RAIRO. INFORMATIQUE THÉORIQUE

A. COLMERAUER

Un sous-ensemble intéressant du français

RAIRO. Informatique théorique, tome 13, nº 4 (1979), p. 309-336
<http://www.numdam.org/item?id=ITA_1979__13_4_309_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 13, n° 4, 1979, p. 309 à 336)

UN SOUS-ENSEMBLE INTÉRESSANT DU FRANÇAIS (*)

PAR A. COLMERAUER [1]

Communiqué par M. NIVAT

Résumé. — *Nous nous intéressons à définir un sous-ensemble minimal du français qui puisse servir, notamment, pour créer et consulter des banques de données. Après avoir examiné les énoncés les plus simples que l'on puisse faire autour d'un verbe, d'un nom commun et d'un adjectif, nous passons aux énoncés plus complexes faisant intervenir des articles, des relatives et des négations. L'accent est mis sur la transformation systématique d'une phrase en une formule sémantique. Il apparaît que ces formules sémantiques ne peuvent être interprétées correctement que si l'on se place dans une logique à trois valeurs de vérité et que l'on suppose que les relations élémentaires que l'on associe aux verbes, aux noms communs et aux adjectifs ne portent pas sur des individus mais sur des ensembles d'individus.*

Abstract. — *We are interested in defining a minimal natural language subset which can be used as an instrument, in particular, in order to create and consult data bases. After examining the simplest statements that can be based on a verb, a noun and an adjective, we go on to more complex statements involving articles, relative clauses and negations. Emphasis is laid on the systematic transformation of a sentence into a semantic formula. It appears that such semantic formulae can be interpreted correctly only in a logical system with three truth-values. Moreover one must suppose that the elementary relations associated with the verbs, nouns and adjectives range not on individuals, but on sets of individuals.*

INTRODUCTION

Notre but est d'essayer de délimiter et d'étudier un sous-ensemble de langue naturelle qui puisse servir de point de départ à de nombreuses applications telles que la consultation, la création et la mise à jour d'une banque de données évoluée. Ce sous-ensemble doit être tel que :

(1) sa syntaxe soit simple mais naturelle;

(2) sa sémantique puisse être rigoureusement définie.

(*) Reçu octobre 1977, révisé mai 1979.

[1] Groupe d'Intelligence artificielle, E.R.A. n° 363, Université d'Aix - Marseille-II, France.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1979/309/$ 4.00

Cependant, malgré notre souci de formaliser nous voulons éviter à tout prix de plaquer un modèle *a priori;* ce modèle se dégagera de lui-même petit à petit.

Nous introduisons tout d'abord la notion d'énoncés élémentaires autour de noms propres. Nous nous gardons de leur attacher une sémantique trop précise. Nous montrons ensuite le rôle fondamental que jouent les articles : lier deux énoncés et introduire une variable qui se comporte comme un nom propre. Ceci nous amène aux quantificateurs à trois branches qui permettent de bien poser les problèmes de hiérarchie dans la quantification.

Après quelques considérations sur la négation, la conjonction entre deux énoncés et les propositions relatives nous précisons les restrictions auxquelles doit satisfaire un énoncé pour se traduire facilement en logique du premier ordre classique. Ces restrictions étant trop fortes, nous abandonnons alors cette logique et nous réexaminons le problème sous un angle nouveau. Nous sommes amenés à faire deux hypothèses plus précises sur la sémantique de notre sous-ensemble de langue naturelle :

(1) Certaines phrases dans une situation donnée n'ont pas de sens et il faut donc considérer la valeur de vérité « indéfini » en plus des valeurs de vérité « vrai » et « faux ». (Cette technique a été utilisée par E. Keenan [5] pour formaliser la notion de présupposition.)

(2) Les propriétés qu'introduisent les phrases doivent porter sur des ensembles d'individus et non pas sur des individus : ceci surtout pour traiter correctement les pluriels. (R. Pasero [10] avait déjà fait cette hypothèse.)

Tout ceci nous conduit à un certain type de système logique sous-jacent aux langues naturelles. La sémantique d'une formule de ce système n'est rien d'autre que les variations de ses valeurs de vérité (vrai, faux et indéfini) lorsque l'on se place dans différentes situations. Nous concluons en montrant comment associer une formule de ce système à chaque phrase du sous-ensemble du français.

ÉNONCÉS ÉLÉMENTAIRES

Intéressons-nous tout d'abord aux énoncés les plus simples que l'on puisse faire autour de noms propres. Nous en distinguerons trois sortes :

– ceux qui sont construits avec un nom commun et le verbe être :

Garigou *est (un) chat*
Pierre *est (le) fils de* Paul

– ceux qui sont construits avec un verbe :

Garigou *trotte*
Sophie *prête* Garigou *à* Anne

– ceux qui sont construits avec un adjectif et le verbe être :

Garigou *est tigré*
Anne *est contente de* Garigou

A ces six exemples nous ferons correspondre les formules suivantes :

estchat(garigou)
estfilsde(pierre,paul)
trotte(garigou)
prêteà(sophie,garigou,anne)
esttigré(garigou)
estcontentde(anne,garigou)

HYPOTHÈSE 1 : A chaque verbe, à chaque adjectif et à chaque nom commun nous associerons une propriété à *n* arguments portant sur les noms propres.

Il faut remarquer que non seulement les verbes, mais aussi les adjectifs et les noms peuvent introduire des propriétés à plus d'un argument.

INTRODUCTION DES QUANTIFICATEURS A TROIS BRANCHES

Considérons la phrase :

Arthur possède une voiture

on pourrait lui associer une formule du genre :

possède(arthur,un(voiture))

mais alors à la phrase :

Otto possède une voiture

correspondrait

possède(otto,un(voiture))

et on risquerait de conclure qu'Arthur et Otto possèdent la même voiture, alors qu'il est fort probable qu'il s'agisse de voitures différentes.

D'autre part dans la phrase :

Arthur possède une voiture

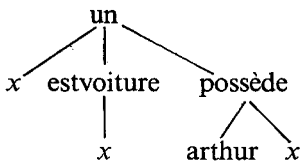
se cachent probablement deux énoncés élémentaires l'un autour du verbe « possède » et l'autre autour du nom « voiture ». Il serait intéressant de les faire apparaître clairement. Nous proposons de remplacer l'énoncé par la paraphrase pseudo-française

pour un x tel que x est (une) voiture,
il est vrai qu'Arthur possède x

à laquelle nous associons la formule :

$$\text{un}(x,\text{estvoiture}(x),\text{possède}(\text{arthur},x))$$

qui se visualise mieux par l'arbre :



HYPOTHÈSE 2 : D'une façon générale à chaque article α correspondra un « quantificateur à trois branches » q qui, à partir d'une variable x et de deux formules $f1$ et $f2$, crée la nouvelle formule :

$$q(x,f1,f2)$$

correspondant à l'énoncé :

pour α x tel que $e1$, il est vrai que $e2$

où $e1$ et $e2$ sont les énoncés correspondant à $f1$ et $f2$.

Par article nous entendons des mots comme

"un", "chaque", "le", "la", "les",...

mais aussi des groupes de mots comme

"beaucoup de", "peu de", "aucun(ne)", "tous les",...

Nous avons introduit le terme de « quantificateur à trois branches » pour rappeler que ce type de quantificateur lie une variable à deux formules, contrairement aux quantificateurs classiques $\exists$ et $\forall$ qui ne lient une variable qu'à une seule formule. Nous indiquerons plus loin les cas où l'on peut se ramener directement à ces quantificateurs classiques.

HIÉRARCHIE DANS LA QUANTIFICATION

Soit la phrase :

aucun homme (ne) possède une trompe

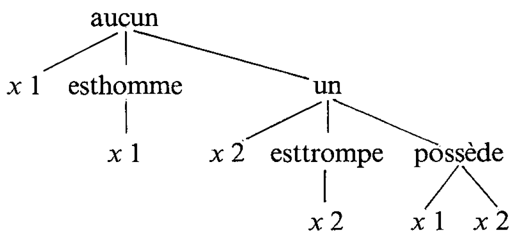
faut-il la paraphraser par :

(1) pour aucun $x\,1$ tel que $x\,1$ est (un) homme,
il est vrai que
pour un $x\,2$ tel que $x\,2$ est (une) trompe,
il est vrai que $x\,1$ possède $x\,2$

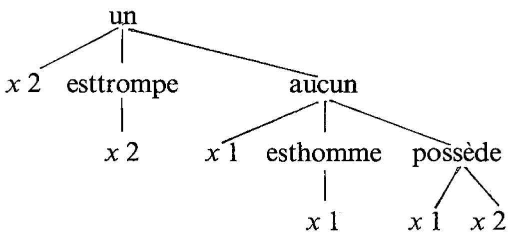
ou par :

(2) pour un $x\,2$ tel que $x\,2$ est (une) trompe,
il est vrai que
pour aucun $x\,1$ tel que $x\,1$ est (un) homme,
il est vrai que $x\,1$ possède $x\,2$

Le cas (1) se traduit par la formule :



et le cas (2) par :



Bien entendu c'est la paraphrase (1) qui est correcte, la paraphrase (2) signifiant quelque chose comme : il y a une trompe et aucun homme ne la possède.

En essayant plusieurs exemples avec toutes sortes d'articles on arrive à l'hypothèse suivante :

HYPOTHÈSE 3 *a* : La quantification introduite par l'article du sujet d'un verbe domine la ou les quantifications introduites par les compléments étroitement liés à ce verbe.

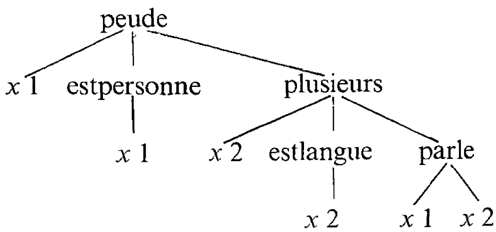
En parlant de compléments étroitement liés au verbe nous excluons notamment les compléments circonstanciels que nous n'étudierons pas ici.

Ce principe permet de mieux saisir l'altération de sens qui se produit lorsque l'on transforme une phrase au mode actif en une phrase au mode passif.

Alors que :

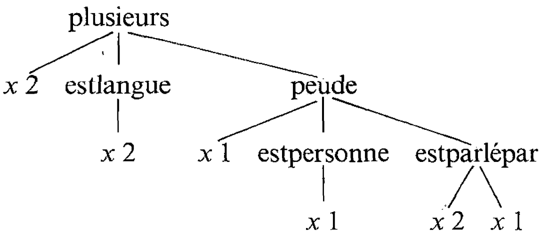
(1) peu de personnes parlent plusieurs langues

se traduit par :

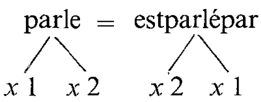


(2) plusieurs langues sont parlées par peu de personnes

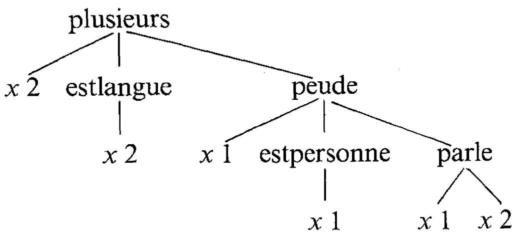
se traduit par :



et si l'on dispose d'une règle qui précise que :



l'exemple (2) peut se traduire finalement par :

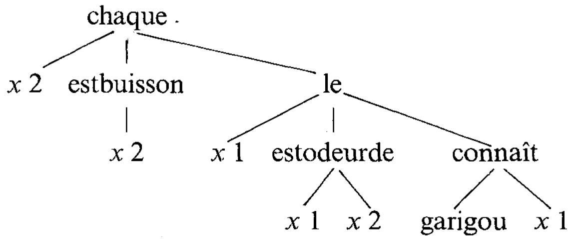


Le passage de la voix active à la voix passive inverse donc la hiérarchie de certaines quantifications.

Considérons maintenant les phrases :

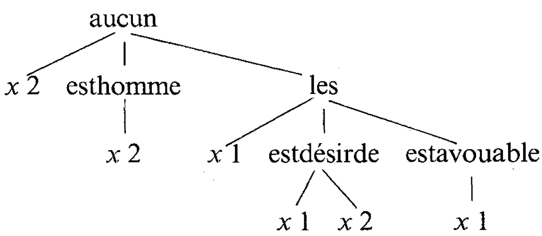
Garigou connaît l'odeur de chaque buisson

qui donne :



les désirs d'aucun homme (ne) sont avouables

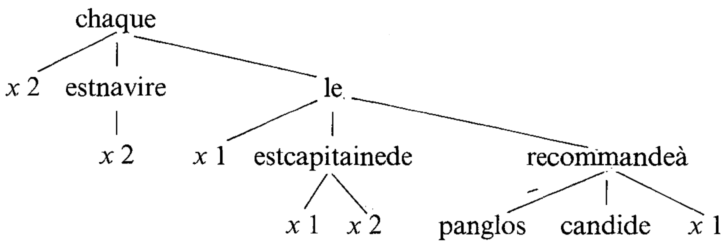
qui donne :



et finalement :

Panglos recommande Candide au capitaine de chaque navire

qui donne :



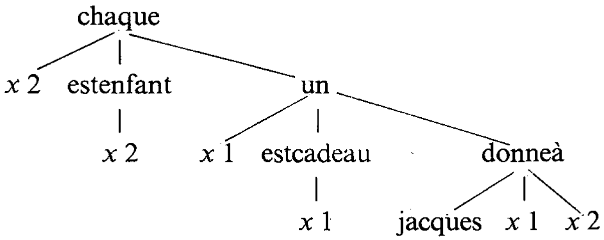
Ces exemples nous induisent à faire l'hypothèse :

HYPOTHÈSE 3 *b* : Dans une construction faisant intervenir un nom commun et un complément de ce nom, la quantification introduite par l'article du complément domine celle qui est introduite par l'article de ce nom commun.

Il ne nous reste plus qu'à considérer la hiérarchie de la quantification entre plusieurs compléments. Considérons la phrase :

Jacques donne un cadeau à chaque enfant

qui se traduit par :



Dans cet exemple le complément le plus à droite engendre une quantification qui domine celle du complément qui le précède. Nous généraliserons ce cas en adoptant l'hypothèse qui suit. Nous avons cependant bien conscience de faire une hypothèse un peu trop simplificatrice.

HYPOTHÈSE 3 *c* : Lorsqu'un verbe, un adjectif ou un nom, a deux compléments, la quantification se fait dans l'ordre inverse de l'ordre d'apparition naturel de ceux-ci; c'est-à-dire le complément le plus à droite engendre une quantification qui domine la quantification engendrée par le complément le plus à gauche.

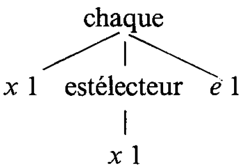
Pour illustrer les hypothèses 3 *a*, 3 *b* et 3 *c* traitons la phrase :

le vote de chaque électeur procure des frissons
à beaucoup de membres du gouvernement

L'ordre des quantifications correspondant à chaque article sera donc :

(2) le vote de (1) chaque électeur procure (5) des frissons à (4) beaucoup de membres de (3) le gouvernement

En procédant par étapes successives nous obtenons :

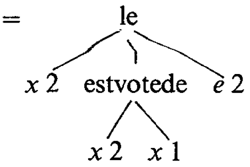


avec :

$e\,1$ = le vote de $x\,1$ procure des frissons
à beaucoup de membres de le gouvernement

c'est-à-dire :

$e\,1 =$

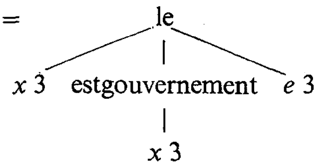


avec :

$e\,2 = x\,2$ procure des frissons à beaucoup de membres de le gouvernement

c'est-à-dire :

$e\,2 =$

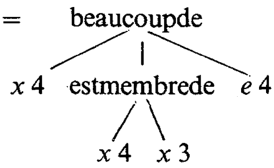


avec :

$e\,3 = x\,2$ procure des frissons à beaucoup de membres de $x\,3$

c'est-à-dire :

$e\,3 =$

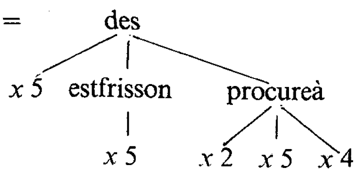


avec :

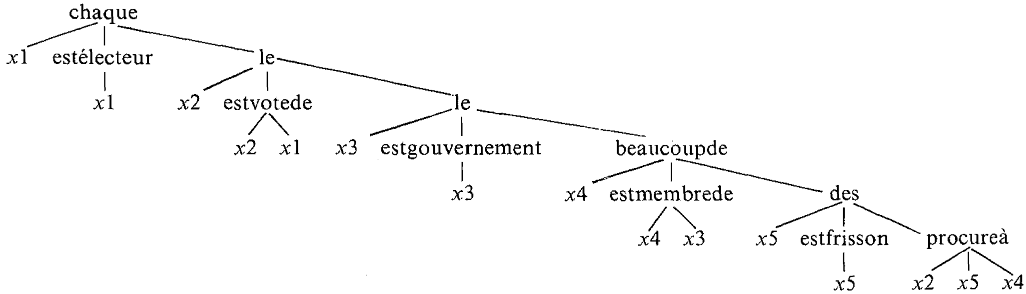
$e\,4 = x\,2$ procure des frissons à $x\,4$

c'est-à-dire :

$e\,4 =$



Finalement en recollant tous les morceaux :



NÉGATION

Étant donné un énoncé e, considérons le nouvel énoncé :

il est faux que e

Nous le traduirons par :

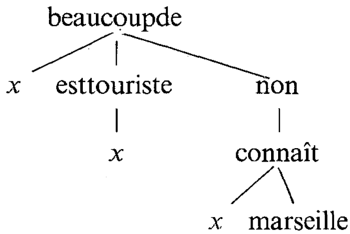
non(f)

où f est la traduction de e.

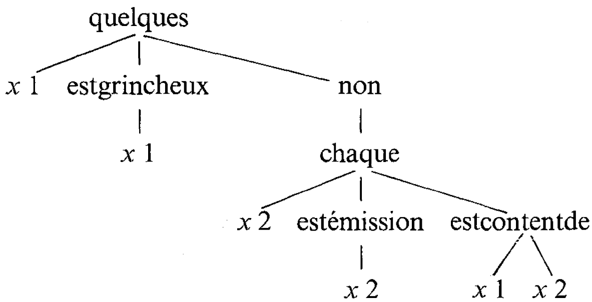
Étudions les liens entre la négation « ne pas » et l'opérateur « il est faux que ».

Voici quelques phrases négatives avec leur traduction :

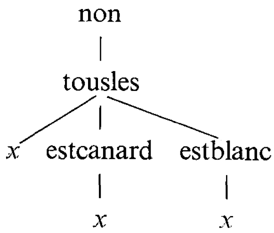
beaucoup de touristes ne connaissent pas Marseille



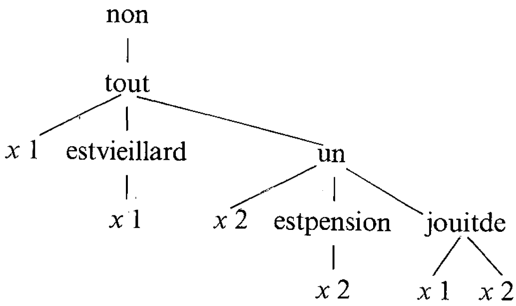
quelques grincheux ne sont pas contents de chaque émission



tous les canards ne sont pas blancs



tout vieillard ne jouit pas d'une pension



Nous formulerons l'hypothèse suivante :

HYPOTHÈSE 4 : Dans un énoncé la négation introduite par « ne pas » se traduit par l'opérateur « non » situé juste en dessous de la (ou les) quantification(s) introduite(s) par le sujet. Cependant, si l'article du sujet est « chaque », « chacun des », « chacunes des », « tout », « toute », « tous les », « toutes les », l'opérateur « non » porte sur tout l'énoncé.

CONJONCTION ENTRE ÉNONCÉS, INTRODUCTION DES RELATIVES

Considérons deux énoncés $e\,1$ et $e\,2$ et le nouvel énoncé :

$$e\,1 \quad \text{et} \quad e\,2.$$

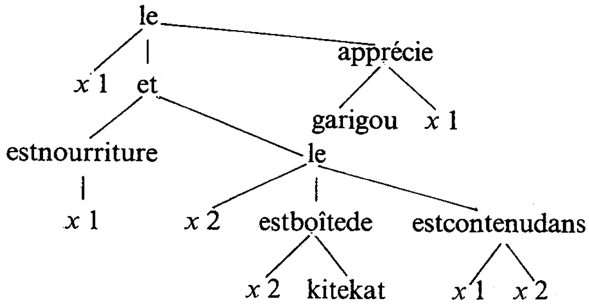
Nous le traduirons par :

$$\text{et}(f\,1, f\,2)$$

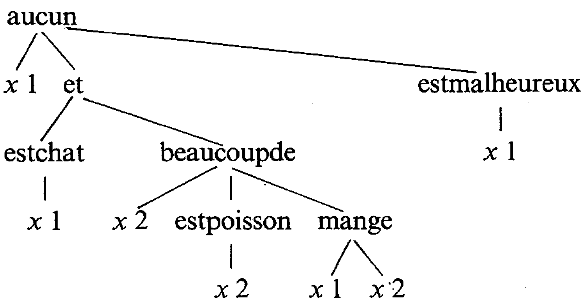
où $f\,1$ et $f\,2$ sont les traductions respectives de $e\,1$ et $e\,2$.

Nous pouvons maintenant traiter des phrases contenant des propositions relatives (nous ne nous intéressons qu'aux relatives dites restrictives) :

Garigou apprécie la nourriture
qui est contenue dans la boîte de Kitekat



aucun chat qui mange beaucoup de poissons (n') est malheureux.



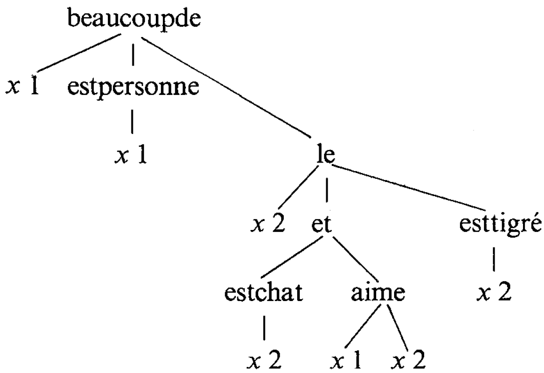
Nous formulerons l'hypothèse :

Hypothèse 5 : Toute proposition relative se traite comme un énoncé ordinaire; le pronom relatif est remplacé par la variable adéquate et le tout est lié à la traduction du nom commun par la conjonction « et ».

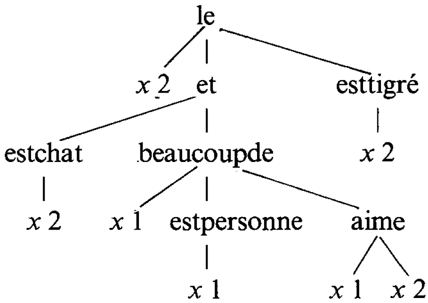
Il faut remarquer que l'hypothèse 5 exclut qu'une quantification apparaissant dans une relative passe au-dessus du « et » qui la lie au nom commun.

Il est donc incorrect de traduire :

le chat que beaucoup de personnes aiment est tigré par :



il faut le traduire par :



UTILISATION DIRECTE DE LA LOGIQUE DU PREMIER ORDRE

Limitons nous aux articles suivants et à leurs quantificateurs à trois branches associés :

un, une	→	un(x, e 1, e 2)
aucun(ne), aucune(ne)	→	aucun(x, e 1, e 2)
tout, toute, chaque	→	chaque(x, e 1, e 2)

Considérons la logique du premier ordre classique avec ses connecteurs $\neg$, $\supset$, $\wedge$, ses quantificateurs $\exists$, $\forall$, ses symboles relationnels, ses constantes et ses variables.

– Assimilons les noms propres à des constantes.

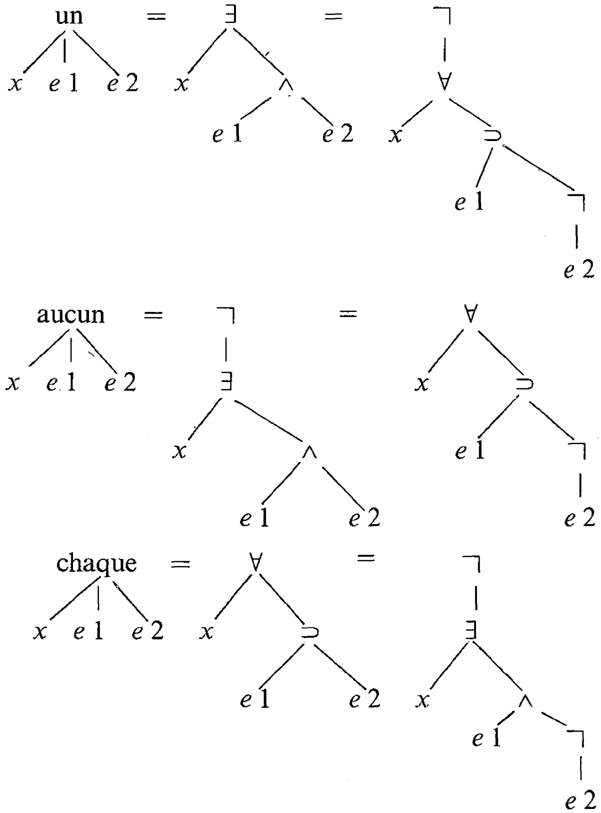
– Faisons correspondre à chaque verbe, chaque nom et chaque adjectif un symbole relationnel.

– Assimilons « et » à $\wedge$.

– Assimilons « non » à $\neg$.

– Assimilons les variables des quantificateurs à trois branches à celles de la logique du premier ordre.

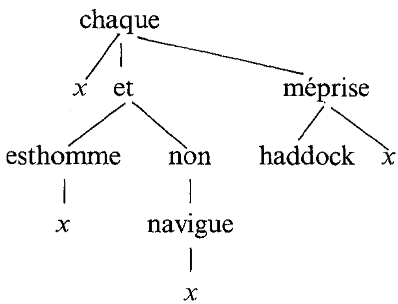
Les trois quantificateurs à trois branches précédents peuvent alors se définir par :



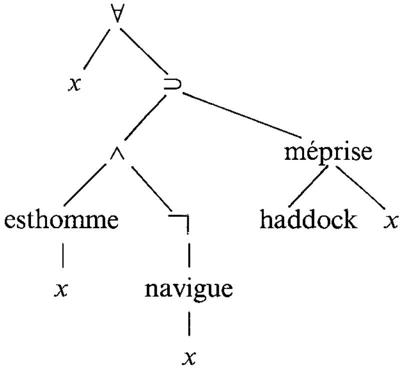
Nous pouvons donc maintenant traduire facilement certains énoncés en des formules de logique du premier ordre. Par exemple :

Haddock méprise tout homme qui ne navigue pas

se traduit tout d'abord par :



qui se transforme en :



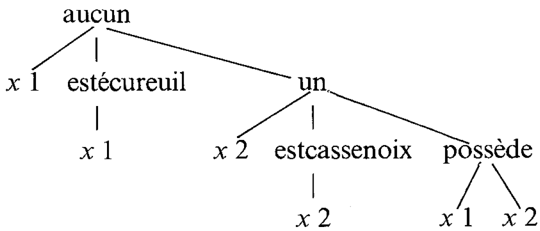
c'est-à-dire :

$$\forall x(\text{esthomme}(x) \;\wedge\; \neg\text{navigue}(x) \;\supset\; \text{méprise}(\text{Haddock}, x))$$

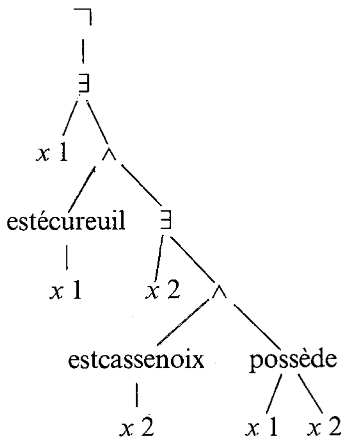
De même

aucun écureuil ne possède un casse-noix

se traduit tout d'abord par :



qui se transforme en :



c'est-à-dire :

$$\neg \exists x\,1(\text{estécureuil}(x\,1) \;\wedge\; \exists x\,2(\text{estcassenoix}(x\,2) \;\wedge\; \text{possède}(x\,1,x\,2)))$$

qui est la même chose que :

$$\forall x\,1\;\forall x\,2(\text{estécureuil}(x\,1) \;\wedge\; \text{estcassenoix}(x\,2) \;\supset \neg\; \text{possède}(x\,1,x\,2))$$

Les articles introduisant le pluriel et l'article défini ne se laissent pas traiter aussi simplement et nous allons être amenés à étudier une logique différente de la logique classique.

NÉCESSITÉ DE RAISONNER SUR PLUS DE DEUX VALEURS DE VÉRITÉ

Considérons l'énoncé :

le chat que Sophie retient miaule

Si l'on se place dans une situation où il y a un chat que Sophie retient, suivant qu'il miaule ou qu'il ne miaule pas, cet énoncé sera vrai ou sera faux. Par contre, si nous nous plaçons dans une situation où il n'y a pas de chat que Sophie retient, l'énoncé ne sera ni vrai, ni faux car :

il est faux que le chat que Sophie retient miaule

n'est pas vrai. L'énoncé dans ce cas n'a pas de sens, nous lui attribuerons la valeur de vérité « indéfini ». Ce cas ne se présente pas seulement avec l'article défini. Soit l'énoncé

Jacques rencontre Jacques

c'est-à-dire

Jacques se rencontre

Il est difficile d'imaginer une situation où cet énoncé est vrai, ou est faux. Il semble que d'une façon générale la propriété binaire que l'on associera au verbe « rencontrer » aura la valeur de vérité « indéfini » chaque fois que ses arguments seront identiques.

HYPOTHÈSE 6 : Dans une situation donnée un énoncé aura trois valeurs de vérité possibles : « vrai », « faux » ou « indéfini ».

A la lumière de ceci nous pouvons dès maintenant donner les tables de vérité des opérations « non » et « et » :

non(vrai) = faux
non(faux) = vrai
non(indéfini) = indéfini

et(vrai,vrai) = vrai
et(vrai,faux) = et(faux,vrai) = faux

et(faux,faux) = faux
et(vrai,indéfini) = et(indéfini,vrai) = indéfini
et(faux,indéfini) = et(indéfini,faux) = indéfini
et(indéfini,indéfini) = indéfini

Par la suite nous aurons besoin d'un nouvel opérateur binaire noté « si » :

si $f1$ = vrai alors si($f1$,$f2$) = $f2$
si $f1 \neq$ vrai alors si($f1$,$f2$) = indéfini

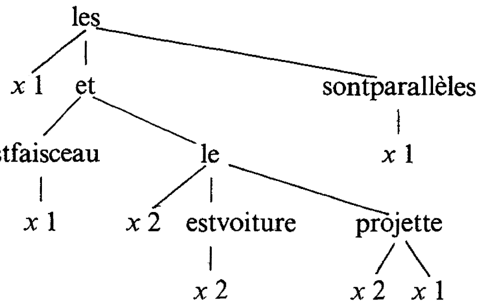
et aussi d'un autre opérateur de négation « nonvrai »

nonvrai(vrai) = faux
nonvrai(faux) = vrai
nonvrai(indéfini) = vrai

LES PROPRIÉTÉS DOIVENT PORTER SUR DES ENSEMBLES

Considérons la phrase :

les faisceaux que projette la voiture sont parallèles



La propriété unaire « sontparallèles » n'a de sens que si son argument x 1 est un ensemble. Bien entendu on peut dans une phase ultérieure se ramener à une propriété binaire sur des individus (« estparallèleavec ») en convenant par exemple que :

sontparallèles(X) = vrai ssi $X \neq \emptyset$ et $\forall x,y \in X$ estparallèleavec(x,y) = vrai
sontparallèles(X) = faux ssi $X \neq \emptyset$ et $\forall x,y \in X$ estparallèleavec(x,y) = faux
sontparallèles(X) = indéfini dans les autres cas

Mais toutes les propriétés sur des ensembles ne sont pas aussi facilement réductibles à des propriétés sur des individus; surtout celles faisant intervenir des notions de cardinalité. Par contre, une propriété sur des individus peut toujours se ramener à une propriété sur des ensembles en représentant un individu par un ensemble d'individus de cardinalité 1. Nous ferons donc l'hypothèse :

HYPOTHÈSE 7 : Les propriétés n-aires introduites par les verbes, les noms communs et les adjectifs ne portent pas sur des individus mais sur des ensembles d'individus.

Dans le présent article nous ne nous intéresserons pas aux différents moyens plus ou moins systématiques qui permettraient d'exprimer ces propriétés sur des ensembles à partir de propriétés sur des individus.

UN SYSTÈME LOGIQUE SOUS-JACENT AUX LANGUES NATURELLES

Nous allons maintenant introduire un système logique rigoureux dont nous pourrons définir clairement la sémantique. Bien entendu, ce système fera intervenir, entre autres, trois valeurs de vérité, des propriétés (ou relations) sur des ensembles et les opérateurs « et » et « non ». La quantification se fera par un mécanisme unique consistant à introduire la formule ces(x,e) signifiant grossièrement « ces x qui satisfont l'énoncé e ». Ceci nous permettra de définir le quantificateur à trois branches pour(x,e,e') signifiant « pour ces x qui satisfont l'énoncé e, l'énoncé e' est vrai ». Nous montrerons ensuite comment transformer les formules que nous associons aux phrases en des formules de ce système logique.

- Soit K un ensemble fini de symboles dits « noms propres ».
- Soit R un ensemble fini de symboles dits « relationnels ». A chaque symbole $r \in R$ est associé un entier positif n et l'on note ordre$[r]=n$.
- Soit X un ensemble de variables.

Nous définirons trois types de formules (chaque formule est bien entendu de longueur finie) e_i, s_i et n_i :

DÉFINITION : une *formule de type énoncé* e_i est de l'une des 8 formes

(1) $r(s_{i_1},s_{i_2},\ldots,s_{i_p})$ avec $r \in R$ et ordre$[r]=p$
(2) et(e_j,e_k)
(3) si(e_j,e_k)
(4) non(e_j)
(5) nonvrai(e_j)
(6) égal(n_j,n_k)
(7) inf(n_j,n_k)
(8) pour(x,e_j,e_k) avec $x \in X$

une *formule de type ensemble* s_i est de l'une des 5 formes

(1) $\emptyset$
(2) c avec $c \in K$
(3) $\{c_1, c_2, \ldots, c_n\}$ avec $c_i \in K$
(4) $\text{ces}(x, e_j)$ avec $x \in X$
(5) x avec $x \in X$

une *formule de type entier* n_i est de l'une des 2 formes

(1) j où j est un entier et $j \geqq 0$
(2) $\text{card}(s_j)$

L'occurence d'une variable x dans une formule f (de type énoncé, ensemble ou entier) est dite *libre* si elle n'a pas lieu à l'intérieur d'une sous-formule de la forme $\text{ces}(x,e)$ ou $\text{pour}(x,e,e')$. Une formule ne contenant aucune occurrence de variable libre est dite *fermée.*

Lorsque l'on se placera dans une situation bien définie une formule fermée de type énoncé aura une valeur de vérité (vrai, faux ou indéfini), une formule fermée de type ensemble aura pour valeur un ensemble et une formule fermée de type entier aura pour valeur un entier. Précisons cette notion de « situation » qui est une simplification de la notion d'« interprétation » en logique classique.

DÉFINITION : *Une situation g* est une application qui à chaque symbole relationnel $r \in R$ d'ordre n fait correspondre une relation n-aire $\rho = g[r]$ dont les arguments K_i sont les sous-ensembles de l'ensemble K des noms propres, et dont la valeur $\rho[K_1, K_2, \ldots, K_n]$ est soit « vrai », soit « faux », soit « indéfini » suivant les valeurs des K_i.

Nous pouvons maintenant clairement définir la valeur d'une formule fermée dans une situation.

DÉFINITION : Soit g une situation :

la valeur $\text{val}[e_i]$ *d'une formule fermée de type énoncé* e_j est définie par :

(1) si $e_i = r(s_{i_1}, s_{i_2}, \ldots, s_{i_p})$ alors
$\text{val}[e_i] = \rho[\text{val}[s_{i_1}], \text{val}[s_{i_2}], \ldots, \text{val}[s_{i_p}]]$ avec $\rho = g[r]$

(2) si $e_i = \text{et}(e_j, e_k)$ alors
$\text{val}[e_i] = \min[\text{val}(e_j), \text{val}(e_k)]$ avec vrai > faux > indéfini

(3) si $e_i = \text{si}(e_j, e_k)$ alors
si $\text{val}[e_j] = \text{vrai}$ alors $\text{val}[e_i] = \text{val}[e_k]$
si $\text{val}[e_j] \neq \text{vrai}$ alors $\text{val}[e_i] = \text{indéfini}$

(4) si $e_i = \text{non}(e_j)$ alors
si $\text{val}[e_j] = \text{vrai}$ alors $\text{val}[e_i] = \text{faux}$
si $\text{val}[e_j] = \text{faux}$ alors $\text{val}[e_i] = \text{vrai}$
si $\text{val}[e_j] = \text{indéfini}$ alors $\text{val}[e_i] = \text{indéfini}$

(5) si $e_i = \text{nonvrai}(e_j)$ alors
si $\text{val}[e_j] = \text{vrai}$ alors $\text{val}[e_i] = \text{faux}$
si $\text{val}[e_j] \neq \text{vrai}$ alors $\text{val}[e_i] = \text{vrai}$

(6) si $e_i = \text{égal}(n_j,n_k)$ alors
si $\text{val}[n_j] = \text{val}[n_k]$ alors $\text{val}[e_i] = \text{vrai}$
si $\text{val}[n_j] \neq \text{val}[n_k]$ alors $\text{val}[e_i] = \text{faux}$

(7) si $e_i = \text{inf}(n_j,n_k)$ alors
si $\text{val}[n_j] < \text{val}[n_k]$ alors $\text{val}[e_i] = \text{vrai}$
si $\text{val}[n_j] \geqslant \text{val}[n_k]$ alors $\text{val}[e_i] = \text{faux}$

(8) si $e_i = \text{pour}(x,e_j,e_k)$ alors
$\text{val}[e_j] = \text{val}[e'_k]$ où e'_k représente la formule e_k dans laquelle on a substitué à toute occurrence libre de x l'ensemble $\text{val}[\text{ces}(x,e_j)]$

la valeur val s_i d'une formule fermée de type ensemble s_i est définie par :

(1) si $s_i = \emptyset$ alors
$\text{val}[s_i] = \emptyset$

(2) si $s_i = c$ alors
$\text{val}[s_i] = \{ c \}$

(3) si $s_i = \{ c_1,c_2,\ldots,c_n \}$ alors
$\text{val}[s_i] = \{ c_1,c_2,\ldots,c_n \}$

(4) si $s_i = \text{ces}(x,e)$
$\text{val}[s_i] =$ l'union de tous les sous-ensembles E de K tels que
$\text{val}[e_{x\leftarrow E}] = \text{vrai}$
où $e_{x\leftarrow E}$ représente la formule e dans laquelle on a substitué E à toute occurrence libre de x.

la valeur val $[n_i]$ d'une formule fermée de type entier n_i est définie par :

(1) si $n_i = j$ où j est un entier non négatif alors
$\text{val}[n_i] = j$

(2) si $n_i = \text{card}(s_j)$ alors
$\text{val}[n_i] =$ nombre d'éléments de l'ensemble $\text{val}[s_j]$

La sémantique d'une formule fermée (de type énoncé, ensemble ou entier) f n'est rien d'autre que les variations de ses valeurs $\text{val}[f]$ lorsque l'on se place dans différentes situations.

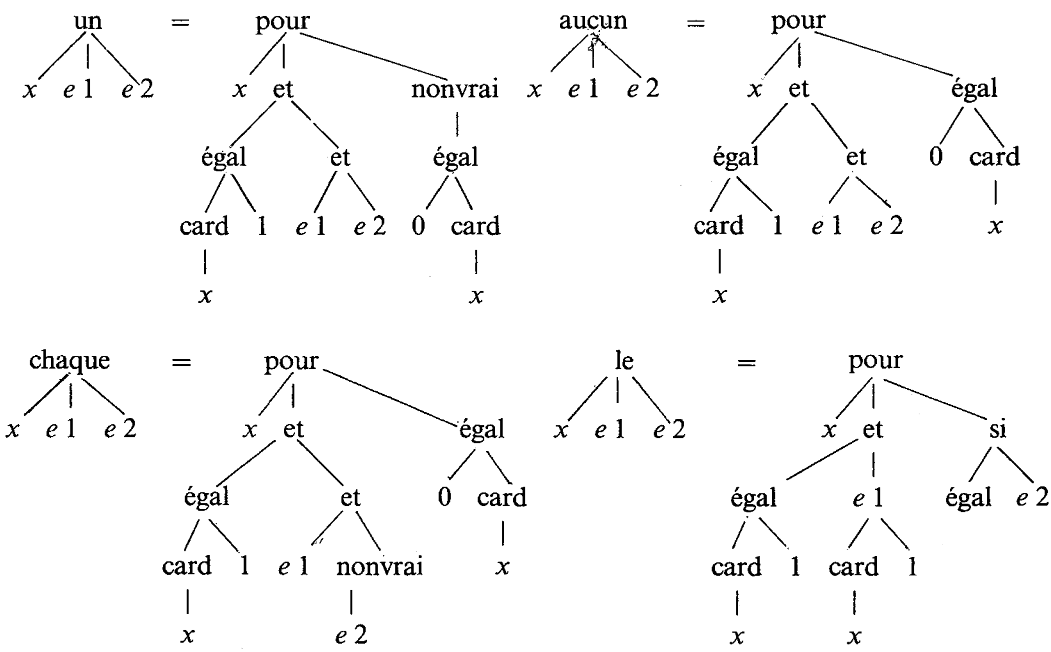
PASSAGE AU SYSTÈME LOGIQUE

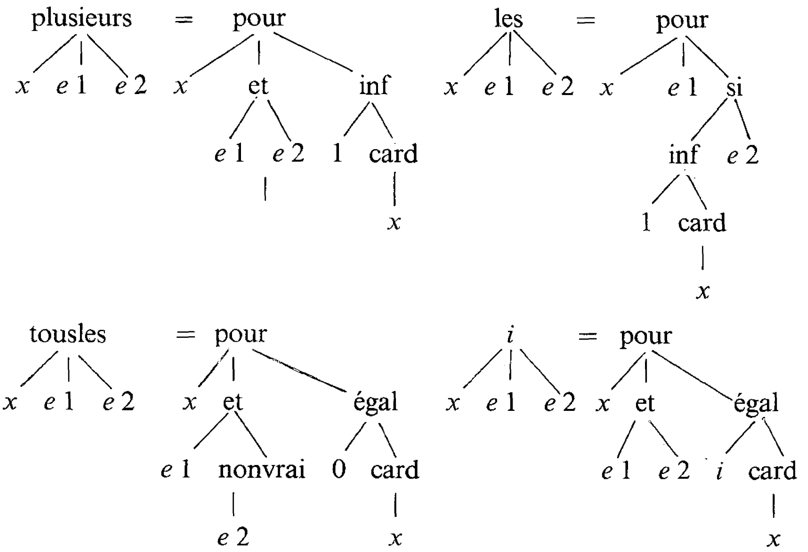
Les formules que nous avons introduites au début de cet article ne diffèrent des formules fermées de type énoncé que par les quantificateurs à trois branches. Nous allons montrer comment les éliminer.

Soit la liste des articles suivants ainsi que les quantificateurs à trois branches qu'ils introduisent :

un, une	→	un $(x,e\,1,e\,2)$
aucun(ne), aucune(ne)	→	aucun $(x,e\,1,e\,2)$
tout, toute, chaque	→	chaque $(x,e\,1,e\,2)$
l', le, la	→	le $(x,e\,1,e\,2)$
des, quelques, plusieurs	→	plusieurs $(x,e\,1,e\,2)$
les	→	les $(x,e\,1,e\,2)$
tous les, toutes les	→	tous les $(x,e\,1,e\,2)$
i : un entier non négatif comme dans « Pierre a 5 vaches »	→	i $(x,e\,1,e\,2)$

HYPOTHÈSE 8 : Nous choisirons les égalités suivantes :

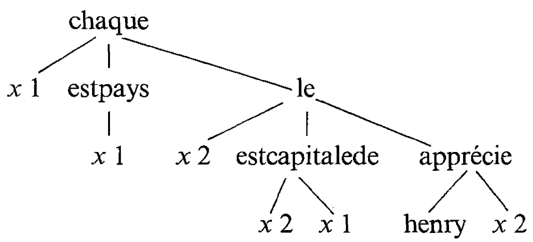




COMMENT RÉPONDRE A DES QUESTIONS

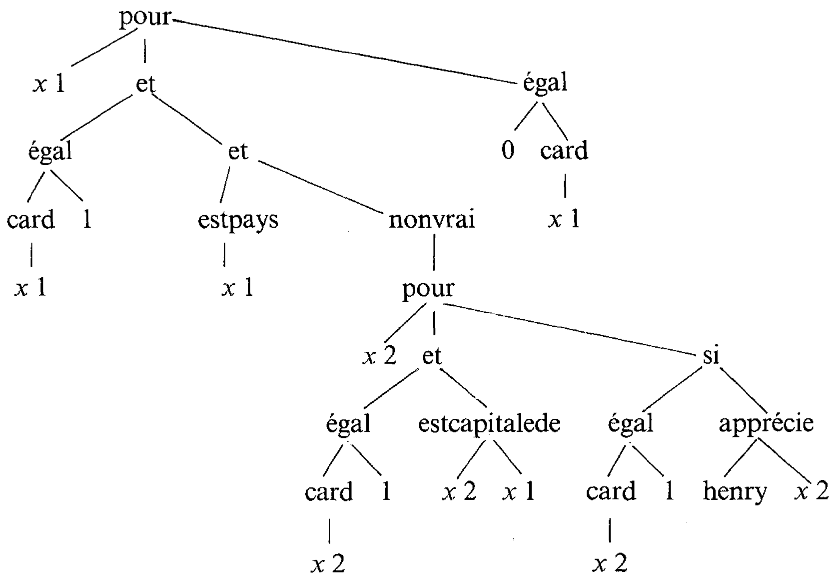
Considérons la phrase :

Henry apprécie la capitale de chaque pays



En utilisant l'hypothèse 8 nous obtenons la formule f (*voir* page suivante), c'est-à-dire une formule de notre système logique sous-jacent au français. Imaginons une banque de données où sont enregistrés un certain nombre de villes, de pays, de noms de personnes et de faits concernant ceux-ci. Cette banque décrirait une situation dans laquelle la valeur val$[f]$ de la formule f serait soit « vrai », soit « faux », soit « indéfini ». Cette valeur ne serait rien d'autre que la réponse à la question :

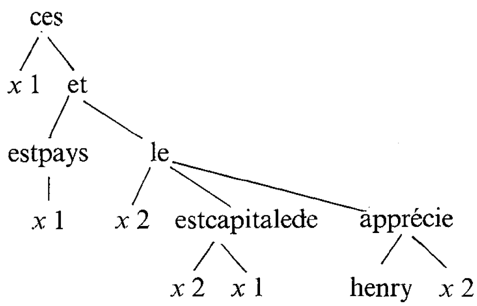
est-ce que Henry apprécie la capitale de chaque pays ?



De la même façon on pourrait calculer la réponse à la question :

de quels pays est-ce que Henry apprécie la capitale ?

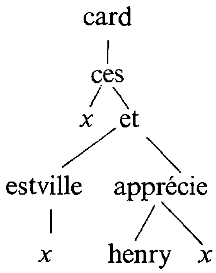
en partant de la formule :



et en la transformant en une formule de type ensemble. La réponse serait alors un ensemble de noms propres, chacun représentant un pays.

Pour répondre à la question :

combien de villes est-ce que Henry apprécie il suffirait de calculer la valeur de la formule de type entier :



Si l'on veut donc écrire un algorithme pour calculer la réponse à une question quelconque, dans une situation précise, il suffit de suivre exactement notre définition de val $[f]$.

CONCLUSION

Quelles sont les limites de la présente étude ? Tout d'abord l'hypothèse 1 qui associe notamment à chaque verbe une propriété n-aire ne portant, ni sur les temps, ni sur certains états, ignore tous les problèmes posés par les verbes d'action. En effet, de telles propriétés ne peuvent traduire que des états permanents alors que la sémantique des verbes d'action ne se définit qu'en termes de modifications d'états. Notre étude risque donc d'être de faible portée pour tout ce qui concerne l'utilisation de la langue naturelle en robotique. Signalons à ce sujet l'excellent article de J. McCarty et P. Hayes [7] où ce problème est traité par l'introduction d'une variable de « situation ».

Nous n'avons pas examiné le problème posé par la sémantique des complétives, c'est-à-dire d'arguments de verbes qui sont des propositions. Une grande partie de l'œuvre de R. Montague porte sur ce sujet [9].

Sans même aller aussi loin que les complétives, nous ne savons pas traiter le problème posé par le fait que si l'on considère les six phrases :

(1 *a*) Jacques connaît le frère de Paul;

(2 *a*) Henry est le frère de Paul;

(3 *a*) Jacques connaît Henry;

(1 *b*) Jacques connaît le numéro de téléphone de Paul;

(2 *b*) le numéro de téléphone de Paul est 42 70 41;

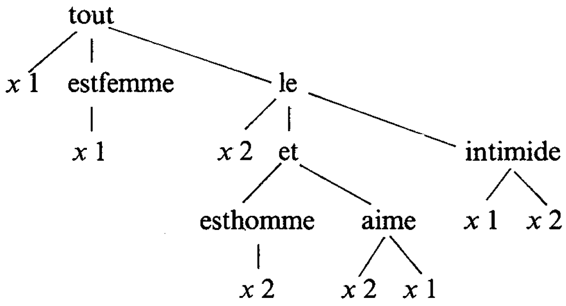
(3 *b*) Jacques connaît 42 70 41;

la phrase (3 *a*) se déduit de (1 *a*) et (2 *a*), mais la phrase (3 *b*) ne se déduit pas de (1 *b*) et (2 *b*). Ce type de problème est aussi longuement abordé par R. Montague [9] et J. McCarty [8].

D'autre part, nous n'avons pas abordé le problème des pronoms. Si nos variables permettent de représenter facilement certains cas comme par exemple :

Toute femme intimide l'homme qui l'aime

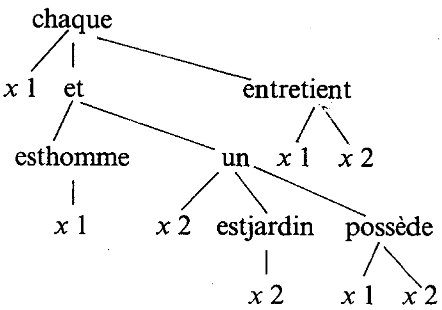
qui devient



la phrase

Chaque homme qui possède un jardin l'entretient

deviendrait



qui est une formule contenant une variable libre : l'occurrence de la variable $x\,2$ dans « entretient ($x\,1$, $x\,2$) » n'est pas quantifiée. Le même problème se présente pour représenter des pronoms de phrase à phrase. R. Pasero a longuement étudié de type de problème [10].

Nous n'avons pas abordé non plus l'ambiguïté de certains articles comme par exemple l'article « un » en position sujet. Dans

un éléphant arrive

l'article « un » a le sens que nous lui attribuons normalement, mais dans

un éléphant est gris

« un » signifie « chaque ». Un problème similaire se pose lorsqu'il faut reconstituer un article absent; ceci n'est pas trop fréquent en français mais peut l'être beaucoup plus dans d'autres langues.

Nous avons aussi laissé de côté toutes les incompatibilités entre les différents jeux d'articles et éventuellement la négation. Par exemple, ainsi qu'il a été remarqué en [2] un nom précédé de « aucun » ne peut être complément d'un nom précédé de « un » comme dans :

Victor n'a ramassé une feuille d'aucun arbre

D'autres contraintes sur la combinatoire des déterminants en français ont été répertoriées et analysées par M. Gross [4].

Parlons maintenant des applications. Notre travail a influencé profondément la conception de plusieurs systèmes de consultation de banque de données dans différentes langues :

– pour l'espagnol, le système de V. Dahl [3];

– pour le français, le système de J. F. Pique [11];

– pour le portugais, le système concernant la législation de la construction civile de H. Coelho et L. Pereira à Lisbonne (opérationnel mais non encore publié);

– pour l'anglais, le système de F. Pereira et D. Warren, à Edimbourg, qui porte sur les fichiers existant à un instant donné sous le système d'exploitation d'une machine (opérationnel mais non encore publié).

Tous ces systèmes sont programmés en logique du premier ordre [6] ou plus exactement en PROLOG [12] avec des grammaires de métamorphose [1] pour l'analyse.

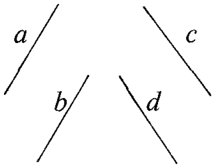
Disons quelques mots sur le premier système que nous connaissons bien : les réponses sont calculées en utilisant un algorithme qui est une transcription fidèle de notre définition de val $[f]$; cependant deux simplifications ont été faites :

d'une part pour passer systématiquement d'une relation n-aire R sur des ensembles à une relation n-aire r sur des individus il suppose que

$$\begin{aligned}
R(X_1,\ldots,X_n) &= \text{vrai ssi } X_1 \neq \emptyset,\ldots,X_n \neq \emptyset \text{ et} \\
&\qquad \forall x_1 \in X_1,\ldots,\forall x_n \in X_n \quad r(x_1,\ldots,x_n) = \text{vrai} \\
R(X_1,\ldots,X_n) &= \text{faux ssi } X_1 \neq \emptyset,\ldots,X_n \neq \emptyset \text{ et} \\
&\qquad \forall x_1 \in X_1,\ldots,\forall x_n \in X_n \quad r(x_1,\ldots,x_n) = \text{faux} \\
R(X_1,\ldots,X_n) &= \text{indéfini dans les autres cas}
\end{aligned}$$

et d'autre part, par souci d'efficacité pour calculer val [ces (x, e)] il ne prend que l'union des sous-ensembles x de cardinalité 1 tels que e soit vrai.

Si l'on considère quatre droites a, b, c, d parallèles deux à deux



et l'ensemble

$$E = \mathrm{val}[\mathrm{ces}(x, \mathrm{sontparallèles}(x))]$$

ce système trouvera que $E = \emptyset$ alors que si l'on prend comme définition de « sontparallèles » celle que nous avions proposée, on trouverait $E = \{a,b,c,d\}$. Il faut remarquer que « sontparallèles » est une propriété qui paradoxalement dans cette situation est telle que :

$$\mathrm{val}[\mathrm{sontparallèles}(\mathrm{ces}(x, \mathrm{sontparallèles}(x))] \neq \mathrm{vrai}$$

Nous concluerons en mentionnant que, du fait que tous ces systèmes ont été programmés en logique du premier ordre, il doit être possible de se passer de notre logique à trois valeurs. Celle-ci doit être considérée comme un outil commode pour clarifier le passage direct d'une forme syntaxique à une forme sémantique.

BIBLIOGRAPHIE

1. A. Colmerauer, *Metamorphosis Grammars,* in *Natural Language Communication with Computer,* Lectures Notes in Computer Science n° 63. Springer Verlag. 1978, p. 133-189.
2. C. Coursaget-Colmerauer, *Étude des structures du type Nom de Nom,* Thèse de doctorat, Université de Montréal, 1975.
3. V. Dahl, *Un système déductif d'interrogation de banques de données en espagnol,* Thèse de 3e cycle, Groupe d'Intelligence Artificielle, Université d'Aix-Marseille, novembre 1977.
4. M. Gross, *Grammaire transformationnelle du français, syntaxe du nom,* Collection « Langue et langage », Larousse, 1977.
5. E. L. Keenan, *On Semantically Based Grammar,* Linguistic Inquiry, vol. 111, n° 4, automne 1972.
6. R. Kowalski et M. van Emden, *The Semantics of Predicate Logic as a Programming Language,* J. Assoc. Comput. Mach., 1972, *23,* p. 733-743.
7. J. McCarthy et P. Hayes, *Some philosophical Problems from the Standpoint of Artificial Intelligence,* Machine Intelligence, vol. 4, 1969.
8. J. McCarthy, *First-Order Theories of Individual Concepts and Propositions,* Version préliminaire, Stanford Artificial Intelligence Laboratory, avril 1977.
9. R. Montague, *Formal Philosophy, Selected Papers,* R. H. Thomason, éd., Yale University Press, 1974.

10. R. PASERO, *Représentation du français en logique du 1er ordre en vue de dialoguer avec un ordinateur,* Thèse de 3e cycle, Groupe d'Intelligence Artificielle, Université d'Aix-Marseille, mai 1973.
11. J. F. PIQUE, *Interrogation en français d'une base de données relationnelles,* Diplôme d'Étude Approfondie, Groupe d'Intelligence Artificielle, Université d'Aix-Marseille, juillet 1978.
12. Ph. ROUSSEL, *Manuel d'utilisation PROLOG,* Rapport interne, Groupe d'Intelligence Artificielle, Université d'Aix-Marseille, septembre 1975.