, g(i), r, s)$$

$$\text{not}(\text{et}, p, q, u, v, i, r, s) \leftarrow \text{not}(q, u, v, d(i), r, s)$$

$$\text{not}(\text{ou}, p, q, u, v, i, r, s) \leftarrow \text{not}(p, u, w, g(i), r, s_1) \wedge \text{not}(q, w, v, d(i), r, s_2) \wedge c(s_1, s_2, s)$$

$$\text{not}(p, u, v, i, r, s) \leftarrow t(p, q) \wedge \text{not}(q, u, v, i, r, s)$$

Le prédicat $\text{quant}(q)$ est vraissi q est égal à un, le, chaque ou pron.

Le prédicat $c(s_1, s_2, s)$ "combine" les deux énoncés élémentaires s_1 et s_2 pour en obtenir un troisième, s .
Cela n'est possible que si s_1 ou s_2 est de la forme

absurde(r).s

Le prédicat t est défini par :

$t(\text{aucun.p}, \text{non.un.p})$

$t(\text{non.aucun.p}, \text{un.p})$

$t(\text{non.s.x.p.q}, \text{t.x.p.non.q}) \leftarrow \text{homol}(\underline{s}, \underline{t})$

$t(\text{non.et.p.q}, \text{ou}(\text{non.p}), \text{non.q})$

$t(\text{non.ou.p.q}, \text{et}(\text{non.p}), \text{non.q})$

$t(\text{non.non.p}, \underline{p})$

$t(\underline{a}(\underline{p}, \underline{x}), \underline{q}) \leftarrow \text{app}(\underline{p}, \underline{x}, \underline{q})$

$t(\text{non.a}(\underline{p}, \underline{x}), \text{non.q}) \leftarrow \text{app}(\underline{p}, \underline{x}, \underline{q})$

avec :

$\text{homol}(\text{un}, \text{chaque})$

$\text{homol}(\text{chaque}, \text{un})$

$\text{homol}(\text{le}, \text{le})$

$\text{homol}(\text{pron}, \text{pron})$

et :

$\text{app}(p, x, q)$ vrai ssi q est le résultat de l'application de x sur la λ -expression p .

Si A est la structure profonde d'une phrase n° 1, la relation décrite par le verbe est alors traduite par l'ensemble des énoncés élémentaires S qui vérifient :

$\text{not}(A, \text{nil}, u, i, r, S)$

2°/ PRODUCTION DES ENONCES ELEMENTAIRES DEFINISSANT LES DOMAINES

Nous la décrirons également en terme de règles de réécriture à l'aide du prédicat :

$$d(A, u, i, S)$$

signifiant : l'arbre A de numéro i et dont la liste des variables libres est donnée par la liste de couple u se réécrit en S. par les règles suivantes :

$$d(\text{intero. } \underline{x.p.q.u.i}, \text{sn}(\underline{i.x.y}, \text{nil}, \text{rep}(\text{qui}(\underline{x.y})))) \cdot \text{nil}$$

$$d(\text{chaque. } \underline{x.p.q.u.i}, \text{sn}(\underline{i.x.y.v.r}, \underline{s}) + \text{not}(\text{non.a}(\underline{p.x.y}, \text{nil}, \underline{v}, \underline{g(i)}, \underline{r}, \text{absurde}(\underline{r}).\text{resoud}(\underline{v}, \text{nil}, \underline{w}, \underline{r}).\underline{s}))$$

$$d(\text{un. } \underline{x.p.q.u.i}, \text{sn}(\underline{i.x.y.v.r}, \text{nil})) + \text{choix}(\underline{y.z.et.a}(\underline{p.z}).\underline{a}(\underline{q.z}), \underline{i}, \underline{u}, \underline{v})$$

$$d(\text{le. } \underline{x.p.q.u.i}, \text{sn}(\underline{i.x.y.v.r}, \text{nil})) + \text{choix}(\underline{y.p.g(i)}, \text{nil}, \underline{v})$$

$$d(\text{pron. } \underline{x.p.q.u.i}, \text{sn}(\underline{i.x.y}, (\underline{j.x.y}), \text{nil}, \underline{r}).\underline{\text{ant}}(\underline{i}, \underline{j}).\text{nil})$$

$$d(\underline{p.u.i.s}) + \underline{t(p,q)} \wedge d(\underline{q.u.i.s})$$

Le prédicat choix est défini par :

$$\text{choix}(\underline{\text{ch}}(\underline{x}), \underline{p.i.u.v}) + \underline{\text{tch}}(\underline{x.p.i.u.v})$$

$$\underline{\text{tch}}(\underline{t.x.p_1.q_1}, \underline{t.x.p.q.i.u.v}) + \underline{\text{quant}}(\underline{t}) \wedge \underline{\text{dif}}(\underline{t}, \text{pron}) \wedge \underline{\text{tch}}(\underline{p_1.p.g(i)}, \underline{u.w}) \wedge \underline{\text{tch}}(\underline{q_1.q.d(i)}, \underline{w.v})$$

$$\underline{\text{tch}}(\underline{r.pron.x.p.q.i.u.v}) + \underline{\text{tch}}(\underline{r.a(q.y)}, \underline{d(i)}, (\underline{i.y}).\underline{u.v})$$

$$\underline{\text{tch}}(\underline{c.p_1.q_1}, \underline{c.p.q.i.u.v}) + \underline{\text{tch}}(\underline{p_1.p.g(i)}, \underline{u.w}) \wedge \underline{\text{tch}}(\underline{q_1.q.d(i)}, \underline{w.v}) \wedge \underline{\text{connect}}(\underline{c})$$

$$\underline{\text{tch}}(\underline{\text{pr}}(\underline{x}), \underline{\text{pr}}(\underline{x}), \underline{i}, \underline{u}, \underline{u})$$

$$\underline{\text{tch}}(\underline{\text{non.pr}}(\underline{x}), \underline{\text{non.pr}}(\underline{x}), \underline{i}, \underline{u}, \underline{u})$$

$$\underline{\text{tch}}(\underline{\text{vrai}}, \underline{\text{vrai}}, \underline{i}, \underline{u}, \underline{u})$$

$$\underline{\text{tch}}(\underline{\text{non.vrai}}, \underline{\text{non.vrai}}, \underline{i}, \underline{u}, \underline{u})$$

$$\underline{\text{tch}}(\underline{p_1.p.i.u.v}) + \underline{t(p,q)} \wedge \underline{\text{tch}}(\underline{p_1.q.i.u.v})$$

$\text{ch}(x)$ est en fait la "fonction de Skolem" vérifiant la formule $\exists p$ dont la liste des variables libres est donnée par la liste de couples u. x est défini comme étant égal à un arbre p' obtenu à partir de p en substituant dans p chaque occurrence d'un sous-arbre n° i de la forme :

$$\text{pron. } \underline{y.q.e}$$

par : $\underline{a(r,z)}$ avec \underline{v} alors égal à $\underline{(i.z).u}$

Le prédicat connect(c) est vrai ssi c est un connecteur.

Si A est un arbre (ou un sous-arbre) de numéro i, la définition de domaine associée au noeud correspondant à la racine de A est alors donnée par l'ensemble des énoncés élémentaires S qui vérifient :

$$d(A,u,i,S)$$

si u est la liste de couples associée à chaque variable libre de A.

3°/ PRESUPPOSITION

Les présuppositions que nous considérons sont celles associées à l'emploi du défini ou d'un pronom. Leur génération est décrite par des règles de réécritures à l'aide du prédicat :

$$p(A, i, S)$$

signifiant que l'arbre A de numéro i se réécrit en S.

Ces règles sont les suivantes :

$$p(\text{le. } \underline{x.p.q.i.s}) \leftarrow \text{not}(a(\underline{p.y}), (\underline{i.y}).\text{nil.v.g}(\underline{i}), \underline{r.s})$$

$$p(\text{le. } \underline{x.p.q.i}, \text{pr}(\underline{y.(x.z)}. \text{dans.nil.r}). \underline{s}) \leftarrow \text{choix}(\underline{z.p.g}(\underline{i}), \text{nil.v}) \wedge \text{not}(\text{non.a}(\underline{p.y}), \underline{v.u.g}(\underline{i}), \underline{r. \text{absurde}(\underline{r}).s})$$

$$p(\text{pron. } \underline{x.p.q.i.s}) \leftarrow \text{not}(a(\underline{p.y}), (\underline{i.y}).\text{nil.v.g}(\underline{i}), \underline{r.s})$$

$$p(\text{pron. } \underline{x.p.q.i}, \text{ant}(\underline{i.j}).\text{nil}) \leftarrow \text{antpos}(\underline{i.j})$$

$$p(\underline{q.i.s}) \leftarrow t(\underline{q.r}) \wedge p(\underline{r.i.s})$$

Le prédicat $\text{antpos}(i, j)$ est vrai si "le syntagme n° j est un antécédent possible du pronom n° i". Nous qualifions d'antécédent possible tout syntagme satisfaisant les critères syntaxiques dans la résolution des pronoms.

Si A est un arbre de numéro i, la présupposition associée au noeud correspondant à la racine de A est donnée par l'ensemble des énoncés élémentaires S qui vérifient :

$$p(A, i, S)$$

L'ensemble des énoncés élémentaires S correspondant aux définitions de domaines et aux présuppositions d'une phrase n° i représentée par la structure profonde A est défini comme l'ensemble des énoncés S qui vérifient :

$$\text{pres}(A, u, i, S)$$

avec u la liste de couples associée aux variables libres de A. Le prédicat pres est défini par les règles suivantes :

$\text{pres}(\text{pourquoi.p.u.i.s}) \leftarrow \text{pres}(\text{non.p.u.i.s})$
 $\text{pres}(\text{interro.x.p.q.u.i.s}) \leftarrow \text{pres}(\text{non.a(p.y),(i.y).nil,g(i),s})$
 $\text{pres}(\text{chaque.x.p.q.u.i.s}) \leftarrow \text{pres}(\text{non.a(p.y),(i.y).nil,g(i),s})$
 $\text{pres}(\text{un.x.p.q.u.i.s}) \leftarrow \text{pres}(\text{a(p.y),(i.y).nil,g(i),s})$
 $\text{pres}(\text{le.x.p.q.u.i.s}) \leftarrow \text{pres}(\text{a(p.y),(i.y).nil,g(i),s})$
 $\text{pres}(\text{pron.x.p.q.u.i.s}) \leftarrow \text{pres}(\text{a(p.y),(i.y).nil,g(i),s})$
 $\text{pres}(\text{t.p.u.i.s}) \leftarrow \text{quant}(\text{t}) \wedge \text{d}(\text{t.p.u.i.s})$
 $\text{pres}(\text{t.p.u.i.s}) \leftarrow \text{quant}(\text{t}) \wedge \text{p}(\text{t.p.i.s})$
 $\text{pres}(\text{interro.p.u.i.s}) \leftarrow \text{d}(\text{interro.p.u.i.s})$
 $\text{pres}(\text{t.x.p.q.u.i.s}) \leftarrow \text{quant}(\text{t}) \wedge \text{pres}(\text{a(q.y),(i.y).u.d(i),s})$
 $\text{pres}(\text{interro.x.p.q.u.i.s}) \leftarrow \text{pres}(\text{non.a(q.y),(i.y).u.d(i),s})$
 $\text{pres}(\text{c.p.q.u.i.s}) \leftarrow \text{connect}(\text{c}) \wedge \text{pres}(\text{p.u.g(i),s})$
 $\text{pres}(\text{c.p.q.u.i.s}) \leftarrow \text{connect}(\text{c}) \wedge \text{pres}(\text{q.u.d(i),s})$
 $\text{pres}(\text{p.u.i.s}) \leftarrow \text{t}(\text{p.q}) \wedge \text{pres}(\text{q.u.i.s})$

Définition

Etant donné un ensemble consistant S d'énoncés élémentaires décrivant un monde, et une phrase de numéro i représentée par la structure profonde A, nous noterons :

S_1 l'ensemble des énoncés élémentaires s tels que

$$\text{not}(A, \text{nil}, \underline{v}, i, r, s)$$

S_2 l'ensemble des énoncés élémentaires s tels que

$$\text{pres}(A, \text{nil}, i, s)$$

et nous dirons que la phrase est :

vraie ssi SUS_2 est consistant et SUS_1US_2 est consistant

fausse ssi SUS_2 est consistant et SUS_1US_2 est inconsistant

n'a pas de sens sinon

V - TRAITEMENT DES PRESUPPOSITIONS.1°/ RESOLUTION DES PRONOMS

La résolution des pronoms est "retardée" de manière à faire intervenir dans la recherche de l'antécédent des critères sémantiques et syntaxiques. Ceci ne concerne pas les pronoms relatifs ou réfléchis dont la résolution est purement syntaxique.

- Critères syntaxiques

Les critères syntaxiques sont de plusieurs natures.

* le genre et le nombre

* la position respective de l'antécédent et du pronom dans la structure profonde.

- le noeud correspondant au pronom ne doit pas être "frère" du noeud correspondant à l'antécédent.

Exemple 11 : un homme lui donne le livre

un (x.homme(x),x.pron(y.vrai,y.le(z.livre,z.donne(x,z,y)))

Le pronom lui ne peut référer ni à "homme" ni à "livre", cette contrainte n'étant due qu'à la position respective des arguments.

- deux pronoms dont les positions dans l'arbre sont "frère" ne peuvent avoir le même antécédent.

Exemple 12 : a) il le voit

b) l'homme qu'il regarde

Dans 12 a) il et le ne peuvent avoir le même antécédent,

dans 12 b) que et il ne peuvent avoir le même antécédent, c'est-à-dire que l'antécédent de il ne peut être l'homme.

Pour aider le démonstrateur, d'autres critères ont été introduits, limitant la "distance" entre le pronom et son antécédent et l'emploi à certains cas d'antécédents de type "chaque(p,q)" (antécédent et pronom dans la même phrase).

- Critères sémantiques

Les critères sémantiques consistent à dire que l'antécédent du pronom i n'est pas un syntagme j possible, si la présupposition associée au pronom n'est pas vraie.

Exemple 13 : a) Pierre n'aime pas Médor

b) Ce chien est pourtant affectueux.

Le pronom n'est pas ambigu, car si l'on suppose que l'antécédent est Pierre, la présupposition "Pierre est chien" est fausse (si l'on sait que "Pierre est homme" et que "aucun homme n'est un chien").

Si l'on remplace b) par :

c) il est pourtant affectueux,

l'ambiguïté existe alors vraiment.

Définition

Etant donné un ensemble consistant S d'énoncés élémentaires décrivant un monde, et un arbre A (ou un sous-arbre) de numéro 1 ayant la structure :

pron.x.p.r u la liste de couples des variables libres de A ,

nous noterons :

S_1 l'ensemble des énoncés élémentaires s tels que $d(A, u, i, s)$

S_2 l'ensemble des énoncés élémentaires s tels que $p(A, i, s)$

S_3 l'ensemble des énoncés élémentaires s tels que $\text{pres}(a(p, y), (i, y), \text{nil}, g(i), s)$

S_2 pouvant contenir plusieurs énoncés de la forme $\text{ant}(i, j)$:

$$S_2 = S_2' \cup \text{ant}(i, j_1) \cup \dots \cup \text{ant}(i, j_n)$$

nous considérons plusieurs ensembles possibles :

$$S_2^1 = S_2' \cup \text{ant}(i, j_1)$$

.....

$$S_2^n = S_2' \cup \text{ant}(i, j_n)$$

et nous dirons que la présupposition associée au pronom est :

vraie ssi SUS_3 est consistant et
il existe un et un seul k tel que
 $SUS_1US_2^kUS_3$ soit consistant
(l'antécédent du pronom i est alors le syntagme j^k)

fausse ssi SUS_3 est consistant et
pour tout k $SUS_1US_2^kUS_3$ est inconsistant
(le pronom n'a pas d'antécédent)

n'a pas de sens ssi SUS_3 est inconsistant.

Il y a ambiguïté dans le cas où il existe plusieurs indices k tels que

$SUS_1US_2^kUS_3$ est consistant

Dans ce cas le système pose des questions pour savoir lequel de ces antécédents j il faut choisir.

Remarque 1 :

Dans l'état actuel du système, une contrainte a dû être introduite affirmant que :

- un noeud correspondant au pronom ne doit pas être un descendant du noeud correspondant à son antécédent.

Ceci supprime les énoncés du genre :

Exemple 14 :

a) la femme qui regarde le miroir qu'elle tient

b) $ls(x.et.femme(x).ls(y.et.miroir(y).pron(z.vrai,z.tient(z,y)),y.regarde(x,y)),x....)$

En effet, avec la représentation actuelle des objets, "la femme qui regarde le miroir qu'elle tient" serait représenté par une "fonction de Skölem" x dépendant de l'antécédent y du pronom elle.

$x = f(y)$ avec ici $y = x$ puisque l'antécédent du pronom est justement "la femme ..."

Nous avons donc $x = f(x)$

ce qui pose le problème de l'unification de x avec $f(x)$.

Remarque 2 :

Dans la résolution retardée des pronoms, les présuppositions d'ordre général, qui expriment une restriction (sur le sujet par exemple) doivent orienter la recherche de l'antécédent.

Exemple 15 : Il s'attend à ce que Marie vienne.

Le pronom il, sujet du verbe "s'attendre à" présuppose que l'antécédent est humain.

Ce type de présupposition appartient aux définitions et aux axiomes de la sémantique du lexique.

2°/ TRAITEMENT DES PRESUPPOSITIONS ASSOCIEES A L'EMPLOI DU DEFINI

Dans l'emploi du défini, le traitement des présuppositions consiste à vérifier l'existence et l'unicité de l'objet quantifié par le.

Exemple 16 : a) la femme qui est assise sur le divan est belle

b) il y a une et une seule femme qui est assise sur le divan

c) il y a un et un seul divan.

La phrase 16 a) n'a de sens que si 16 b) est vraie.

La phrase 16 b) n'a de sens que si 16 c) est vraie.

La phrase 16 c) a toujours un sens.

Le système :

- si 16 c) est fausse, rejette toute l'information (16 c), b, a);
- sinon l'information 16 c) est conservée et :
 - si 16 b) est fausse, 16 b) et 16 c) sont rejetées;
 - sinon 16 b) est conservée.

Exemple 17 : a) la femme regarde le miroir

b) il y a une et une seule femme

c) il y a un et un seul miroir.

La présupposition 17 b) peut être fausse alors que 17 c) peut être vraie (et inversement).

Dans ce cas 17 b) est refusée ainsi que 17 a) et 17 c) est conservée.

Définition :

Etant donné un ensemble consistant S d'énoncés élémentaires, et un arbre A (ou un sous-arbre) de numéro 1 ayant la structure :

le.x.p.r u la liste de couples des variables libres de A

soit S_1 l'ensemble des énoncés élémentaires s tels que $d(A, u, i, s)$

S_2 l'ensemble des énoncés élémentaires s tels que $p(A, i, s)$

S_3 l'ensemble des énoncés élémentaires s tels que
près(a(p, y), (i, y), nil, g(i), s)

Nous dirons alors que la présupposition est :

vraie ssi SUS_3 est consistant et $SUS_1US_2US_3$ est consistant

fausse ssi SUS_3 est consistant et $SUS_1US_2US_3$ est inconsistant

n'a pas de sens sinon

VI - DEDUCTIONS.1°/ CONTROLE DES DEDUCTIONS

Les contraintes imposées au niveau des déductions pour limiter l'arbre de recherche sont de deux ordres principalement :

- contrainte qui consiste à n'avoir que des clauses régulières pour représenter les faits, cf(1). C'est-à-dire que chaque clause ne doit avoir au maximum qu'un seul littéral positif.

Ce type de contrainte permet de simplifier l'algorithme de résolution de J. A. ROBINSON tout en n'étant pas trop contraignant au niveau de l'utilisation du français. Si une phrase ne s'exprime pas par des clauses régulières, le système la refuse en signalant sa trop grande complexité;

- contrainte qui consiste à ne pas utiliser deux fois la même clause au cours du parcours d'une branche de l'arbre de recherche. Cette contrainte assure une "profondeur" finie à la preuve.

Pour réaliser cette dernière contrainte, on prévoit de donner à chaque énoncé élémentaire un numéro différent en fonction du numéro de la phrase ou de la présupposition dont il provient, et on intercale au milieu de chaque énoncé le prédicat :

$$\text{valide}(i,x)$$

- où - i est un numéro d'énoncé élémentaire;
 - x une liste de numéros d'énoncés élémentaires;

qui a pour valeur vrai ssi - i est une clause valide
 - i n'est pas déjà dans x.

Ainsi, tout énoncé $p \leftarrow q \wedge r$ de numéro i est transformé en :

$$p(x) \leftarrow \text{valide}(i,x) \wedge q(i,x) \wedge r(i,x)$$

C'est à l'aide du numéro i que l'on va pouvoir simuler la suppression d'un énoncé en affirmant que i n'est pas valide.

Par ailleurs, on a prévu d'insérer dans chaque littéral une variable u qui, en cours de démonstration accumule la liste de tous les numéros i d'énoncés intervenus dans cette démonstration; c'est à l'aide de l'information contenue dans u à la fin d'une preuve d'inconsistance que l'on peut :

- répondre aux questions de type "pourquoi..." puisqu'à partir de u on a accès à la liste des faits (phrases ou présuppositions) ayant servi dans la démonstration;
- supprimer, dans le cas d'une contradiction dans le système, le fait le plus ancien intervenant dans la démonstration.

2°/ REGLES DE RAISONNEMENT

Pour raisonner, la machine dispose essentiellement de "l'axiome du pluriel" qui, pour une relation binaire (aimer par exemple) se traduit par :

$$\begin{aligned} & \forall x_1 \exists y_1 x_1 \text{Cx} \rightarrow y_1 \text{Cy} \wedge x_1 \text{ aime } y_1 \\ x \text{ aime } y \equiv & \quad \quad \quad \wedge \\ & \forall y_1 \exists x_1 y_1 \text{Cy} \rightarrow x_1 \text{Cx} \wedge x_1 \text{ aime } y_1 \end{aligned}$$

x, y, x_1, y_1 étant des ensembles d'individus.

Cette définition d'une relation entre ensembles demande, bien entendu, à être modulée en fonction du type des verbes, mais a l'avantage de marquer une différence entre les quantificateurs les et chaque, tout en permettant de faire, dans le cas d'emploi de les, suffisamment de déductions.

De : les hommes aiment les femmes
et Pierre est un homme

on peut inférer :
Pierre aime des femmes.

Pour raisonner, la machine doit disposer également des axiomes de l'inclusion et de l'égalité.

3°/ CONTROLE DES ECHANGES

Que ce soit pour vérifier la validité d'un fait (phrase ou présupposition), pour trouver l'antécédent d'un pronom, ou pour répondre à une question, le système lance un "test d'inconsistance". Pour cela, il "cancèle" le littéral négatif absurde(r), et à partir de r calcule une réponse :

- s'il s'agit d'une réponse à une question posée, la réponse est formulée et la question est supprimée, mais non les présuppositions associées;

La question Qui connaît Pierre ?

présuppose l'existence de Pierre, existence qui sera conservée pour la suite du dialogue.

- s'il s'agit d'une contradiction dans le système, deux cas sont à envisager :

• le test a pour but de vérifier la validité d'une présupposition (pronom-défini) auquel cas la présupposition et la phrase associées sont supprimées;

• le test a pour but de vérifier la validité d'une phrase. Dans ce cas, on supprime les énoncés provenant des faits les plus anciens, intervenant dans la preuve de l'inconsistance.

Au sujet de la suppression des faits les plus anciens, on doit faire la remarque suivante : quand on supprime un fait f

- les présuppositions qui lui sont associées sont conservées,
- les faits pour lesquels f est une présupposition sont supprimés.

Exemple 18 : a) l'homme qui porte la valise est ...

b) il y a un et un seul homme qui porte la valise

c) il y a une et une seule valise.

La suppression de 18 b) entraîne la suppression de 18 a) alors que 18 c) est conservé.

EXEMPLES DE CONVERSATIONS

Voici quelques exemples de conversations. Les entrées de l'interlocuteur sont précédées d'un numéro et sont des structures profondes de phrases. Les sorties de la machine sont également données sous forme de structure profonde.

Le système est "lancé." à la console par l'interlocuteur à l'aide de la commande bonjour.

Nous donnons à la suite la traduction en français des structures profondes rentrées par l'interlocuteur en faisant correspondre leur numéro.

EXEMPLE 1

BØNJØUR..

A VØUS:

1 CHAQUE.(M.S).(L.*X.PR(*X.HØMME.NIL)).L.*Y.PR(*Y.MARCHER.NIL)..

A VØUS:

2 CHAQUE.(M.S).(L.*X.PR(*X.MARCHER.NIL)).L.*Y.PR(*Y.PRØGRESSER.NIL)..

A VØUS:

3 AUCUN.(M.S).(L.*X.PR(*X.RECULER.NIL)).L.*Y.PR(*Y.PRØGRESSER.NIL)..

A VØUS:

4 LE.(M.S).(L.*X.PR(*X.BØB.NØM.NIL)).L.*Y.PR(*Y.HØMME.NIL)..

A VØUS:

5 PØUPQUØ1.AUCUN.(M.S).(L.*X.PR(*X.HØMME.NIL)).L.*Y.PR(*Y.RECULER.NIL)..

PARCEQUE/ JE/ SAIS/ RAISONNER/ ET/ QUE/ VØUS/ M/'AVEZ/ DIT/:

PH(CHAQUE.(M.S).(L.*X1.PR(*X1.HØMME.NIL)).L.*X2.PR(*X2.MARCHER.NIL))

PH(CHAQUE.(M.S).(L.*X1.PR(*X1.MARCHER.NIL)).L.*X2.PR(*X2.PRØGRESSER.NIL))

PH(AUCUN.(M.S).(L.*X1.PR(*X1.RECULER.NIL)).L.*X2.PR(*X2.PRØGRESSER.NIL))

A VØUS:

6 PRØN.(M.S).(L.*X.PR(*X.RECULER.NIL)).L.*Y.PR(*Y.BEAU.NIL)..

JE/ NE/ COMPEND/ PAS/ A/ QUI/ SE/ REFERE/ LE/ PRNØM/ /:

SN(PRØN.(M.S).L.*X1.PR(*X1.RECULER.NIL))

J/'OUBLIE/ CETTE/ PHRASE/!

A VØUS:

7 AUREVØIR..

AU/ REVØIR/!

EXEMPLE 2

BØNJØUR..

A VØUS:

1 LE.(M.P).(L.*X.PR(*X.HØMME.NIL)).L.*Y.LE.(F.P).(L.*Z.PR(*Z.FEMME.NIL)).
L.*U.PR(*Y.*U.AIMER.NIL)..

A VØUS:

2 LE.(M.S).(L.*X.PR(*X.BØB.NØM.NIL)).L.*Y.PR(*Y.HØMME.NIL)..

A VØUS:

3 INTERØ.(*G).(L.*X.VRAI).L.*Y.PRØN.(M.S).(L.*Z.VRAI).L.*U.PR(*U.*Y.AIMER.
NIL)..

UNE/ FEMME/!

A VØUS:

4 POURQUØ1.PRØN.(M.S).(L.*X.VRAI).L.*Y.UN.(F.S).(L.*Z.PR(*Z.FEMME.NIL)).
L.*U.PR(*Y.*U.AIMER.NIL)..

PARCEQUE/ JE/ SAIS/ RAISONNER/ ET/ QUE/ VØUS/ M/'AVEZ/ DIT/:
PH(LE.(M.S).(L.*X1.PR(*X1.BØB.NØM.NIL)).L.*X2.PR(*X2.HØMME.NIL))
PH(LE.(M.P).(L.*X1.PR(*X1.HØMME.NIL)).L.*X2.LE.(F.P).(L.*X3.PR(*X3.FE
MME.NIL)).L.*X4.PR(*X2.*X4.AIMER.NIL))

A VØUS:

5 PRØN.(M.S).(L.*X.VRAI).L.*Y.AUCUN.(F.S).(L.*Z.PR(*Z.FEMME.NIL)).L.*U.
PR(*Y.*U.AIMER.NIL)..

VØUS/ VØUS/ CØNTREDISEZ/.J/'ØUBLIE/ QUE/ VØUS/ M/'AVEZ/ DIT/:
PH(LE.(M.P).(L.*X1.PR(*X1.HØMME.NIL)).L.*X2.LE.(F.P).(L.*X3.PR(*X3.FE
MME.NIL)).L.*X4.PR(*X2.*X4.AIMER.NIL))

A VØUS:

6 AUCUN.(M.S).(L.*X.PR(*X.HØMME.NIL))/1.L.*Y.VRAI..

VØUS/ VØUS/ CØNTREDISEZ/.J/'ØUBLIE/ QUE/ VØUS/ M/'AVEZ/ DIT/:
PH(LE.(M.S).(L.*X1.PR(*X1.BØB.NØM.NIL)).L.*X2.PR(*X2.HØMME.NIL))

VØUS/ VØUS/ CØNTREDISEZ/.J/'ØUBLIE/ QUE/:
SN(LE.(M.P).L.*X1.PR(*X1.HØMME.NIL))/ EXISTE/.

A VØUS:

7 LE.(M.S).(L.*X.PR(*X.HØMME.NIL)).L.*Y.PR(*Y.VENIR.NIL)..

VØUS/ SUPPØSEZ/ L/'EXISTENCE/ DE/:
SN(LE.(M.S).L.*X1.PR(*X1.HØMME.NIL))
CELA/ EST/ ABSURDE/.J/'ØUBLIE/ CETTE/ PHRASE/.

A VØUS:

8 AUREVØIR..

AU/ REVØIR/!

EXEMPLE 3

BONJOUR..

A VØUS:

1 LE.(M.S).(L.*X.PR(*X.CHAT.NIL)).L.*Y.PR(*Y.BEAU.NIL)..

A VØUS:

2 PRØN.(M.S).(L.*X.PR(*X.MATØU.NIL)).L.*Y.PR(*Y.MIAULER.NIL)..

A VØUS:

3 AUCUN.(M.S).(L.*X.PR(*X.MATØU.NIL)).L.*Y.VRAI..

VØUS/ VØUS/ CØNTREDISEZ// J/'ØUBLIE/ QUE/ VØUS/ M/'AVEZ/ DIT/ /:
 SN(PRØN.(M.S).L.*X1.PR(*X1.MATØU.NIL))
 EN/ PARLANTLE/ /:
 SN(LE.(M.S).L.*X1.PR(*X1.CHAT.NIL))

A VØUS:

4 INTERØ.(M.S).(L.*X.VRAI).L.*Y.PR(*Y.MIAULER.NIL)..

JE/ NE/ SAIS/ PAS/!

A VØUS:

5 INTERØ.(M.S).(L.*X.VRAI).L.*Y.PR(*Y.BEAU.NIL)..

LE/ CHAT/!

A VØUS:

6 LE.(M.S).(L.*X.PR(*X.CHIEM'N.NIL)).L.*Y.PR(*Y.VENIR.NIL)..

A VØUS:

7 LE.(M.S).(L.*X.PR(*X.CHAT.NIL)).L.*Y.PR(*Y.VENIR.NIL)..

A VØUS:

8 PRØN.(M.S).(L.*X.VRAI).L.*Y.NØN.PR(*Y.VENIR.NIL)..

L/'ANTECEDENT/ DU/ PRØNØM/ /:
 SN(PRØN.(M.S).L.*X1.VRAI)
 EST/ BIEN/ LE/ SYNTAGME/ /:
 SN(LE.(M.S).L.*X1.PR(*X1.CHAT.NIL))

A VØUS:

9 ØUI..

VØUS/ VØUS/ CØNTREDISEZ// J/'ØUBLIE/ QUE/ VØUS/ M/'AVEZ/ DIT/ /:
 PH(LE.(M.S).(L.*X1.PR(*X1.CHAT.NIL)).L.*X2.PR(*X2.VENIR.NIL))

A VØUS:

10 PRØN.(M.S).(L.*X.VRAI).L.*Y.PR(*Y.PARTIR.NIL)..

A VØUS:

11 INTERØ.(M.S).(L.*X.VRAI).L.*Y.PR(*Y.VENIR.NIL)..

LE/ CHIEN/!

A VOUS:

12 INTERØ.(M.S).(L.*X.VRAI).L.*Y.NØN.PR(*Y.VENIR.NIL)..

LE/ CHAT/!

A VOUS:

13 INTERØ.(M.S).(L.*X.VRAI).L.*Y.PR(*Y.PARTIR.NIL)..

LE/ CHAT/!

A VOUS:

14 CHAQUE.(M.S).(L.*X.ET.PR(*X.HØMME.NIL).UN.(M.S).(L.*Z.PR(*Z.CHIEN.NIL))
.L.*U.PR(*X.*U.PØSSEDER.NIL)).L.*T.PRØN.(M.S).(L.*V.VRAI).L.*S.PR
*T.*S.AIMER.NIL)..

A VOUS:

15 NØN.LE.(M.S).(L.*X.PR(*X.PIERRE.NØM.NIL)).L.*Y.LE.(M.S).(L.*Z.ET.PR(
*Z.CHIEN.NIL).PRØN.(M.S).(L.*V.VRAI).L.*S.PR(*S.*Z.PØSSEDER.NIL)).L.
*U.PR(*Y.*U.AIMER.NIL)..

A VOUS:

16 INTERØ.(M.S).(L.*X.VRAI).L.*Y.UN.(M.S).(L.*Z.PR(*Z.CHIEN.NIL)).L.*U.
PR(*Y.*U.PØSSEDER.NIL)..

PIERRE/!

A VOUS:

17 PRØN.(M.S).(L.*X.VRAI).L.*Y.PR(*Y.HØMME.NIL)..

L/'ANTECEDENT/ LU/ PRØNØM/ /:

SN(PRØN.(M.S).L.*X1.VRAI)

EST/ BIEN/ LE/ SYNTAGME/ /:

SN(LE.(M.S).L.*X1.ET.PR(*X1.CHIEN.NIL).PRØN.(M.S).(L.*X2.VRAI).L.*X3.
PR(*X3.*X1.PØSSEDER.NIL))

A VOUS:

18 NØN..

VOUS/ VOUS/ CØNTREDISEZ/.J/'ØUBLIE/ QUE/ VOUS/ M/'AVEZ/ DIT/:

PH(CHAQUE.(M.S).(L.*X1.ET.PR(*X1.HØMME.NIL).UN.(M.S).(L.*X2.PR(*X2.CH
IEN.NIL)).L.*X3.PR(*X1.*X3.PØSSEDER.NIL)).L.*X4.PRØN.(M.S).(L.*X5.VRA
I).L.*X6.PR(*X4.*X6.AIMER.NIL))

A VOUS:

19 AUCUN.(M.S).(L.*X.PR(*X.PIEE+RRE.NØM.NIL)).L.*Y.VRAI..

VOUS/ VOUS/ CØNTREDISEZ// J/'ØUBLIE/ QUE/:

SN(LE.(M.S).L.*X1.PR(*X1.PIERRE.NØM.NIL))/ EXISTE/.

A VOUS:

20 INTERØ.(M.S).(L.*X.VRAI).L.*Y.PR(*Y.HØMME.NIL)..

JE/ NE/ SAIS/ PAS/!

A VOUS:

21 AUREVØIR..

AU/ REVØIR/!

Voici la traduction en français des structures profondes rentrées par l'interlocuteur :

Exemple 1

- 1 chaque homme marche.
- 2 chaque "individu" qui marche progresse.
- 3 aucun "individu" qui recule ne progresse.
- 4 bob est un homme.
- 5 pourquoi aucun homme ne recule?
- 6 celui qui recule est beau.

Exemple 2

- 1 les hommes aiment les femmes.
- 2 bob est un homme.
- 3 qui aime-t-il ?
- 4 pourquoi aime-t-il une femme ?
- 5 il n'aime aucune femme.
- 6 il n'y a pas d'homme.
- 7 l'homme vient.

Exemple 3

- 1 le chat est beau.
- 2 ce matou miaule.
- 3 il n'y a pas de matou.
- 4 qui miaule ?
- 5 qui est beau ?
- 6 le chien vient.
- 7 le chat vient.
- 8 il ne vient pas.
- 9 oui
- 10 il part.
- 11 qui vient ?
- 12 qui ne vient pas ?
- 13 qui part ?
- 14 chaque homme qui possède un chien l'aime.
- 15 pierre n'aime pas le chien qu'il possède.
- 16 qui possède un chien ?
- 17 il est homme.
- 18 non
- 19 personne ne se nomme pierre.
- 20 qui est homme ?

BIBLIOGRAPHIE

- (1) COLMERAUER A., Les grammaires de métamorphoses, Groupe d'Intelligence Artificielle, UER de LUMINY, Université d'AIX-MARSEILLE, novembre 1975.
- (2) COLMERAUER A., KANOUI H., PASERO R. et ROUSSEL Ph., Un système de communication homme-machine en français, rapport de recherche, Groupe d'Intelligence Artificielle, UER de LUMINY, Université d'AIX-MARSEILLE, juin 1973.
- (3) KEENAN E L., A logical base for a transformational grammar of english, 1969.
- (4) MONTAGUE R., English as a Formal Language, Linguaggi nella società e nella tecnica, Milan, 1970.
- (5) PASERO R., Représentation du français en logique du 1er ordre, en vue de dialoguer avec un ordinateur, thèse de 3ème cycle, Groupe d'Intelligence Artificielle, UER de LUMINY, Université d'AIX-MARSEILLE, juin 1973.
- (6) ROBINSON J.A., A machine oriented logic based on the resolution principle, Journal of the Association for Computing Machinery, décembre 1965.
- (7) ROUSSEL Ph., Prolog, manuel d'utilisation, rapport interne, Groupe d'Intelligence Artificielle, UER de LUMINY, Université d'AIX-MARSEILLE, septembre 1975.