Orbis

Alain Colmerauer

Mai 2015

1 Fonctionement

Il s'agit d'une banque de données qu'on interroge en anglais ou en français. Cette banque concerne le système planétaire, le soleil et toutes ses planètes. Voire section 2. Ce système est tiré d'un vieux système qui avait été fait par Richard Kittredge et moi-même en 1980. La banque de données est en Prolog :

Voici quelques exemples d'interrogation :

```
Who discovered Pluto?
Answer:
>> Tombaugh
Quel est le diametre de Leda ?
Reponse :
>> 140 km
```

L'interrogation porte toujours sur un argument, c'est-à-dire, un groupe nominal et se termine par un point d'interrogation. Deux exceptions : le point d'exclamation pour terminer

```
Stop!
>> Bye.
Stopper !
>> Au revoir.
```

et le simple point pour interrompre et passer à la phrase suivante :

```
Who est Io.
```

Dans le cas où la phrase ne peut pas être totalement analysé, le système donne le début correct le plus long possible suivi des mots qu'il s'attend à rencontrer.

```
|: What's the size of Earth?
[what,is,the,size,of,Earth,?]
[what,is,the]
Next possible word:
Earth           Moon            Sun             bodies          body
diameter        diameters       moon            moons           planet
planets         satellite       satellites
|: diameter of Earth?
[what,is,the,diameter,of,Earth,?]
Query: [mea,wh(#0,and(diameter(#0),the(sin,#1,diameterof(#1,Earth),equal(#0,#1))))]
Answer:
>> 12756 km
|: Autour de quo tourne Io ?
[autour,de,quo,tourne,Io,?]
[autour,de]
Mot possible suivant :
<entier naturel><nombre de km>  <un astronome>  au              aucun
aucune          cinq            combien         des             deux
diametre        diametres       exactement      l               la
le              les             lune            lunes           quatre
quel            quelle          quelles         quels           qui
```

```
quoi            satellite       satellites      trois           un
une
|: quoi tourne Io ?
[autour,de,quoi,tourne,Io,?]
Requete : [bod,wh(#0,and(body(#0),satelliteof(Io,#0)))]
Reponse :
>> Jupiter
```

2 Banque de donnée

Voici la banque de données en Prolog :

```
/* Universal, human and non human */

universal(X) :- human(X).
universal(X) :- nonhuman(X).

human(X) :- astronomer(X).

nonhuman(X) :- body(X).
nonhuman(X) :- diameter(X).

/* Astronomer diameter and body */

astronomer(A) :- discoveredby(S,A).

diameter(D) :- diameterof(D,S).
diameter(1).
diameter(10).
diameter(100).
diameter(1000).
diameter(10000).
diameter(100000).
diameter(1000000).
diameter(10000000).

body('the Sun').
body(S) :- planet(S).
body(S) :- moonof(S,Sp).

/* Sun, Earth and Moon */

thesun('the Sun').
theearth('Earth').
themoon('the Moon').

/* Planet and moon */

planet('Mercury').
planet('Venus').
planet(S) :- moonof(Sp,S).

moon(S) :- moonof(S,Sp).

/* Equal and less */

equal(D,D).
```

```
less(N,Np) :- N < Np.

/* Satellite and satellite of */

satellite(S) :- satelliteof(S,Sp).

satelliteof(S,'the Sun') :- planet(S).
satelliteof(S,Sp) :- moonof(S,Sp).

/* Diameter of */

diameterof(D,S) :- hasdiameter(S,D).

hasdiameter('Amalthea',270).
hasdiameter('Earth',12756).
hasdiameter('Jupiter',142800).
hasdiameter('Mars',6787).
hasdiameter('Mercury',4878).
hasdiameter('Neptune',49500).
hasdiameter('Pluto',3000).
hasdiameter('Saturn',120600).
hasdiameter('the Moon',3473).
hasdiameter('the Sun',1392000).
hasdiameter('Umbriel',700).
hasdiameter('Uranus',51800).
hasdiameter('Venus',12100).
hasdiameter(M,D) :- moonPlaAstroDiam(M,P,A,D).

/* Discovered by */

discoveredby('Pluto','Tombaugh').
discoveredby('Umbriel','Lassel').
discoveredby('Uranus','Herschel').
discoveredby('1980S1','Dollfus').
discoveredby(M,A) :- moonPlaAstro(M,P,A).
discoveredby(M,A) :- moonPlaAstroDiam(M,P,A,D).

/* Moon of */

moonof('Adrastea','Jupiter').
moonof('Amalthea','Jupiter').
moonof('Charon','Pluto').
moonof('Sinope','Jupiter').
moonof('the Moon','Earth').
moonof('1979J2','Jupiter').
moonof('1979J3','Jupiter').
moonof('1980S25','Saturn').
moonof('1980S26','Saturn').
moonof('1980S27','Saturn').
moonof('1980S28','Saturn').
moonof(S,Sp) :- moonPlaAstro(S,Sp,A).
moonof(S,Sp) :- moonPlaAstroDiam(S,Sp,A,D).

moonPlaAstro('1980S1','Saturn','Cassini').
moonPlaAstro('1980S3','Saturn','Fountain').
moonPlaAstro('1980S6','Saturn','Lacques').
moonPlaAstro('1980S13','Saturn','Smith').
```

```
moonPlaAstroDiam('Ananke','Jupiter','Nicholson',30).
moonPlaAstroDiam('Ariel','Uranus','Lassel',900).
moonPlaAstroDiam('Callisto','Jupiter','Galileo',4848).
moonPlaAstroDiam('Carme','Jupiter','Melotte',30).
moonPlaAstroDiam('Deimos','Mars','Hall',15).
moonPlaAstroDiam('Dione','Saturn','Cassini',1100).
moonPlaAstroDiam('Elara','Jupiter','Nicholson',30).
moonPlaAstroDiam('Enceladus','Saturn','Herschel',500).
moonPlaAstroDiam('Europa','Jupiter','Galileo',3126).
moonPlaAstroDiam('Ganymede','Jupiter','Galileo',5276).
moonPlaAstroDiam('Himalia','Jupiter','Nicholson',30).
moonPlaAstroDiam('Hyperion','Saturn','Bond',400).
moonPlaAstroDiam('Iapetus','Saturn','Cassini',1500).
moonPlaAstroDiam('Io','Jupiter','Galileo',3638).
moonPlaAstroDiam('Leda','Jupiter','Perrine',140).
moonPlaAstroDiam('Lysithea','Jupiter','Nicholson',30).
moonPlaAstroDiam('Mimas','Saturn','Herschel',400).
moonPlaAstroDiam('Miranda','Uranus','Kuiper',200).
moonPlaAstroDiam('Nereid','Neptune','Kuiper',300).
moonPlaAstroDiam('Oberon','Uranus','Herschel',1600).
moonPlaAstroDiam('Pasiphae','Jupiter','Nicholson',30).
moonPlaAstroDiam('Phobos','Mars','Hall',27).
moonPlaAstroDiam('Phoebe','Saturn','Pickering',20).
moonPlaAstroDiam('Rhea','Saturn','Cassini',1400).
moonPlaAstroDiam('Tethys','Saturn','Cassini',1200).
moonPlaAstroDiam('Titan','Saturn','Huygens',5150).
moonPlaAstroDiam('Titania','Uranus','Herschel',1700).
moonPlaAstroDiam('Triton','Neptune','Lassel',5000).
moonPlaAstroDiam('Umbriel','Uranus','Lassel',700).
```

Cette banque de données est interrogé au moyen de la forme logique suivante :

```
1 <main query> -> wh ( <variable> , <query> )

2 <query> -> true
3          | neg ( <query> )
4          | and ( <query> , <query> )
5          | exists ( <variable> , <query> )
6          | number ( <modifier> , <integer> , <variable> , <query> )
7          | the ( <sing plu> , <variable> , <query> , <query> )
8          | <predicat> | <predicat> ( <args> )

9 <modifier> -> exactly | atleast | atmost

10 <sing plu> -> sin | plu

11 <args> -> <arg> | <arg>, <args>

12 <arg> -> <variable>
13        | <atom>
14        | card ( <variable> , <query> )

15 <variable> -> # <integer>

16 <integer> -> <digit> | <digit> <integer>

17 <atom> -> <letter> | <letter> <atom>
```

```
18 <digit> -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

19 <letter> -> a | b | ... | z | A | B | ... | Z
```

3 Grammaire de l'anglais

Cette forme logique est produit par l'analyse de l'anglais dont la grammaire sous forme context-free est :

```
(1) Main clause
(2) Interrogative and relative clause
(3) Argument
(4) Noun phrase
(5) Right possesive adjunction
(6) Interogative and relative pronoun
(7) Determiner
(8) Clauses minus one argument
(9) Conjunction of relative clauses
(10) Lexical entries
(11) Lexical entries next

(1) Main clause
<english sentence> -> stop ! | <clause> ?

(2) Interrogative and relative clause
<clause> -> <arg> <verb compl> | <arg> <do> <suj verb prep>
<do> -> did | do | does

(3) Argument
<arg> -> <case> <np> | <whose> <noun 2>
<case> -> <empty> | <prep>
<prep> -> around | of | than | to

(4) Noun phrase
<np> -> <pron> | <np p>
<np p> -> <noun 0> <radj> | <number of km> km | <det> <noun 1> <radj> |
          <det> <noun 1> <rels> | <det> <noun 2> <arg>
<prep> -> around | of | than | to

(5) Right possesive adjunction
<radj> -> <empty> | ' s <radj>

(6) Interogative and relative pronoun
<pron> -> that | what | which | who | whom

(7) Determiner
<det> -> <empty> | <a> | <the> | <what>

(8) Clauses minus one argument
<verb compl> -> <verb 2> <arg> | <be> <arg> | <be> <adjv>
<suj verb prep> -> <arg> <verb 2> <prep dangle>
<adjv> -> <adjecvtive 2> <arg>
<prep dangle> -> <empty> | <prep>

(9) Conjunction of relative clauses
<rels> -> <clause> <and rels>
<and rels> -> <empty> | and <rels>

(10) Lexical entries
```

```
<noun 1> -> astronomer | astronomers | body | bodies | diameter |
            diameters | Earth | Moon | moon | moons | planet | planets |
            satellite | satellites | Sun
<noun 2> -> diameter | diameters | moon | moons | satellite | satellites
<verbe 2> -> discover | discovers | discovered | exceed | exceeds |
             revolve | revolves
<adjective 2> -> equal | greater | less

(11) Lexical entries next
<number of km> ->
   15 | 20 | 27 | 30| 140 | 200 | 270 | 300 | 400 | 500 | 700 |
   900 | 1100 | 1200 | 1400| 1500 | 1600 | 1700 | 3000 | 3126 |
   3473 | 3638 | 4848 | 5000 | 5150 | 5276 | 6787 | 12100 |
   12756 | 49500 | 51800 | 120600 | 142800| 1392000 |
   1 | 10 | 100 | 1000 | 10000 | 100000 | 1000000 | 10000000 |
   100000000
<noun 0> -> <astronomer> | <celestial body>
<astronomer> ->  <uk/fr astronommer> | Galileo
<star> ->
   <uk/fr star> | Adrastea | Amalthea | Enceladus | Europa |
   Iapetus | Lysithea | Mercury | Nereid | Pluto | Saturn |
   the Earth | the Moon | the Sun
<uk/fr astronomer> ->
   Bond | Cassini | Dollfus | Fountain |
   Hall | Herschel | Huygens | Kuiper | Lacques |
   Lassel | Melotte | Nicholson | Perrine | Pickering
   Smith | Tombaugh
<uk/fr star> ->
   1979J2 | 1979J3 | 1980S1 | 1980S13 | 1980S25 | 1980S26
   1980S27 | 980S28 | 1980S3 | 1980S6 |
   Ananke | Ariel | Callisto | Carme | Charon | Deimos | Dione
   Elara |  Ganymede | Himalia | Hyperion
   Io | Jupiter | Leda |  Mars
   Mimas | Miranda | Neptune | Oberon | Pasiphae
   Phobos | Phoebe | Rhea | Sinope | Tethys
   Titan | Titania | Triton | Umbriel | Uranus | Venus
```

4 Grammaire du français

La forme logique est aussi produit par l'analyse du français dont la grammaire sous forme context-free est :

```
(1) Proposition principale
(2) Proposition interrogative et relative
(3) Groupe nominal
(4) Groupe nominal incomplet
(5) Article
(6) Pronom interrogatif
(7) Pronom relatif
(8) Cas et preposition
(9) Propositions moins un argument
(10) Propositions moins deux arguments
(11) Conjonction de qualicatifs
(12) Entrees lexicales

(1) Proposition principale
<phrase> -> stoppe ! | <prop> ?

(2) Proposition interrogative et relative
```

```
<prop> -> <art> <etre> <gn> | <pron rel> <etre> <gn> |
          <gn> <estceq> <prop 1> | <gn> <gn moins> <verbe compl> |
          <gn> <suj verbe> <gn moins>
<estceq> -> <vide| est ce <quei>
<quei> -> que | qui

(3) Groupe nominal
<gn> -> <gn> | <pron rel> | <gn p>
<gn p -> <pron intero> | <artnombre> de <gn p| <cas> <nom 0> |
          <cas> <nom km 0> km | <art> <nom 1> <qual ev> |
          <art> <nom 2> <gn>

(4) Groupe nominal incomplet
<gn moins> -> <art> <nom 2> <gn moins suite>
<gn moins suite> -> <vide> | <gn moins>

(5) Article
<art> -> <art p> | <cas> <art sc>
<art sc> -> <vide| combien de | <art sc p> | <exact> <art sc num>
<exact> -> au moins | au plus | exactement
<art p> -> au | aux | du | de | des
<art sc p> -> aucun | aucune | des | l | la | le | les | quel |
              quelle | quelles | quels
<art sc num> -> cinq | deux | quatre | trois | un | une | 0 | 1 | 2

(6) Pronom interrogatif
<pron intero> -> <pronom intero> | <prep> <pronom intero sc>
<pronom intero> -> que | qui | quoi
<pronom intero sc> -> qui | quoi

(7) Pronom relatif
<pron rel> -> <pronom rel> | <prep<pronom rel sc>
<pronom rel> -> dont | que | qui
<pronom rel sc> -> qui | quoi

(8) Cas et preposition
<cas> -> <vide| <prep>
<prep> -> <prep ? >| autour <prep>
<prep ? > -> a | de

(9) Propositions moins un argument
<prop 1> -> <verbe compl> | <suj verbe> | <verbe suj>
<verbe compl> -> <prop 2> <gn>
<suj verbe> -> <gn> <prop 2>
<verbe suj> -> <prop 2> <gn>

(10) Proposition moins deux arguments
<prop 2> -> <ne> <aux verbe> <til> <pp adj>
<pp adj> -> <vide| <ppad>
<aux verbe> -> <avoir> | <etre> | <verbe 2>
<ppad> -> <ppavoir 2> | <ppavoir obj 2> | <adjectif 2>
<ne> -> <vide> | ne
<til> -> <vide> | <pronom pers>
<etre> -> est | sont
<avoir> -> a | ont
<pronom pers> -> elle | elles | il | ils

(11) Conjonction de qualicatifs
```

```
<quals ev> -> <quals> | <rels ev>
<quals> -> <adjf> <et quals>
<et quals> -> <vide> | et <quals> | et <rels>
<rels ev> -> <vide> | <rels>
<rels> -> <prop> <et rels>
<et rels> -> <vide> | et <rels>
<adjf> -> <adjectif 2> <gn>

(12) Entrees lexicales
<nom 0? > -> 0 | 1 | 2 | 3
<nom 0> -> <astronome> | <astre>
<astronome> -> Galilee  | <uk/fr astronomer>
<astre> ->
   Adrastee | Amalthee | Encelades | Europe | Japet | Lysithee |
   Mercure | Nereide | Pluton | Saturne | la lune | la terre|
   le soleil | <uk/fr star>
<nom 1> -> astre | astres | astronome | astronomes | lune | planete |
          planetes | soleil | terre
<nom 2> -> diametre | diametres | satellite | satellites
<verbe 2> -> decouvrit | geravite | gravitent | tourne | tournent
<pp avoir 2> -> decouvert
<pp avoir obj 2> -> | decouverte | decouvertes | decouverts
<adjectif 2> ->
   egal | egale | egales | egaux | inferieure | inferieures |
   inferieurs | superieur | superieure | superieures | superieurs

(11) Entree lexicale suite
<number of km> ->
   15 | 20 | 27 | 30| 140 | 200 | 270 | 300 | 400 | 500 | 700 |
   900 | 1100 | 1200 | 1400| 1500 | 1600 | 1700 | 3000 | 3126 |
   3473 | 3638 | 4848 | 5000 | 5150 | 5276 | 6787 | 12100 |
   12756 | 49500 | 51800 | 120600 | 142800| 1392000 |
   1 | 10 | 100 | 1000 | 10000 | 100000 | 1000000 | 10000000 |
   100000000
<nom 0> -> <astronome> | <astre>
<astronome> ->  <uk/fr astronomme> | Galilee
<astre> ->
   <uk/fr star> | Adrastee | Amalthee | Encelade | Europe |
   Japet | Lysithee | Mercure | Nereide | Pluton | Saturne |
   la terre | la lune | le soleil
<uk/fr astronomer> ->
   Bond | Cassini | Dollfus | Fountain |
   Hall | Herschel | Huygens | Kuiper | Lacques |
   Lassel | Melotte | Nicholson | Perrine | Pickering
   Smith | Tombaugh
<uk/fr star> ->
   1979J2 | 1979J3 | 1980S1 | 1980S13 | 1980S25 | 1980S26
   1980S27 | 980S28 | 1980S3 | 1980S6 |
   Ananke | Ariel | Callisto | Carme | Charon | Deimos | Dione
   Elara |  Ganymede | Himalia | Hyperion
   Io | Jupiter | Leda |  Mars
   Mimas | Miranda | Neptune | Oberon | Pasiphae
   Phobos | Phoebe | Rhea | Sinope | Tethys
   Titan | Titania | Triton | Umbriel | Uranus | Venus
```

Pour analyser le français nous utilisons une grammaire d'attributs calquée sur la grammaire context-free précédente. Par exemple à la règle context-free suivante :

```
<prep groupe nominal> -> <prep> <art> <nom>
```

correspond le programme Prolog suivant :

```
zPREPGRROUPENOMINAL(...,L0,L3) :- zPREP(...,L0,L1), zART(...,L1,L2), zNOM(...,L2,L3).
```

Les Li désignent des sous-suites de mots qu'on est en train d'analyser. Plus exactement

```
zSYMBOLE(...,Li,Lj)
```

analyse le début d'une suite Li et retourne la fin Lj de la suite. Les '...' désignent les termes nécessaires aux calculs de la forme logique.

5 Annexe : Orbis, le programme principal

```
/*
consult('~/Orbis/0.orbis.pl').
*/


/* The different part of the program */

:- op(400,fy,'#').
:- consult('~/Orbis/1.dialog').
:- consult('~/Orbis/2.data').
:- consult('~/Orbis/3a.englishGrammar').
:- consult('~/Orbis/3b.frenchGrammar').
:- consult('~/Orbis/4.input').

/* Start */

orbis :-
%  open('Users/alaincolmerauer/Orbis/5.empty.pl',read,X,[alias(sentences)]),
   open('Users/alaincolmerauer/Orbis/5.exemples.pl',read,X,[alias(sentences)]),
%  open('Users/alaincolmerauer/Orbis/5.examplesWithGrammars.pl',read,X,[alias(sentences)]),
   asserta(state(file)),
   step1([]).

:- orbis.
```

6 Annexe : Dialog, le superviseur

```
/* Next sentence and case analysis */

step1(L0) :-
   readsentence(L1), !,
   concsmall(L0,L1,L2),
   nl, write('1 '), write(L2), nl,
   end(L1,W),
   case(W,'.',step1([])),
   case(W,'!',step2(L2)),
   case(W,'?',step2(L2)).

end([W],W).
end([W1,W2|L],W) :- end([W2|L],W).

case(W,W,T) :- call(T).
case(W,Wp,T) :- dif(W,Wp).

concsmall([],[W|L],[Wp|L]) :-
   atom(W), atom_chars(W,[C|K]), capitaltosmall(C,Cp), !, atom_chars(Wp,[Cp|K]).
```

```
concsmall([],L,L) :- !.
concsmall(L1,L2,L3) :- conc(l1,L2,L3).

capitaltosmall(C,Cp) :- char_code(C,N), 64 < N, N < 91, Np is N+32, char_code(Cp,Np).



/* Parsing the sentence  */

step2(L) :-
   findall(P,englishsentence(P,L),Q1),
   findall(P,frenchsentence(P,L),Q2),
   writequeries(Q1),
   writequeries(Q2),
   case([Q1,Q2],[[],[]],step3(L)),
   case([Q1,Q2],[[P],[]],step4(english,P)),
   case([Q1,Q2],[[],[P]],step4(french,P)),
   case([Q1,Q2],[[],[P,Pp|Q]],step1([])),
   case([Q1,Q2],[[P,Pp|Q],[]],step1([])),
   case([Q1,Q2],[[P1|Qp1],[P2|Qp2]],step1([])).

writequeries([]).
writequeries([P|Q]) :- write('2 '), write(P), nl, writequeries(Q).

englishsentence(P,L) :- sentence(P,L), numberingmeta(0,P).
frenchsentence(P,L) :- phraz(P,L), numberingmeta(0,P).

numberingmeta(N,stop) :- !.
numberingmeta(N,[T,wh(#N,P)]) :- !, Np is N+1, numberingmeta(Np,P).
numberingmeta(N,true) :- !.
numberingmeta(N,neg(P)) :- !, numberingmeta(N,P).
numberingmeta(N,and(P,Pp)) :- !, numberingmeta(N,P), numberingmeta(N,Pp).
numberingmeta(N,exists(#N,P)) :- !, Np is N+1, numberingmeta(Np,P).
numberingmeta(N,number(E,Npp,#N,P)) :- !, Np is N+1, numberingmeta(Np,P).
numberingmeta(N,the(Npp,#N,P,Pp)) :-
   !, Np is N+1, numberingmeta(Np,P), numberingmeta(Np,Pp).
numberingmeta(N,P) :- P =.. [F|K], numberingargs(N,K).

numberingargs(N,[]).
numberingargs(N,[X|K]) :- numberingarg(N,X), numberingargs(N,K).

numberingarg(N,#Np).
numberingarg(N,X) :- atomic(X).
numberingarg(N,card(#N,P)) :- Np is N+1, numberingmeta(Np,P).

/* Next possible word */

step3(L) :-
   minuslastword(L,Lp),
   conc(Lp,[W|Lpp],Lppp),
   thoseknown(W,englishsentence(P,Lppp),K1), !,
   thoseknown(W,frenchsentence(P,Lppp),K2), !,
   threecases([K1,K2],Lp).

minuslastword([W],[]).
minuslastword([W,Wp|L],[W|Lp]) :- minuslastword([Wp|L],Lp).

threecases([[],[]],L) :- L = [W|Lp], step3(L).
```

```
threecases([[],[]],[]) :- write('>> ERREUR'), nl.
threecases([K1,K2],L) :-
   dif([K1,K2],[[],[]]),
   writebegin(L),
   englishpossibility(K1),
   frenchpossibility(K2),
   state(X),
   case(X,keyboard,step1(L)),
   case(X,file,step1([])).

writebegin(L) :-
   write(3), write(' ['), writebeginbis(L), write('X1,...Xn],'), nl.

writebeginbis([]).
writebeginbis([W|L]) :- write(W), write(','), writebeginbis(L).


minustheend([W],[]).
minustheend([W|L],[W|Lp]). :- minustheend(L,Lp).

englishpossibility([]) :- !.
englishpossibility(K) :-
   write('where X1, if it is an English word, is among:'), nl, writepossibilities(K).

frenchpossibility([]) :- !.
frenchpossibility(K) :-
   write('ou X1, si c''''est un mot francais, est parmis :'), nl, writepossibilities(K).

writepossibilities([]) :- nl.
writepossibilities([W1]) :-  po(W1), nl.
writepossibilities([W1,W2]) :-  po(W1), po(W2), nl.
writepossibilities([W1,W2,W3]) :-  po(W1), po(W2), po(W3), nl.
writepossibilities([W1,W2,W3,W4]) :-  po(W1), po(W2), po(W3), po(W4), nl.
writepossibilities([W1,W2,W3,W4,W5|L]) :-
   po(W1), po(W2), po(W3), po(W4), po(W5), nl, writepossibilities(L).

po(W) :- write(W), atom_length(W,N), Np is 17-N, space(Np).

space(0) :- !.
space(N) :- write(' '), Np is N-1, space(Np).

/* Consulting the data base */

step4(E,stop) :-
   case(E,english,write('>> Bye.')),
   case(E,french,write('>> Au revoir.')),
   nl.
step4(E,P) :-
   dif(P,stop),
   P = [T,Pp],
   findall(Y,queryanswer(Pp,Y),Z), !,
   sorted(Z,Zp),
   answer(E,T,Zp),
   step1([]).

answer(E,bod,[]) :-
   case(E,english,write('>> No eaven body.')),
   case(E,french,write('>> Aucun astre.')), nl.
```

```
answer(E,hum,[]) :-
   case(E,english,write('>> Nobody.')),
   case(E,french,write('>> Personne.')), nl.
answer(E,mea,[]) :-
   case(E,english,write('>> I do not know.')),
   case(E,french,write('>> Je ne sais pas.')), nl.
answer(E,T,Z) :- dif(Z,[]), nonemptyanswer(E,T,Z).

nonemptyanswer(E,T,[]).
nonemptyanswer(E,T,[Y|Z]) :-
   translate(E,T,Y,Yp),
   write('>> '), write(Yp), nl,
   nonemptyanswer(E,T,Z).

translate(E,mea,X,Y) :-
   !, number_chars(X,L), conc(L,[' ',k,m],Lp), atom_chars(Y,Lp).
translate(french,bod,X,Y) :-
   inlist([X,Y],[
   ['Adrastea','Adrastee'],['Amalthea','Amalthee'],
   ['Enceladus','Encelade'],['Europa','Europe'],
   ['Iapetus','Japet'],['Lysithea','Lysithee'],
   ['Mercury','Mercure'],['Nereid','Nereide'],
   ['Pluto','Pluton'],['Saturn','Saturne'],
   ['Earth','la terre'],['the Moon','la lune'],
   ['the Sun','le soleil']]), !.
translate(french,hum,'Galileo','Galilee') :- !.
translate(E,T,X,X).


%:- phrasefrancaise(Q,[qui,decouvrit,un,satellite,'?']), write(Q), nl, fail.
%:- phrasefrancaise(Q,[quel,astronome,a,decouvert,un,astre,'?']), write(Q), nl, fail.
%:- phrasefrancaise(Q,[quels,sont,les,satellites,qui,gravitent,autour,de,'Saturne','?']), write(Q), nl, fail.

/* The procedure thoseknown  execute P until X is known and collect the values of X in L. The list L is sorted.

do(X,P,Y) :- findall(X,P,L), inlist(Y,L).

thoseknown(X,P,L) :-
   asserta(item(zzzzzz)), freeze(X,assertfail(X)), call(P), fail.
thoseknown(X,P,Lp) :-  retract(item(A)), !, recover(A,[],L), sorted(L,Lp).

assertfail(X) :- asserta(item(X)), fail.

recover(zzzzzz,L,L) :- !.
recover(A,L,Lp) :- retract(item(Ap)), !, recover(Ap,[A|L],Lp).

/*
p(a).
p(b).
p(d) :- fail.
p(a).
p(c) :- fail.
p(d) :- fail.
p(d) :- fail.


bb :- thoseknown(X,p(X),L), write(L), nl, fail.
*/
```

```
/* SORT */

sorted(X,Y) :- size(X,N), sortbis(N,X,Y).

sortbis(N,Y,Y) :- N =< 1.
sortbis(N,X,Y) :-
   N >= 2,
   N1 is N//2,
   N2 is N-N1,
   concntimes(N1,X1,X2,X),
   sortbis(N1,X1,X1p),
   sortbis(N2,X2,X2p),
   fusion(X1p,X2p,Y).

fusion([],[],[]).
fusion(X,[],X) :- X=[A|Xp].
fusion([],X,X) :- X=[A|Xp].
fusion([A|X1],[A|X2],[A|X3]) :- fusion(X1,X2,X3).
fusion([A1|X1],[A2|X2],[A1|X3]) :- A1 @< A2, fusion(X1,[A2|X2],X3).
fusion([A1|X1],[A2|X2],[A2|X3]) :- A1 @> A2, fusion([A1|X1],X2,X3).

size([],0).
size([A|X],N) :- size(X,Np), N is Np+1.

concntimes(0,[],Z,Z).
concntimes(N,[A|X],Y,[A|Z]) :- Np is N-1, concntimes(Np,X,Y,Z).

/*
exemple1([a,c,d,e,o,y]).
exemple2([a,b,f,n]).

ok :- exemple1(X), exemple2(Y), fusion(X,Y,Z), write(Z), nl, fail.
ok :- exemple1(X), exemple2(Y), fusion(Y,X,Z), write(Z), nl, fail.
ok :- sorted([z,c,z,e,f,z,a,y,e,i,b,a,b,o],Y), write(Y), nl, fail.
*/

/* PETITS TRUCS */

neg(P) :- call(P), !, fail.     neg(P).

inlist(X,L) :- inlist(X,L,true).

outlist(X,L) :- inlist(X,L,false).

inlist(X,[],false).
inlist(X,[X|L],true).
inlist(X,[Xp|L],B) :- dif(X,Xp), inlist(X,L,B).

list(N,[]).
list(N,[W|L]) :- N > 0, M is N-1, list(M,L).

conc([],L,L).
conc([W|L],Lp,[W|Lpp]) :- conc(L,Lp,Lpp).


/* Mise au point */
```

```
d(N,P) :- write('['), write(N), write(' '), write(P), nl,
          call(P).

f(N,P) :- call(P),
          write(']'), write(N), write(' '), write(P), nl.

a(N,P) :- write('['), write(N), write(' '), write(P), nl,
          call(P),
          write(']'), write(N), write(' '), write(P), nl.
```

7 Annexe : Banque de données

```
/* CONSULTING THE DATA BASE */

queryanswer(wh(X,P),Y) :- ok(P,[is(X,Y)]).

ok(true,S) :- !.
ok(neg(P),S) :- !, neg(ok(P,S)).
ok(and(P,Pp),S) :- !, ok(P,S), ok(Pp,S).
ok(exists(X,P),S) :-  !, ok(P,[is(X,Y)|S]).
ok(number(E,N,X,P),S) :- !,
   findall(Y,ok(P,[is(X,Y)|S]),L), sizelist(L,E,N).
ok(the(sin,X,P,Pp),S) :- !,
   findall(Y,ok(P,[is(X,Y)|S]),[Ya]),
   findall(Yp,ok(and(P,Pp),[is(X,Yp)|S]),[Ya]).
ok(the(plu,X,P,Pp),S) :- !,
   findall(Y,ok(P,[is(X,Y)|S]),L),  L=[Ya,Yaa|La],
   findall(Yp,ok(and(P,Pp),[is(X,Yp)|S]),L).
ok(P,S) :-
   P =.. [F|K],
   arguments(K,S,L),
   Q =.. [F|L],
   call(Q).

arguments([],S,[]).
arguments([X|K],S,[Y|L]) :- argument(X,S,Y), arguments(K,S,L).

argument(#N,S,Y) :- inlist(is(#N,Y),S).
argument(X,S,X) :- atomic(X).
argument(card(X,P),S,Y) :- cardinality(Yp,ok(P,[is(X,Yp)|S]),Y).

cardinality(Y,P,N) :- findall(Y,P,L), sizel(L,N).

sizelist([],exactly,0).
sizelist(L,atleast,0).
sizelist([],atmost,N).
sizelist([Y|L],A,Np) :- Np>0, N is Np-1, sizelist(L,A,N).

sizel([],0).
sizel([Y|L],Np) :- sizel(L,N), Np is N+1.


/* THE DATA BASE */

/* Universal, human and non human */

universal(X) :- human(X).
```

```
universal(X) :- nonhuman(X).

human(X) :- astronomer(X).

nonhuman(X) :- body(X).
nonhuman(X) :- diameter(X).

/* Astronomer diameter and body */

astronomer(A) :- discoveredby(S,A).

diameter(D) :- diameterof(D,S).
diameter(1).
diameter(10).
diameter(100).
diameter(1000).
diameter(10000).
diameter(100000).
diameter(1000000).
diameter(10000000).

body('the Sun').
body(S) :- planet(S).
body(S) :- moonof(S,Sp).

/* Sun, Earth and Moon */

thesun('the Sun').
theearth('Earth').
themoon('the Moon').

/* Planet and moon */

planet('Mercury').
planet('Venus').
planet(S) :- moonof(Sp,S).

moon(S) :- moonof(S,Sp).

/* Equal and less */

equal(D,D).

less(N,Np) :- N < Np.

/* Satellite and satellite of */

satellite(S) :- satelliteof(S,Sp).

satelliteof(S,'the Sun') :- planet(S).
satelliteof(S,Sp) :- moonof(S,Sp).

/* Diameter of */

diameterof(D,S) :- hasdiameter(S,D).

hasdiameter('Amalthea',270).
hasdiameter('Earth',12756).
```

```
hasdiameter('Jupiter',142800).
hasdiameter('Mars',6787).
hasdiameter('Mercury',4878).
hasdiameter('Neptune',49500).
hasdiameter('Pluto',3000).
hasdiameter('Saturn',120600).
hasdiameter('the Moon',3473).
hasdiameter('the Sun',1392000).
hasdiameter('Umbriel',700).
hasdiameter('Uranus',51800).
hasdiameter('Venus',12100).
hasdiameter(M,D) :- moonPlaAstroDiam(M,P,A,D).

/* Discovered by */

discoveredby('Pluto','Tombaugh').
discoveredby('Umbriel','Lassel').
discoveredby('Uranus','Herschel').
discoveredby('1980S1','Dollfus').
discoveredby(M,A) :- moonPlaAstro(M,P,A).
discoveredby(M,A) :- moonPlaAstroDiam(M,P,A,D).

/* Moon of */

moonof('Adrastea','Jupiter').
moonof('Amalthea','Jupiter').
moonof('Charon','Pluto').
moonof('Sinope','Jupiter').
moonof('the Moon','Earth').
moonof('1979J2','Jupiter').
moonof('1979J3','Jupiter').
moonof('1980S25','Saturn').
moonof('1980S26','Saturn').
moonof('1980S27','Saturn').
moonof('1980S28','Saturn').
moonof(S,Sp) :- moonPlaAstro(S,Sp,A).
moonof(S,Sp) :- moonPlaAstroDiam(S,Sp,A,D).

moonPlaAstro('1980S1','Saturn','Cassini').
moonPlaAstro('1980S3','Saturn','Fountain').
moonPlaAstro('1980S6','Saturn','Lacques').
moonPlaAstro('1980S13','Saturn','Smith').

moonPlaAstroDiam('Ananke','Jupiter','Nicholson',30).
moonPlaAstroDiam('Ariel','Uranus','Lassel',900).
moonPlaAstroDiam('Callisto','Jupiter','Galileo',4848).
moonPlaAstroDiam('Carme','Jupiter','Melotte',30).
moonPlaAstroDiam('Deimos','Mars','Hall',15).
moonPlaAstroDiam('Dione','Saturn','Cassini',1100).
moonPlaAstroDiam('Elara','Jupiter','Nicholson',30).
moonPlaAstroDiam('Enceladus','Saturn','Herschel',500).
moonPlaAstroDiam('Europa','Jupiter','Galileo',3126).
moonPlaAstroDiam('Ganymede','Jupiter','Galileo',5276).
moonPlaAstroDiam('Himalia','Jupiter','Nicholson',30).
moonPlaAstroDiam('Hyperion','Saturn','Bond',400).
moonPlaAstroDiam('Iapetus','Saturn','Cassini',1500).
moonPlaAstroDiam('Io','Jupiter','Galileo',3638).
moonPlaAstroDiam('Leda','Jupiter','Perrine',140).
```

```
moonPlaAstroDiam('Lysithea','Jupiter','Nicholson',30).
moonPlaAstroDiam('Mimas','Saturn','Herschel',400).
moonPlaAstroDiam('Miranda','Uranus','Kuiper',200).
moonPlaAstroDiam('Nereid','Neptune','Kuiper',300).
moonPlaAstroDiam('Oberon','Uranus','Herschel',1600).
moonPlaAstroDiam('Pasiphae','Jupiter','Nicholson',30).
moonPlaAstroDiam('Phobos','Mars','Hall',27).
moonPlaAstroDiam('Phoebe','Saturn','Pickering',20).
moonPlaAstroDiam('Rhea','Saturn','Cassini',1400).
moonPlaAstroDiam('Tethys','Saturn','Cassini',1200).
moonPlaAstroDiam('Titan','Saturn','Huygens',5150).
moonPlaAstroDiam('Titania','Uranus','Herschel',1700).
moonPlaAstroDiam('Triton','Neptune','Lassel',5000).
moonPlaAstroDiam('Umbriel','Uranus','Lassel',700).


/* TESTS */
/*
queryanswer(wh(#0,and(planet(#0),the(#1,themoon(#1),satelliteof(#1,#0)))),Y).
queryanswer(wh(#0,planet(#0)),Y).
queryanswer(wh(#0, and(body(#0), discoveredby(#0,'Cassini'))),Y).

> qui decouvrit [] Mars ?
Requete :
:- queryanswer(wh(#0,and(astronomer(#0),discoveredby('Mars',#0))),Y).
*/
```

8 Annexe : grammaire de l'anglais

```
/* 1 Main clause */

sentence(stop,[stop,'!']).
sentence(P,L0) :- xCLAUSE(P,L0,L1), xWORD('?',L1,[]).

xWORD(W,[W|L],L).

/* 2 Interrogative and relative clause */

xCLAUSE(Pp,L0,L2) :-
   xARG([Y,P,Pp],L0,L1),
   xVERBxCOMPL([Y,P],L1,L2).
xCLAUSE(Pp,L0,L3) :-
   xARG([Y,P,Pp],L0,L1),
   xDO(Y,L1,L2),
   xSUJxVERBxPREP([Y,P],L2,L3).

xDO(Y,L0,L0) :- Y=[D,C,N,T,X], dif(D,intero).
xDO(Y,L0,L1) :- Y=[intero,C,N,T,X], does(W,[N]), xWORD(W,L0,L1).

does(do,[plu]).
does(does,[sin]).
does(did,[N]).


/* 3 Argument */

xARG([Y,P,Pp],L0,L2) :- xCASE(Y,L0,L1), xNP([Y,P,Pp],L1,L2).
xARG([Y,P,[T,wh(Xp,and(Ppp,exists(X,and(Pp,P))))]],L0,L2) :-
```

```
   Y=[D,C,N,T,X],
   Yp=[Dp,Cp,Np,Tp,Xp],
   inlist(C,[obj,suj]),
   xWORD(whose,L0,L1),
   xWORD(W,L1,L2),
   noun2(W,[N,T,s(Xp)],[Tp,Xp],Pp),
   domain(Tp,Xp,Ppp).
xARG([Y,P,rel(Yp,exists(X,and(Pp,P)))],L0,L2) :-
   Y=[D,C,N,T,X],
   Yp=[Dp,Cp,Np,Tp,Xp],
   inlist(C,[obj,suj]),
   xWORD(whose,L0,L1),
   xWORD(W,L1,L2),
   noun2(W,[N,T,s(Xp)],[Tp,Xp],Pp).

xCASE(Y,L0,L0) :- Y=[D,C,N,T,X], inlist(C,[obj,suj]).
xCASE(Y,L0,L1) :- Y=[D,C,N,T,X], prep(C), xWORD(C,L0,L1).

prep(around).  prep(of).  prep(than).  prep(to).

/* 4 Noun phrase */

xNP([Y,P,Pp],L0,L1) :-
   xPRON([Y,P,Pp],L0,L1).
xNP([Y,P,Pp],L0,L1) :-
   xNPxP([Y,P,Pp],L0,L1).

xNPxP([Y,P,Pp],L0,L2) :-
   Yp=[proper,Cp,sin,Tp,W],
   xWORD(W,L0,L1),
   noun0(W,[T]),
   xRADJ([Y,Yp,P,Pp],L1,L2).
xNPxP([Y,P,P],L0,L2) :-
   Y=[proper,C,N,mea,W],
   xWORD(W,L0,L1),
   noun0(W),
   xWORD(km,L1,L2).
xNPxP([Y,Pp,Pppp],L0,L2) :-
   Yp=[Dp,Cp,Np,Tp,Xp],
   Y=[D,C,N,T,X],
   xDET([Y,Ppp,Pp,Pppp],L0,L1),
   noun1(W,[N,T,X],Ppp),
   xWORD(W,L1,L2).
   %xRADJ([Yp,Y,P,Pp],L2,L3).
xNPxP([Y,Pppp,Ppppp],L0,L3) :-
   Y=[D,C,N,T,X],
   xDET([Y,Ppp,Pppp,Ppppp],L0,L1),
   noun1(W,[N,T,X],P),
   xWORD(W,L1,L2),
   xRELS([Y,Pp],L2,L3),
   and(P,Pp,Ppp).
/*
xNPxP([Y,Pp,Pppp],L0,L3) :-
   Y=[D,C,N,T,X],
   xDET([Y,P,Pp,Ppp],L0,L1),
   Yp=[Dp,of,Np,Tp,Xp],
   noun2(W,[N,T,X],[Tp,Xp],P),
   xWORD(W,L1,L2),
```

```
   dif(Dp,rel),
   xARG([Yp,Ppp,Pppp],L2,L3). */

xNPxP([Y,Pp,Pppp],L0,L3) :-
   Y=[D,C,N,T,X],
   xDET([Y,P,Pp,Ppp],L0,L1),
   Yp=[Dp,of,Np,Tp,Xp],
   noun2(W,[N,T,X],[Tp,Xp],P),
   xWORD(W,L1,L2),
   dif(Dp,rel),
   xARG([Yp,Ppp,Pppp],L2,L3).

/* 5 Right possesive adjunction */

xRADJ([Y,Y,P,P],L0,L0).
xRADJ([Y,Ypp,Pp,Pppp],L0,L3) :-
   Ypp=[Dpp,Cpp,Npp,Tpp,Xpp],
   Yp=[empty,Cp,Np,Tp,Xp],
   xWORD(s,L0,L1),
   xWORD(W,L1,L2),
   noun2(W,[Np,Tp,Xp],[Tpp,Xpp],Ppp),
   xRADJ([Y,Yp,P,Pp],L2,L3),
   semantics(Yp,Ppp,Pp,Pppp).

/*
xRADJ([Yp,Ypp,P,Pppp],L0,L3) :-
   Ypp=[Dpp,Cpp,Npp,Tpp,Xpp],
   Yp=[poss,Cp,Np,Tp,Xp],
   xWORD(s,L0,L1),
   xWORD(W,L1,L2),
   noun2(W,[Np,Tp,Xp],[Tpp,Xpp],Ppp),
   semanticx(Yp,Ppp,Pp,Pppp),
   xRADJ([Y,Yp,P,Pp],L2,L3).*/


/* 6 Interogative  pronouns */

xPRON([Y,Pp,[T,wh(X,Ppp)]],L0,L1) :-
   Y=[D,C,N,T,X],
   pron(W,[intero,C,T]),
   xWORD(W,L0,L1),
   domain(T,X,P),
   and(P,Pp,Ppp).

/* 7 Relative pronouns */

xPRON([Y,P,rel(Yp,P)],L0,L1) :-
   Y=[rel,C,N,T,X],
   Yp=[Dp,Cp,N,T,X],
   pron(W,[rel,C,T]),
   xWORD(W,L0,L1).

pron(that,[rel,C,T]) :- dif(T,hum).
pron(what,[intero,C,T]) :- dif(T,hum).
pron(which,[rel,C,T]) :- dif(T,hum).
pron(who,[D,suj,hum]).
pron(whom,[D,C,hum]) :- dif(C,suj).
```

```
/* 8 Determiner */

xDET([Y,P,Pp,Ppp],L0,L0) :-
   Y=[a,C,plu,T,X],
   semantics(Y,P,Pp,Ppp).
xDET([Y,P,Pp,Ppp],L0,L1) :-
   Y=[D,C,N,T,X],
   det(W,[D,N]),
   xWORD(W,L0,L1),
   semantics(Y,P,Pp,Ppp).
xDET([Y,P,Pp,number(E,Ep,X,and(P,Pp))],L0,L2) :-
   Y=[D,C,N,T,X],
   xEXACTLY(E,L0,L1),
   xWORD(W,L1,L2),
   detnum(W,[Ep,N]).

xEXACTLY(exactly,L0,L0).
xEXACTLY(atleast,L0,L2) :-
   xWORD(at,L0,L1),
   xWORD(least,L1,L2).
xEXACTLY(atmost,L0,L2) :-
   xWORD(at,L0,L1),
   xWORD(most,L1,L2).
xEXACTLY(atmost,L0,L2) :-
   xWORD(up,L0,L1),
   xWORD(to,L1,L2).
xEXACTLY(exactly,L0,L1) :-
   xWORD(exactly,L0,L1).

det(a,[a,sin]).
det(an,[a,sin]).
det(no,[no,N]).
det(the,[the,N]).
det(what,[intero,N]).

detnum(five,[5,plu]).
detnum(four,[4,plu]).
detnum(one,[1,sin]).
detnum(three,[3,plu]).
detnum(two,[2,plu]).

/* 9 Clauses minus one argument */

xVERBxCOMPL([Y,Pp],L0,L2) :-
   Y=[D,C,N,T,X],
   Yp=[Dp,Cp,Np,Tp,Xp],
   verb2(W,[N,T,X],[Cp,Tp,Xp],P),
   xWORD(W,L0,L1),
   xARG([Yp,P,Pp],L1,L2).
xVERBxCOMPL([Y,Pp],L0,L2) :-
   Y=[D,C,N,T,X],
   Yp=[Dp,Cp,Np,T,Xp],
   be(W,[N]),
   xWORD(W,L0,L1),
   xARG([Yp,equal(X,Xp),Pp],L1,L2).
xVERBxCOMPL([Y,Pp],L0,L2) :-
   Y=[D,C,N,T,X],
   be(W,[N]),
```

```
   xWORD(W,L0,L1),
   xADJV([Y,Pp],L1,L2).

xSUJxVERBxPREP([Y,Pp],L0,L3) :-
   Y=[D,C,N,T,X],
   Yp=[Dp,Cp,Np,Tp,Xp],
   xARG([Yp,P,Pp],L0,L1),
   xWORD(W,L1,L2),
   verb2(W,[Np,Tp,Xp],[C,T,X],P),
   xPREPxDANGLE(Y,L2,L3).

xADJV([Y,P],L0,L2) :-
   Y=[D,C,N,T,X],
   Yp=[Dp,Cp,Np,Tp,Xp],
   dif(Dp,intero),
   adjective2(W,[T,X],[Cp,Tp,Xp],P),
   xWORD(W,L0,L1),
   xARG([Yp,P,Pp],L1,L2).

xPREPxDANGLE(Y,L0,L0).
xPREPxDANGLE(Y,L0,L1) :- Y=[D,C,N,T,X], prep(C), xWORD(C,L0,L1).

be(s,[sin]).
be(is,[sin]).
be(are,[plu]).

/* 10 Conjunction of relative clauses */

xRELS([Yp,Ppp],L0,L2) :-
   Yp=[D,Cp,N,T,X],
   Y=[rel,C,N,T,X],
   xCLAUSE(rel(Y,P),L0,L1),
   xANDxRELS([Yp,Pp],L1,L2),
   and(P,Pp,Ppp).

xANDxRELS([Y,true],L0,L0).
xANDxRELS([Y,P],L0,L2) :-
   xWORD(and,L0,L1),
   xRELS([Y,P],L1,L2).

/* 11 Lexical entries */

noun0(W) :- var(W), !, W='<number of km>'.
noun0(W) :- do(Wp,diameter(Wp),W).

noun0(W,[bod]) :- var(W), !, W='<a body>'.
noun0(W,[bod]) :- do(Wp,body(Wp),W).
noun0(W,[hum]) :- var(W), !, W='<an astronomer>'.
noun0(W,[hum]) :- do(Wp,astronomer(Wp),W).
noun0(W,[num]) :- var(W), !, W='<an integer>'.
noun0(W,[num]) :- integer(W), W >= 0.

noun1(astronomer,[sin,hum,X],astronomer(X)).
noun1(astronomers,[plu,hum,X],astronomer(X)).
noun1(body,[sin,bod,X],body(X)).
noun1(bodies,[plu,bod,X],body(X)).
noun1(diameter,[sin,mea,X],diameter(X)).
noun1(diameters,[plu,mea,X],diameter(X)).
```

```
noun1('Earth',[sing,bod,X],theearth(X)).
noun1('Moon',[sin,bod,X],themoon(X)).
noun1(moon,[sin,bod,X],moon(X)).
noun1(moons,[plu,bod,X],moon(X)).
noun1(planet,[sin,bod,X],planet(X)).
noun1(planets,[plu,bod,X],planet(X)).
noun1(satellite,[sin,bod,X],satellite(X)).
noun1(satellites,[plu,bod,X],satellite(X)).
noun1('Sun',[sin,bod,X],thesun(X)).

noun2(diameter,[sin,mea,X],[bod,Xp],diameterof(X,Xp)).
noun2(diameters,[plu,mea,X],[bod,Xp],diameterof(X,Xp)).
noun2(moon,[sin,bod,X],[bod,Xp],moonof(X,Xp)).
noun2(moons,[plu,bod,X],[bod,Xp],moonof(X,Xp)).
noun2(satellite,[sin,bod,X],[bod,Xp],satelliteof(X,Xp)).
noun2(satellites,[plu,bod,X],[bod,Xp],satelliteof(X,Xp)).

verb2(discover,[N,hum,X],[obj,bod,Xp],discoveredby(Xp,X)).
verb2(discovers,[sin,hum,X],[obj,bod,Xp],discoveredby(Xp,X)).
verb2(discovered,[N,hum,X],[obj,bod,Xp],discoveredby(Xp,X)).
verb2(exceed,[N,mea,X],[obj,mea,Xp],less(Xp,X)).
verb2(exceeds,[sin,mea,X],[obj,mea,Xp],less(Xp,X)).
verb2(revolve,[N,bod,X],[around,bod,Xp],satelliteof(X,Xp)).
verb2(revolves,[sin,bod,X],[around,bod,Xp],satelliteof(X,Xp)).

adjective2(equal,[T,X],[than,T,Xp],equal(Xp,X)) :- inlist(T,[mea,num]).
adjective2(greater,[T,X],[than,T,Xp],less(Xp,X)) :- inlist(T,[mea,num]).
adjective2(less,[T,X],[than,T,Xp],less(X,Xp)) :- inlist(T,[mea,num]).

/* 12 Semantics */

semantics([a,C,sin,T,X],P,Pp,number(atleast,1,X,and(P,Pp))).
semantics([a,C,plu,T,X],P,Pp,number(atleast,2,X,and(P,Pp))).
semantics([intero,C,N,T,X],P,Pp,[T,wh(X,and(P,Pp))]).
semantics([no,C,N,T,X],P,Pp,neg(exists(X,and(P,Pp)))).
semantics([empty,C,N,T,X],P,Pp,exists(X,and(P,Pp))).
semantics([the,C,N,T,X],P,Pp,the(N,X,P,Pp)).

domain(hum,X,astronomer(X)).
domain(mea,X,diameter(X)).
domain(bod,X,body(X)).

and(true,true,true).
and(true,P,P) :- dif(P,true).
and(P,true,P) :- dif(P,true).
and(P,Pp,and(P,Pp)) :- dif(P,true), dif(Pp,true).
```

9 Annexe : grammaire du français

```
/* 1 Proposition principale */

phraz(stop,[stoppe,'!']).
phraz(P,L0) :- P = [T,Pp], zPROP(P,L0,L1), zMOT('?',L1,[]).

zMOT(M,[M|L],L).

/* 2 Proposition interrogative et relative */
```

```
zPROP(Ppp,L0,L3) :-
   Yp=[intero,suj,G,N,T,X],
   Y=[D,suj,G,N,T,X],
   zART([Yp,P,Pp,Ppp],L0,L1),
   etre(M,[N]),
   zMOT(M,L1,L2),
   zGN([Y,true,the(N,X,P,Pp)],L2,L3).
zPROP(rel(Yp,Pp),L0,L3) :-
   Yp=[rel,suj,G,N,T,X],
   Y=[D,suj,G,N,T,X],
   zPRONzREL([Yp,P,Pp],L0,L1),
   etre(M,[N]),
   zMOT(M,L1,L2),
   zGN([Y,true,P],L2,L3).
zPROP(Pp,L0,L3) :-
   zGN([Y,P,Pp],L0,L1),
   zESTCEQ(Y,L1,L2),
   zPROP1([Y,P],L2,L3).
zPROP(Ppp,L0,L3) :-
   Yp=[Dp,de,Gp,Np,Tp,Xp],
   zGN([Yp,Pp,Ppp],L0,L1),
   zGNzMOINS([Y,P],[Yp,Pp],L1,L2),
   zVERBEzCOMPL([Y,P],L2,L3).
zPROP(Ppp,L0,L3) :-
   Yp=[Dp,de,Gp,Np,Tp,Xp],
   zGN([Yp,P,Ppp],L0,L1),
   zSUJzVERBE([Y,Pp],L1,L2),
   zGNzMOINS([Y,P],[Yp,Pp],L2,L3).

zESTCEQ(Y,L0,L0).
zESTCEQ(Y,L0,L3) :-
   Y=[intero,C,G,N,T,X],
   zMOT(est,L0,L1),
   zMOT(ce,L1,L2),
   quei(M,[C]),
   zMOT(M,L2,L3).


quei(qu,[C]) :- dif(C,suj).
quei(que,[C]) :- dif(C,suj).
quei(qui,[suj]).

/* 3 Groupe nominal */

zGN([Y,P,Pp],L0,L2) :-
   Y=[autourde,C,G,N,T,X],
   Yp=[de,C,G,N,T,X],
   zMOT(autour,L0,L1),
   zGN([Yp,P,Pp],L1,L2).
zGN([Y,P,Pp],L0,L1) :-
   zPRONzREL([Y,P,Pp],L0,L1).
zGN([Y,P,Pp],L0,L1) :-
   Y=[D,C,G,N,T,X],
   dif(D,rel),
   zGNzP([Y,P,Pp],L0,L1).

zGNzP([Y,P,Pp],L0,L1) :-
   Y=[D,C,G,N,T,X],
```

```
   zPRONzINTERO([Y,P,Pp],L0,L1).
zGNzP([Y,P,Ppp],L0,L4) :-
   Yp=[Dp,Cp,mas,sin,num,Xp],
   Y=[vide,C,G,N,T,X],
   zART([Yp,true,equal(Xp,card(X,Pp)),Ppp],L0,L1),
   zMOT(nombre,L1,L2),
   zMOT(de,L2,L3),
   zGNzP([Y,P,exists(X,Pp)],L3,L4).
zGNzP([Y,P,P],L0,L2) :-
   Y=[propre,C,G,sin,T,X],
   zCAS(Y,L0,L1),
   zMOT(M,L1,L2),
   nom0(M,[G,T,X]).

zGNzP([Y,P,P],L0,L3) :-
   Y=[propre,C,G,N,mea,M],
   zCAS(Y,L0,L1),
   zMOT(M,L1,L2),
   nom0(M),
   zMOT(km,L2,L3).

 zGNzP([Y,Pppp,Ppppp],L0,L3) :-
   Y=[D,C,G,N,T,X],
   zART([Y,Ppp,Pppp,Ppppp],L0,L1),
   nom1(M,[G,N,T,X],P),
   zMOT(M,L1,L2),
   zQUALSzEV([Y,Pp],L2,L3),
   and(P,Pp,Ppp).
zGNzP([Y,Pp,Pppp],L0,L3) :-
   Y=[D,C,G,N,T,X],
   zART([Y,P,Pp,Ppp],L0,L1),
   Yp=[Dp,de,Gp,Np,Tp,Xp],
   nom2(M,[G,N,T,X],[Tp,Xp],P),
   zMOT(M,L1,L2),
   zGN([Yp,Ppp,Pppp],L2,L3).

/* 4 Groupe nominal incomplet */

zGNzMOINS([Y,Pp],[Yppp,Pppp],L0,L3) :-
   Y=[D,C,G,N,T,X],
   Yp=[Dp,de,Gp,Np,Tp,Xp],
   zART([Y,P,Pp,Ppp],L0,L1),
   nom2(M,[G,N,T,X],[Tp,Xp],P),
   zMOT(M,L1,L2),
   zGNzMOINSzSUITE([Yp,Ppp],[Yppp,Pppp],L2,L3).
zGNzMOINSzSUITE([Y,P],[Y,P],L0,L0).
zGNzMOINSzSUITE([Y,P],[Yp,Pp],L0,L1) :-
   zGNzMOINS([Y,P],[Yp,Pp],L0,L1).

/* 5 Article */

zART([Y,P,Pp,Ppp],L0,L1) :-
   Y=[D,C,G,N,T,X],
   art(M,[D,C,G,N]),
   zMOT(M,L0,L1),
   semantique(Y,P,Pp,Ppp).
zART([Y,P,Pp,Ppp],L0,L2) :-
   Y=[D,C,G,N,T,X],
```

```
   zCAS(Y,L0,L1),
   outlist([C,M],[[a,le],[a,les],[de,le],[de,des],[de,les]]),
   zARTzSC([Y,P,Pp,Ppp],L1,L2).

zARTzSC([Y,P,Pp,Ppp],L0,L0) :- Y=[vide,C,G,N,T,X], semantique(Y,P,Pp,Ppp).
zARTzSC([Y,P,Pp,Ppp],L0,L2) :-
   Y=[interocard,C,G,plu,T,X],
   zMOT(combien,L0,L1),
   zMOT(M,L1,L2),
   de(M),
   semantique(Y,P,Pp,Ppp).
zARTzSC([Y,P,Pp,Ppp],L0,L1) :-
   Y=[D,C,G,N,T,X],
   zMOT(M,L0,L1),
   artsc(M,[D,G,N]),
   semantique(Y,P,Pp,Ppp).
zARTzSC([Y,P,Pp,Ppp],L0,L2) :-
   Y=[nombre,C,G,N,T,X],
   zEXACT(E,Ep,L0,L1),
   zMOT(M,L1,L2),
   artscnum(M,[E,Np,G,N]),
   semantique([nombre,C,G,[Ep,Np],T,X],P,Pp,Ppp).

de(d).
de(de).

zEXACT(E,E,L0,L0).
zEXACT(E,atleast,L0,L2) :- zMOT(au,L0,L1), zMOT(moins,L1,L2).
zEXACT(E,atmost,L0,L2) :- zMOT(au,L0,L1), zMOT(plus,L1,L2).
zEXACT(E,exactly,L0,L1) :- zMOT(exactement,L0,L1).

art(au,[def,hum,mas,sin]).
art(aux,[def,hum,mas,plu]).
art(d,[def,de,G,plu]).
art(de,[def,de,G,plu]).
art(des,[def,de,G,plu]).
art(du,[def,de,mas,sin]).

artsc(aucun,[aucun,mas,sin]).
artsc(aucune,[aucun,fem,sin]).
artsc(des,[des,G,plu]).
artsc(l,[def,G,sin]).
artsc(la,[def,fem,sin]).
artsc(le,[def,mas,sin]).
artsc(les,[def,G,plu]).
artsc(quel,[intero,mas,sin]).
artsc(quelle,[intero,fem,sin]).
artsc(quelles,[intero,fem,plu]).
artsc(quels,[intero,mas,plu]).

artscnum(cinq,[exactly,5,G,plu]).
artscnum(deux,[exactly,2,G,plu]).
artscnum(quatre,[exactly,4,G,plu]).
artscnum(trois,[exactly,3,G,plu]).
artscnum(un,[atleast,1,mas,sin]).
artscnum(une,[atleast,1,fem,sin]).
artscnum(M,[E,M,G,N]) :- var(M), !, M='<entier naturel>'.
artscnum(M,[E,M,G,N]) :- integer(M), M >= 0.
```

```
/* 6 Pronom interrogatif */

zPRONzINTERO([Y,Pp,[T,wh(X,Ppp)]],L0,L1) :-
   Y=[intero,C,G,N,T,X],
   pronomintero(M,[C,T]),
   zMOT(M,L0,L1),
   domain(T,X,P),
   and(P,Pp,Ppp).
zPRONzINTERO([Y,Pp,[T,wh(X,Ppp)]],L0,L2) :-
   Y=[intero,C,G,N,T,X],
   zPREP(Y,L0,L1),
   pronominterosc(M,[T]),
   zMOT(M,L1,L2),
   domain(T,X,P),
   and(P,Pp,Ppp).

pronomintero(qu,[C,T]) :- inlist(C,[obj,suj]), dif(T,hum).
pronomintero(que,[C,T]) :- inlist(C,[obj,suj]), dif(T,hum).
pronomintero(qui,[C,hum]) :- inlist(C,[obj,suj]).
pronomintero(quoi,[obj,T]) :- dif(T,hum).

pronominterosc(qui,[hum]).
pronominterosc(quoi,[T]) :- dif(T,hum).

/* 7 Pronom relatif */

zPRONzREL([Y,P,rel(Yp,P)],L0,L1) :-
   Y=[rel,C,G,N,T,X],
   Yp=[Dp,Cp,G,N,T,X],
   pronomrel(M,[C]),
   zMOT(M,L0,L1).
zPRONzREL([Y,P,rel(Yp,P)],L0,L2) :-
   Y=[rel,C,G,N,T,X],
   Yp=[Dp,Cp,G,N,T,X],
   zPREP(Y,L0,L1),
   pronomrelsc(Mp,[C,T]),
   zMOT(Mp,L1,L2).

pronomrel(dont,[de]).
pronomrel(qu,[obj]).
pronomrel(que,[obj]).
pronomrel(qui,[suj]).

pronomrelsc(qui,[C,hum]) :- dif(C,obj), dif(C,suj).
pronomrelsc(quoi,[C,T]) :- dif(C,suj), dif(C,obj), dif(T,hum).

/* 8 Cas et preposition */

zCAS(Y,L0,L0) :- Y=[D,C,G,N,T,X], inlist(C,[obj,suj]).
zCAS(Y,L0,L1) :- zPREP(Y,L0,L1).

zPREP(Y,L0,L1) :- Y=[D,C,G,N,T,X], prep(M,[C]), zMOT(M,L0,L1).
zPREP(Y,L0,L2) :-
   Y=[D,autourde,G,N,T,X],
   zMOT(autour,L0,L1),
   Yp=[D,de,G,N,T,X],
   zPREP(Yp,L1,L2).
```

```
prep(a,[a]).
prep(d,[de]).
prep(de,[de]).

/* 9 Propositions moins un argument */

zPROP1([Y,P],L0,L1) :- zVERBEzCOMPL([Y,P],L0,L1).
zPROP1([Y,P],L0,L1) :- zSUJzVERBE([Y,P],L0,L1).
zPROP1([Y,P],L0,L1) :- zVERBEzSUJ([Y,P],L0,L1).

zVERBEzCOMPL([Y,Pp],L0,L2) :-
   Y=[D,suj,G,N,T,X],
   Yp=[Dp,Cp,Gp,Np,Tp,Xp],
   outlist(Dp,[intero,rel]),
   zPROP2([Y,Yp,P],L0,L1),
   zGN([Yp,P,Pp],L1,L2).
zSUJzVERBE([Y,Pp],L0,L2) :-
   Y=[D,C,G,N,T,X],
   Yp=[Dp,suj,Gp,Np,Tp,Xp],
   dif(C,suj),
   dif(Dp,intero),
   outlist(Dp,[intero,rel]),
   zGN([Yp,P,Pp],L0,L1),
   zPROP2([Yp,Y,P],L1,L2).
 zVERBEzSUJ([Y,Pp],L0,L2) :-
   Y=[D,C,G,N,T,X],
   Yp=[Dp,suj,Gp,Np,Tp,Xp],
   outlist(Dp,[intero,rel]),
   zPROP2([Yp,Y,P],L0,L1),
   zGN([Yp,P,Pp],L1,L2).

/* 10 Propositions moins deux arguments */

zPROP2([Y,Yp,P],L0,L4) :-
   Y=[D,suj,G,N,T,X],
   Yp=[Dp,Cp,Gp,Np,Tp,Xp],
   zNE([Y,Yp],L0,L1),
   zMOT(M,L1,L2),
   auxverbe(M,I,[N,T,X],[Cp,Tp,Xp],P),
   zTIL(L2,L3),
   zPPzADJ(I,[Y,Yp,P],L3,L4).

zPPzADJ(I,[Y,Yp,P],L0,L0).
zPPzADJ(I,[Y,Yp,P],L0,L1) :-
   Y=[D,suj,G,N,T,X],
   Yp=[Dp,Cp,Gp,Np,Tp,Xp],
   zMOT(M,L0,L1),
   ppad(M,I,[G,N,T,X],[Cp,Gp,Np,Tp,Xp],P).

auxverbe(M,1,[N,T,X],[Cp,Tp,Xp],P) :- avoir(M,[N]).
auxverbe(M,2,[N,T,X],[Cp,Tp,Xp],P) :- etre(M,[N]).
auxverbe(M,3,[N,T,X],[Cp,Tp,Xp],P) :- verbe2(M,[N,T,X],[Cp,Tp,Xp],P).

ppad(Mp,1,[G,N,T,X],[Cp,Gp,Np,Tp,Xp],P) :- ppavoir2(Mp,[T,X],[Cp,Tp,Xp],P).
ppad(Mp,1,[G,N,T,X],[Cp,Gp,Np,Tp,Xp],P) :-
   ppavoirobj2(Mp,[T,X],[Gp,Np,Tp,Xp],P).
ppad(Mp,2,[G,N,T,X],[Cp,Gp,Np,Tp,Xp],P) :-
```

```
   adjectif2(Mp,[G,N,T,X],[Cp,Tp,Xp],P).


zNE([Y,Yp],L0,L0) :-
   Y=[D,C,G,N,T,X],
   Yp=[Dp,Cp,Gp,Np,Tp,Xp],
   outlist([D,Dp],[[no,Dp],[D,no]]).
zNE([Y,Yp],L0,L1) :-
   Y=[D,C,G,N,T,X],
   Yp=[Dp,Cp,Gp,Np,Tp,Xp],
   inlist([D,Dp],[[no,Dp],[D,no]]),
   zMOT(M,L0,L1),
   ne(M).

zTIL(L0,L0).
zTIL(L0,L1) :- Y=[D,C,G,N,T,X], pronompers(M,[G,N]), zMOT(M,L0,L1).

ne(n).
ne(ne).

etre(est,[sin]).
etre(sont,[plu]).

avoir(a,[sin]).
avoir(ont,[plu]).

pronompers(elle,[fem,sin]).
pronompers(elles,[fem,plu]).
pronompers(il,[mas,sin]).
pronompers(ils,[mas,plu]).

/* 11 Conjonction de qualicatifs */

zQUALSzEV([Y,P],L0,L1) :- zQUALS([Y,P],L0,L1).
zQUALSzEV([Y,P],L0,L1) :- zRELSzEV([Y,P],L0,L1).

zQUALS([Y,Ppp],L0,L2) :-
   xADJF([Y,P],L0,L1),
   zETzQUALS([Y,Pp],L1,L2),
   and(P,Pp,Ppp).

zETzQUALS([Y,P],L0,L0).
zETzQUALS([Y,P],L0,L2) :-
   zMOT(et,L0,L1),
   zQUALS([Y,P],L1,L2).
zETzQUALS([Y,P],L0,L2) :-
   zMOT(et,L0,L1),
   zRELS([Y,P],L1,L2).

zRELSzEV([Y,true],L0,L0).
zRELSzEV([Y,P],L0,L1) :- zRELS([Y,P],L0,L1).

zRELS([Yp,Ppp],L0,L2) :-
   Yp=[D,Cp,G,N,T,X],
   Y=[rel,C,G,N,T,X],
   zPROP(rel(Y,P),L0,L1),
   zETzRELS([Yp,Pp],L1,L2),
   and(P,Pp,Ppp).
```

```
zETzRELS([Y,true],L0,L0).
zETzRELS([Y,P],L0,L2) :-
   zMOT(et,L0,L1),
   zRELS([Y,P],L1,L2).

xADJF([Y,Pp],L0,L2) :-
   Y=[D,C,G,N,T,X],
   Yp=[Dp,Cp,Gp,Np,Tp,Xp],
   dif(Dp,intero),
   adjectif2(M,[G,N,T,X],[Cp,Tp,Xp],P),
   zMOT(M,L0,L1),
   zGN([Yp,P,Pp],L1,L2).

/* 12 Entree lexicale */

nom0(M) :- var(M), !, M='<nombre de km>'.
nom0(M) :- do(Mp,diameter(Mp),M).

nom0(M,[mas,hum,M]) :- var(M), !, M='<un astronome>'.
nom0(M,[mas,hum,Mp]) :- do(Mpp,astronomer(Mpp),Mp), translate(french,hum,Mp,M).
nom0(M,[G,bod,M]) :- var(M), !, M='<un astre>'.
nom0(M,[G,bod,Mp]) :- do(Mpp,body(Mpp),Mp), translate(french,bod,Mp,M).
nom0(M,[G,num,M]) :- var(M), !, M='<un entier>'.
nom0(M,[G,num,Mp]) :- integer(W), W >= 0.

nom1(astre,[mas,sin,bod,X],body(X)).
nom1(astres,[mas,plu,bod,X],body(X)).
nom1(astronome,[mas,sin,hum,X],astronomer(X)).
nom1(astronomes,[mas,plu,hum,X],astronomer(X)).
nom1(diametre,[mas,sin,mea,X],diameter(X)).
nom1(diametres,[mas,plu,mea,X],diameter(X)).
%nom1(km,[mas,N,mea,X],km).
nom1(lune,[fem,sin,bod,X],themoon(X)).
nom1(lunes,[fem,plu,bod,X],moon(X)).
nom1(planete,[fem,sin,bod,X],planet(X)).
nom1(planetes,[fem,plu,bod,X],planet(X)).
nom1(satellite,[mas,sin,bod,X],satellite(X)).
nom1(satellites,[mas,plu,bod,X],satellite(X)).
nom1(soleil,[mas,sin,bod,X],thesun(X)).
nom1(terre,[fem,sin,bod,X],theearth(X)).

nom2(diametre,[mas,sin,mea,X],[bod,Xp],diameterof(X,Xp)).
nom2(diametres,[mas,plu,mea,X],[bod,Xp],diameterof(X,Xp)).
nom2(lune,[fem,sin,bod,X],[bod,Xp],moonof(X,Xp)).
nom2(lunes,[fem,plu,bod,X],[bod,Xp],moonof(X,Xp)).
nom2(satellite,[mas,sin,bod,X],[bod,Xp],satelliteof(X,Xp)).
nom2(satellites,[mas,plu,bod,X],[bod,Xp],satelliteof(X,Xp)).

verbe2(decouvrit,[sin,hum,X],[obj,bod,Xp],discoveredby(Xp,X)).
verbe2(gravite,[sin,bod,X],[autourde,bod,Xp],satelliteof(X,Xp)).
verbe2(gravitent,[plu,bod,X],[autourde,bod,Xp],satelliteof(X,Xp)).
verbe2(tourne,[sin,bod,X],[autourde,bod,Xp],satelliteof(X,Xp)).
verbe2(tournent,[plu,bod,X],[autourde,bod,Xp],satelliteof(X,Xp)).

ppavoir2(decouvert,[hum,X],[obj,bod,Xp],discoveredby(Xp,X)).

ppavoirobj2(decouvert,[hum,X],[mas,sin,bod,Xp],discoveredby(Xp,X)).
```

```
ppavoirobj2(decouverte,[hum,X],[fem,sin,bod,Xp],discoveredby(Xp,X)).
ppavoirobj2(decouvertes,[hum,X],[fem,plu,bod,Xp],discoveredby(Xp,X)).
ppavoirobj2(decouverts,[hum,X],[mas,plu,bod,Xp],discoveredby(Xp,X)).

adjectif2(egal,[mas,sin,T,X],[a,T,Xp],equal(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(egale,[fem,sin,T,X],[a,T,Xp],equal(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(egales,[fem,plu,T,X],[a,T,Xp],equal(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(egaux,[mas,plu,T,X],[a,T,Xp],equal(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(inferieur,[mas,sin,T,X],[a,T,Xp],less(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(inferieure,[fem,sin,T,X],[a,T,Xp],less(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(inferieures,[fem,plu,T,X],[a,T,Xp],less(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(inferieurs,[mas,plu,T,X],[a,T,Xp],less(X,Xp)) :- inlist(T,[mea,num]).
adjectif2(superieur,[mas,sin,T,X],[a,T,Xp],less(Xp,X)) :- inlist(T,[mea,num]).
adjectif2(superieure,[fem,sin,T,X],[a,T,Xp],less(Xp,X)) :- inlist(T,[mea,num]).
adjectif2(superieures,[fem,plu,T,X],[a,T,Xp],less(Xp,X)) :- inlist(T,[mea,num]).
adjectif2(superieurs,[mas,plu,T,X],[a,T,Xp],less(Xp,X)) :- inlist(T,[mea,num]).

/* 13 Semantique */

semantique([def,C,G,N,T,X],P,Pp,the(N,X,P,Pp)).
semantique([des,C,G,plu,T,X],P,Pp,number(atleast,2,X,and(P,Pp))).
semantique([intero,C,G,N,T,X],P,Pp,[T,wh(X,and(P,Pp))]).
semantique([interocard,C,G,plu,T,X],P,Pp,[num,wh(Xp,equal(Xp,card(X,and(P,Pp))))]).
semantique([aucun,C,G,N,T,X],P,Pp,number(exactly,0,X,and(P,Pp))).
semantique([nombre,C,G,[E,N],T,X],P,Pp,number(E,N,X,and(P,Pp))) :-
   inlist(T,[bod,hum]).
semantique([nombre,C,G,[E,N],mea,N],P,Pp,and(P,Pp)).
semantique([propre,C,G,sin,T,X],true,Pp,Pp).
semantique([vide,C,G,N,T,X],P,Pp,exists(X,and(P,Pp))).
```

10 Annexe : entrée

```
/* Read next sentence from a file extended to the keyboard */

readsentence(K2) :-
   readstring(L1), addspaces(L1,L2), makeunits(L2,K1), preprocessing(K1,K2).

/* Read string */

readstring(L) :- nextchar(C), readcomment(0,' ',C,[' '|L]).

readcomment(0,C,Cp,[C,Cp]) :- end(Cp), !.
readcomment(0,C,Cp,L) :- dummy(Cp), !, nextchar(Cpp), readcomment(0,C,Cpp,L).
readcomment(0,C,Cp,L) :- char_code(Cp,N), return(N), !, readcomment(0,C,' ',L).
readcomment(N,'/','*',L) :- !, Np is N+1, nextchar(C), readcomment(Np,'*',C,L).
readcomment(0,C,Cp,[C|L]) :- !, nextchar(Cpp), readcomment(0,Cp,Cpp,L).
readcomment(N,'*','/',L) :- N > 0, !, Np is N-1, nextchar(C), readcomment(Np,' ',C,L).
readcomment(N,C,'%',L) :-  N > 0, !, nextchar(Cp), readcommentline(N,Cp,L).
readcomment(N,C,Cp,L) :-  N > 0, !, nextchar(Cpp), readcomment(N,Cp,Cpp,L).

readcommentline(N,C,L) :- char_code(C,N), return(N), !, nextchar(Cp), readcomment(N,' ',Cp,L).
readcommentline(N,C,L) :- nextchar(Cp), readcommentline(N,Cp,L).

dummy('{').  dummy('}').
end('?').    end('!').   end('.').
return(10).  return(13).

nextchar(Cp) :-  state(keyboard), !, get_char(Cp).
```

```
nextchar(Cp) :-  get_char(sentences,C), tothekeybord(C,Cp).

tothekeybord(end_of_file,C) :-
   !, retract(state(file)), asserta(state(keyboard)), nextchar(C).
tothekeybord(C,C) :- put_char(C).

/* Add some spaces */

addspaces([C],[C]).
addspaces([C,Cp|L],[C,' ',Cp|Lp]) :- case(C,Cp), !, addspaces([Cp|L],[Cp|Lp]).
addspaces([C,Cp|L],[C,Cp|Lp]) :-  addspaces([Cp|L],[Cp|Lp]).

case(C,Cp) :- letter(C), digit(Cp).
case(C,Cp) :- letter(C), special(Cp).
case(C,Cp) :- digit(C), letter(Cp).
case(C,Cp) :- digit(C), special(Cp).

/* Make words, grammar of attributs */

makeunits(L,K) :- yUNITS(K,L,[]).

yUNITS([],L0,L0).
yUNITS([D|K],L0,L3) :- ySPACES(L0,L1), yUNIT(D,L1,L2),  yUNITSEND(K,L2,L3).

ySPACES(L0,L2) :- yCHARACTER(' ',L0,L1), !, ySPACES(L1,L2).
ySPACES(L0,L0).

yUNITSEND([],L0,L0).
yUNITSEND(K,L0,L1) :- yUNITS(K,L0,L1).

yUNIT(D,L0,L2) :- yCHARACTER(C,L0,L1), letter(C), yWORDEND(K,L1,L2), atom_chars(D,[C|K]).
yUNIT(D,L0,L2) :- yCHARACTER(C,L0,L1), digit(C), yINTEGEREND(K,L1,L2), number_chars(D,[C|K]).
yUNIT(C,L0,L1) :- yCHARACTER(C,L0,L1), special(C).

yWORDEND([],L0,L1) :- yCHARACTER(' ',L0,L1).
yWORDEND([C|K],L0,L2) :- yCHARACTER(C,L0,L1), letter(C), yWORDEND(K,L1,L2).

yINTEGEREND([],L0,L1) :- yCHARACTER(' ',L0,L1).
yINTEGEREND([C|K],L0,L2) :- yCHARACTER(C,L0,L1), digit(C), yINTEGEREND(K,L1,L2).

yCHARACTER(C,[C|L0],L0).

digit(C) :- char_code(C,N), 47 < N, N < 58.

letter(C) :- char_code(C,N), 64 < N, N < 91, !.
letter(C) :- char_code(C,N), 96 < N, N < 123.

special(C) :- char_code(C,N), -1 < N, N < 48, C \= ' ', !.
special(C) :- char_code(C,N), 57 < N, N < 65, !.
special(C) :- char_code(C,N), 90 < N, N < 97, !.
special(C) :- char_code(C,N), 122 < N, N < 128.

/* Preprocessing */

preprocessing(['t'|L],Lp)  :- !, preprocessing(L,Lp).
preprocessing([''''|L],Lp)  :- !, preprocessing(L,Lp).
preprocessing(['-'|L],Lp)  :- !, preprocessing(L,Lp).
preprocessing([C|L],[C|Lp])  :- preprocessing(L,Lp).
```

```
preprocessing([],[]).

/* Read string test */

testread([N|L]) :- nextchar(C), char_code(C,N), testreadbis([N|L]).

testreadbis([46]) :- !.
testreadbis([N,Np|L]) :- testread([Np|L]).


/* Tests

toujours.
toujours :- toujours.

:- open('Prolog/Orbis1/sentencesTotal1.pl',read,X,[alias(sentences)]),
   toujours,
   readsentence(L), nl, write(L), nl,
   makewords(L,Lp), write(Lp), nl,
   fail.
*/
```

11 Annexe : exemples

```
/* Exemples  */

Qu'est-ce qui tournent autour de Saturne ?
[que,est,ce,qui,tournent,autour,de,Saturne,?]
Requete : [bod,wh(#0,and(body(#0),satelliteof(#0,Saturn)))]
Reponse :
>> 1980S1
>> 1980S13
>> 1980S25
>> 1980S26
>> 1980S27
>> 1980S28
>> 1980S3
>> 1980S6
>> Dione
>> Encelade
>> Hyperion
>> Japet
>> Mimas
>> Phoebe
>> Rhea
>> Tethys
>> Titan

Quelle lune est-ce que Nicholson a decouverte ?
[quelle,lune,est,ce,que,Nicholson,a,decouverte,?]
Requete : [bod,wh(#0,and(themoon(#0),discoveredby(#0,Nicholson)))]
Reponse :
>> Aucun astre.

A quoi est-ce que le diametre d'Himalia est egal ?
[a,quoi,est,ce,que,le,diametre,d,',Himalia,est,egal,?]
[a,quoi,est,ce,que,le,diametre]
Mot possible suivant :
```

```
<entier naturel><nombre de km>  <un astronome>  a               au
aucun           aucune          autour          aux             cinq
combien         de              des             deux            diametre
diametres       dont            du              egal            est
exactement      inferieur       l               la              le
les             lune            lunes           nombre          quatre
que             quel            quelle          quelles         quels
qui             satellite       satellites      superieur       trois
un              une

Combien de satellites tournent autour de Saturne ?
[combien,de,satellites,tournent,autour,de,Saturne,?]
Requete : [num,wh(#0,equal(#0,card(#1,and(satellite(#1),satelliteof(#1,Saturn)))))]
Reponse :
>> 17

Qui decouvrit au plus un satellite ?
[qui,decouvrit,au,plus,un,satellite,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(atmost,1,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Reponse :
>> Bond
>> Dollfus
>> Fountain
>> Huygens
>> Lacques
>> Melotte
>> Perrine
>> Pickering
>> Smith
>> Tombaugh

Qui decouvrit exactement un satellite ?
[qui,decouvrit,exactement,un,satellite,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(exactly,1,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Reponse :
>> Bond
>> Dollfus
>> Fountain
>> Huygens
>> Lacques
>> Melotte
>> Perrine
>> Pickering
>> Smith
>> Tombaugh

Qui decouvrit un satellite ?
[qui,decouvrit,un,satellite,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(atleast,1,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Reponse :
>> Bond
>> Cassini
>> Dollfus
>> Fountain
>> Galilee
>> Hall
>> Herschel
>> Huygens
```

```
>> Kuiper
>> Lacques
>> Lassel
>> Melotte
>> Nicholson
>> Perrine
>> Pickering
>> Smith
>> Tombaugh

Qui decouvrit au plus deux satellites ?
[qui,decouvrit,au,plus,deux,satellites,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(atmost,2,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Reponse :
>> Bond
>> Dollfus
>> Fountain
>> Hall
>> Huygens
>> Kuiper
>> Lacques
>> Melotte
>> Perrine
>> Pickering
>> Smith
>> Tombaugh

Qui decouvrit exactement deux satellites ?
[qui,decouvrit,exactement,deux,satellites,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(exactly,2,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Reponse :
>> Hall
>> Kuiper

Qui decouvrit deux satellites ?
[qui,decouvrit,deux,satellites,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(exactly,2,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Reponse :
>> Hall
>> Kuiper

Qui decouvrit Europe ?
[qui,decouvrit,Europe,?]
Requete : [hum,wh(#0,and(astronomer(#0),discoveredby(Europa,#0)))]
Reponse :
>> Galilee

Qu'est-ce que Cassini a decouvert ?
[que,est,ce,que,Cassini,a,decouvert,?]
Requete : [bod,wh(#0,and(body(#0),discoveredby(#0,Cassini)))]
Reponse :
>> 1980S1
>> Dione
>> Japet
>> Rhea
>> Tethys

Quel astronome a decouvert un astre ?
```

```
[quel,astronome,a,decouvert,un,astre,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(atleast,1,#1,and(body(#1),discoveredby(#1,#0)))))]
Reponse :
>> Bond
>> Cassini
>> Dollfus
>> Fountain
>> Galilee
>> Hall
>> Herschel
>> Huygens
>> Kuiper
>> Lacques
>> Lassel
>> Melotte
>> Nicholson
>> Perrine
>> Pickering
>> Smith
>> Tombaugh

Autour de quoi est-ce que tourne la terre ?
[autour,de,quoi,est,ce,que,tourne,la,terre,?]
Requete : [bod,wh(#0,and(body(#0),the(sin,#1,theearth(#1),satelliteof(#1,#0))))]
Reponse :
>> le soleil

Autour de quoi est-ce que tourne un satellite de Mars ?
[autour,de,quoi,est,ce,que,tourne,un,satellite,de,Mars,?]
Requete : [bod,wh(#0,and(body(#0),number(atleast,1,#1,and(satelliteof(#1,Mars),satelliteof(#1,#0)))))]
Reponse :
>> Mars

Le satellite de quel astre a decouvert Herschel ?
[le,satellite,de,quel,astre,a,decouvert,Herschel,?]
Requete : [bod,wh(#0,and(body(#0),the(sin,#1,satelliteof(#1,#0),discoveredby(#1,Herschel))))]
Reponse :
>> Aucun astre.

Autour de quel astre un satellite tourne-t-il ?
[autour,de,quel,astre,un,satellite,tourne,il,?]
Requete : [bod,wh(#0,and(body(#0),number(atleast,1,#1,and(satellite(#1),satelliteof(#1,#0)))))]
Reponse :
>> la terre
>> Jupiter
>> Mars
>> Neptune
>> Pluton
>> Saturne
>> Uranus
>> le soleil

Quel est le diametre de Io ?
[quel,est,le,diametre,de,Io,?]
Requete : [mea,wh(#0,and(diameterof(#0,Io),true))]
Reponse :
>> 3638 km
```

```
Quels sont les satellites qui gravitent autour de Saturne ?
[quels,sont,les,satellites,qui,gravitent,autour,de,Saturne,?]
Requete : [bod,wh(#0,and(and(satellite(#0),satelliteof(#0,Saturn)),true))]
Reponse :
>> 1980S1
>> 1980S13
>> 1980S25
>> 1980S26
>> 1980S27
>> 1980S28
>> 1980S3
>> 1980S6
>> Dione
>> Encelade
>> Hyperion
>> Japet
>> Mimas
>> Phoebe
>> Rhea
>> Tethys
>> Titan

Quel est l astre dont le satellite gravite autour du soleil ?
[quel,est,l,astre,dont,le,satellite,gravite,autour,du,soleil,?]
[quel,est,l,astre,dont,le,satellite,gravite,autour]
Mot possible suivant :
autour          de

Quels sont les satellites de Jupiter ?
[quels,sont,les,satellites,de,Jupiter,?]
Requete : [bod,wh(#0,and(satelliteof(#0,Jupiter),true))]
Reponse :
>> 1979J2
>> 1979J3
>> Adrastee
>> Amalthee
>> Ananke
>> Callisto
>> Carme
>> Elara
>> Europe
>> Ganymede
>> Himalia
>> Io
>> Leda
>> Lysithee
>> Pasiphae
>> Sinope

Quel est l astre dont le satellite gravite autour de Mars ?
[quel,est,l,astre,dont,le,satellite,gravite,autour,de,Mars,?]
Requete : [bod,wh(#0,and(and(body(#0),the(sin,#1,satelliteof(#1,#0),satelliteof(#1,Mars))),true))]
Reponse :
>> Aucun astre.

Quel astre a decouvert Galilee ?
[quel,astre,a,decouvert,Galilee,?]
Requete : [bod,wh(#0,and(body(#0),discoveredby(#0,Galileo)))]
```

```
Reponse :
>> Callisto
>> Europe
>> Ganymede
>> Io

De quel astre Cassini a-t-il decouvert un satellite ?
[de,quel,astre,Cassini,a,il,decouvert,un,satellite,?]
[de,quel,astre,Cassini,a,il,decouvert,un,satellite]
Mot possible suivant :
<entier naturel>a               au              aucun           autour
de              des             dont            du              exactement
l               le              quel            un

Qui decouvrit Europe ?
[qui,decouvrit,Europe,?]
Requete : [hum,wh(#0,and(astronomer(#0),discoveredby(Europa,#0)))]
Reponse :
>> Galilee

Quel astronome a decouvert un astre ?
[quel,astronome,a,decouvert,un,astre,?]
Requete : [hum,wh(#0,and(astronomer(#0),number(atleast,1,#1,and(body(#1),discoveredby(#1,#0)))))]
Reponse :
>> Bond
>> Cassini
>> Dollfus
>> Fountain
>> Galilee
>> Hall
>> Herschel
>> Huygens
>> Kuiper
>> Lacques
>> Lassel
>> Melotte
>> Nicholson
>> Perrine
>> Pickering
>> Smith
>> Tombaugh

Le satellite de quoi a decouvert Cassini ?
[le,satellite,de,quoi,a,decouvert,Cassini,?]
Requete : [bod,wh(#0,and(body(#0),the(sin,#1,satelliteof(#1,#0),discoveredby(#1,Cassini))))]
Reponse :
>> Aucun astre.

Le satellite de quel astre a decouvert Herschel ?
[le,satellite,de,quel,astre,a,decouvert,Herschel,?]
Requete : [bod,wh(#0,and(body(#0),the(sin,#1,satelliteof(#1,#0),discoveredby(#1,Herschel))))]
Reponse :
>> Aucun astre.

Autour de quoi est ce que tourne la terre ?
[autour,de,quoi,est,ce,que,tourne,la,terre,?]
Requete : [bod,wh(#0,and(body(#0),the(sin,#1,theearth(#1),satelliteof(#1,#0))))]
Reponse :
```

```
>> le soleil

Autour de quel astre tourne la lune ?
[autour,de,quel,astre,tourne,la,lune,?]
Requete : [bod,wh(#0,and(body(#0),the(sin,#1,themoon(#1),satelliteof(#1,#0))))]
Reponse :
>> la terre

Autour de quel astre un satellite tourne-t-il ?
[autour,de,quel,astre,un,satellite,tourne,il,?]
Requete : [bod,wh(#0,and(body(#0),number(atleast,1,#1,and(satellite(#1),satelliteof(#1,#0)))))]
Reponse :
>> la terre
>> Jupiter
>> Mars
>> Neptune
>> Pluton
>> Saturne
>> Uranus
>> le soleil

Quel est le diametre de Io ?
[quel,est,le,diametre,de,Io,?]
Requete : [mea,wh(#0,and(diameterof(#0,Io),true))]
Reponse :
>> 3638 km

Quels sont les satellites qui gravitent autour de Saturne ?
[quels,sont,les,satellites,qui,gravitent,autour,de,Saturne,?]
Requete : [bod,wh(#0,and(and(satellite(#0),satelliteof(#0,Saturn)),true))]
Reponse :
>> 1980S1
>> 1980S13
>> 1980S25
>> 1980S26
>> 1980S27
>> 1980S28
>> 1980S3
>> 1980S6
>> Dione
>> Encelade
>> Hyperion
>> Japet
>> Mimas
>> Phoebe
>> Rhea
>> Tethys
>> Titan

Quelle est la lune qui tourne autour de la terre ?
[quelle,est,la,lune,qui,tourne,autour,de,la,terre,?]
Requete : [bod,wh(#0,and(and(themoon(#0),the(sin,#1,theearth(#1),satelliteof(#0,#1))),true))]
Reponse :
>> la lune

Quel est l'astre dont le diametre est inferieur a 1000 km ?
[quel,est,l,astre,dont,le,diametre,est,inferieur,a,1000,km,?]
Requete : [bod,wh(#0,and(and(body(#0),the(sin,#1,diameterof(#1,#0),less(#1,1000))),true))]
```

```
Reponse :
>> Amalthee
>> Ananke
>> Ariel
>> Carme
>> Deimos
>> Elara
>> Encelade
>> Himalia
>> Hyperion
>> Leda
>> Lysithee
>> Mimas
>> Miranda
>> Nereide
>> Pasiphae
>> Phobos
>> Phoebe
>> Umbriel

Quel est l'astre dont le satellite gravite autour du soleil ?
[quel,est,l,astre,dont,le,satellite,gravite,autour,du,soleil,?]
[quel,est,l,astre,dont,le,satellite,gravite,autour]
Mot possible suivant :
autour          de

Quels sont les satellites de Jupiter ?
[quels,sont,les,satellites,de,Jupiter,?]
Requete : [bod,wh(#0,and(satelliteof(#0,Jupiter),true))]
Reponse :
>> 1979J2
>> 1979J3
>> Adrastee
>> Amalthee
>> Ananke
>> Callisto
>> Carme
>> Elara
>> Europe
>> Ganymede
>> Himalia
>> Io
>> Leda
>> Lysithee
>> Pasiphae
>> Sinope

Quel astre a decouvert Galilee ?
[quel,astre,a,decouvert,Galilee,?]
Requete : [bod,wh(#0,and(body(#0),discoveredby(#0,Galileo)))]
Reponse :
>> Callisto
>> Europe
>> Ganymede
>> Io

Qu'est-ce qui est inferieur a 100 km ?
[que,est,ce,qui,est,inferieur,a,100,km,?]
```

```
Requete : [mea,wh(#0,and(diameter(#0),less(#0,100)))]
Reponse :
>> 1 km
>> 10 km
>> 15 km
>> 20 km
>> 27 km
>> 30 km

Quel est le diametre de Leda ?
[quel,est,le,diametre,de,Leda,?]
Requete : [mea,wh(#0,and(diameterof(#0,Leda),true))]
Reponse :
>> 140 km


/* Examples  */

What's equal to the diameter of Saturn?
[what,is,equal,to,the,diameter,of,Saturn,?]
[what,is,equal]
Next possible word:
than

Whose satellite did Cassini discover?
[whose,satellite,did,Cassini,discover,?]
[whose,satellite,did,Cassini,discover]
Next possible word:
?

What's the diameter of the Earth?
[what,is,the,diameter,of,the,Earth,?]
Query: [mea,wh(#0,and(diameter(#0),the(sing,#1,theearth(#1),the(sin,#2,diameterof(#2,#1),equal(#0,#2)))))]
Answer:
>> I do not know.

What's the diameter of the Earth's satellite?
[what,is,the,diameter,of,the,Earth,s,satellite,?]
[what,is,the,diameter,of,the,Earth]
Next possible word:
<a body>        <number of km>  ?               a               an
around          at              diameters       exactly         five
four            moons           no              of              one
satellites      than            that            the             three
to              two             up              what            which
whose

What moons did Lassell discover?
[what,moons,did,Lassell,discover,?]
[what,moons,did]
Next possible word:
<a body>        <number of km>  a               an              around
astronomers     at              bodies          diameters       exactly
five            four            moons           no              of
one             planets         satellites      than            that
the             three           to              two             up
what            which           who             whom            whose
```

```
Who discovered Pluto?
[who,discovered,Pluto,?]
Query: [hum,wh(#0,and(astronomer(#0),discoveredby(Pluto,#0)))]
Answer:
>> Tombaugh

Who discovered a body?
[who,discovered,a,body,?]
Query: [hum,wh(#0,and(astronomer(#0),number(atleast,1,#1,and(body(#1),discoveredby(#1,#0)))))]
Answer:
>> Bond
>> Cassini
>> Dollfus
>> Fountain
>> Galileo
>> Hall
>> Herschel
>> Huygens
>> Kuiper
>> Lacques
>> Lassel
>> Melotte
>> Nicholson
>> Perrine
>> Pickering
>> Smith
>> Tombaugh

Who discovered a body which revolves around the satellites of the Sun?
[who,discovered,a,body,which,revolves,around,the,satellites,of,the,Sun,?]
Query: [hum,wh(#0,and(astronomer(#0),number(atleast,1,#1,and(and(body(#1),the(sin,#2,thesun(#2),the(plu,#3,satel
Answer:
>> Nobody.

The satellite of what body revolves around Saturn?
[the,satellite,of,what,body,revolves,around,Saturn,?]
Query: [bod,wh(#0,and(body(#0),the(sin,#1,satelliteof(#1,#0),satelliteof(#1,Saturn))))]
Answer:
>> No eaven body.

Who discovered a satellite of Saturn?
[who,discovered,a,satellite,of,Saturn,?]
Query: [hum,wh(#0,and(astronomer(#0),number(atleast,1,#1,and(satelliteof(#1,Saturn),discoveredby(#1,#0)))))]
Answer:
>> Bond
>> Cassini
>> Dollfus
>> Fountain
>> Herschel
>> Huygens
>> Lacques
>> Pickering
>> Smith

Who discovered Saturn's satellites?
[who,discovered,Saturn,s,satellites,?]
Query: [hum,wh(#0,and(astronomer(#0),exists(#1,and(satelliteof(#1,Saturn),discoveredby(#1,#0)))))]
```

```
Answer:
>> Bond
>> Cassini
>> Dollfus
>> Fountain
>> Herschel
>> Huygens
>> Lacques
>> Pickering
>> Smith

Who discovered the body's satellites?
[who,discovered,the,body,s,satellites,?]
[who,discovered,the,body]
Next possible word:
<a body>        <number of km>  ?               a               an
around          at              diameters       exactly         five
four            moons           no              of              one
satellites      than            that            the             three
to              two             up              what            which
whose

Who discovered Nereid?
[who,discovered,Nereid,?]
Query: [hum,wh(#0,and(astronomer(#0),discoveredby(Nereid,#0)))]
Answer:
>> Kuiper

What revolves around Uranus?
[what,revolves,around,Uranus,?]
Query: [bod,wh(#0,and(body(#0),satelliteof(#0,Uranus)))]
Answer:
>> Ariel
>> Miranda
>> Oberon
>> Titania
>> Umbriel

What's a satellite of Neptune?
[what,is,a,satellite,of,Neptune,?]
Query: [bod,wh(#0,and(body(#0),number(atleast,1,#1,and(satelliteof(#1,Neptune),equal(#0,#1)))))]
Answer:
>> Nereid
>> Triton

What's Triton ' s moon?
[what,is,Triton,s,moon,?]
Query: [bod,wh(#0,and(body(#0),exists(#1,and(moonof(#1,Triton),equal(#0,#1)))))]
Answer:
>> No eaven body.

What's a satellite of the Sun?
[what,is,a,satellite,of,the,Sun,?]
Query: [bod,wh(#0,and(body(#0),the(sin,#1,thesun(#1),number(atleast,1,#2,and(satelliteof(#2,#1),equal(#0,#2))))))]
Answer:
>> Earth
>> Jupiter
>> Mars
```

```
>> Mercury
>> Neptune
>> Pluto
>> Saturn
>> Uranus
>> Venus

What planet does Umbriel revolve around?
[what,planet,does,Umbriel,revolve,around,?]
[what,planet,does,Umbriel]
Next possible word:
discover        discovered      discovers       s

Which satellites of Jupiter did Gallileo Discover?
[which,satellites,of,Jupiter,did,Gallileo,Discover,?]
[]
Next possible word: Mot possible suivant :
<a body>        <entier naturel><nombre de km>  <number of km>  <un astronome>
a               an              around          at              au
aucun           aucune          autour          aux             cinq
combien         de              des             deux            diameters
diametre        diametres       du              exactement      exactly
five            four            l               la              le
les             lune            lunes           moons           no
of              one             quatre          que             quel
quelle          quelles         quels           qui             quoi
satellite       satellites      stop            stoppe          than
the             three           to              trois           two
un              une             up              what            who
whom            whose

Who are the astronomers?
[who,are,the,astronomers,?]
Query: [hum,wh(#0,and(astronomer(#0),the(plu,#1,astronomer(#1),equal(#0,#1))))]
Answer:
>> Nobody.

Who discoverd a moon which revolves around Saturn?
[who,discoverd,a,moon,which,revolves,around,Saturn,?]
[who]
Next possible word:
<a body>        <number of km>  a               an              are
around          astronomers     at              bodies          diameters
did             discover        discovered      discovers       do
does            exactly         five            four            is
moons           no              of              one             planets
satellites      than            that            the             three
to              two             up              what            which
who             whom            whose

What moon which revolves around Neptune did Pickering discover?
[what,moon,which,revolves,around,Neptune,did,Pickering,discover,?]
Query: [bod,wh(#0,and(and(moon(#0),satelliteof(#0,Neptune)),discoveredby(#0,Pickering)))]
Answer:
>> No eaven body.

What's the diameter of Deimos?
[what,is,the,diameter,of,Deimos,?]
```

```
Query: [mea,wh(#0,and(diameter(#0),the(sin,#1,diameterof(#1,Deimos),equal(#0,#1))))]
Answer:
>> 15 km

Around which planet does a moon which is Jupiter's satellite revolve?
[around,which,planet,does,a,moon,which,is,Jupiter,s,satellite,revolve,?]
[around]
Next possible word:
<a body>        <number of km>  a               an              at
diameters       exactly         five            four            moons
no              one             satellites      the             three
two             up              what            whom

Whose moons did Huygens discover?
[whose,moons,did,Huygens,discover,?]
[whose,moons,did,Huygens,discover]
Next possible word:
?

Who discovered an Earth?
[who,discovered,a,Earth,?]
[who,discovered,a]
Next possible word:
Moon            Sun             body            moon            planet
satellite

What's the diameter of Earth?
[what,is,the,diameter,of,Earth,?]
Query: [mea,wh(#0,and(diameter(#0),the(sin,#1,diameterof(#1,Earth),equal(#0,#1))))]
Answer:
>> 12756 km

Who discovered no satellite?
[who,discovered,no,satellite,?]
Query: [hum,wh(#0,and(astronomer(#0),neg(exists(#1,and(satellite(#1),discoveredby(#1,#0))))))]
Answer:
>> Nobody.

Who discovered two satellites?
[who,discovered,two,satellites,?]
Query: [hum,wh(#0,and(astronomer(#0),number(exactly,2,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Answer:
>> Hall
>> Kuiper

Who discovered at least two satellites?
[who,discovered,at,least,two,satellites,?]
Query: [hum,wh(#0,and(astronomer(#0),number(atleast,2,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Answer:
>> Cassini
>> Galileo
>> Hall
>> Herschel
>> Kuiper
>> Lassel
>> Nicholson

Who discovered exactly two satellites?
```

```
[who,discovered,exactly,two,satellites,?]
Query: [hum,wh(#0,and(astronomer(#0),number(exactly,2,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Answer:
>> Hall
>> Kuiper



|: Wh ?

[wh,?]
[]
Next possible word: Mot possible suivant :
<a body>        <entier naturel><nombre de km>  <number of km>  <un astronome>
a               an              around          at              au
aucun           aucune          autour          aux             cinq
combien         de              des             deux            diameters
diametre        diametres       du              exactement      exactly
five            four            l               la              le
les             lune            lunes           moons           no
of              one             quatre          que             quel
quelle          quelles         quels           qui             quoi
satellite       satellites      stop            stoppe          than
the             three           to              trois           two
un              une             up              what            who
whom            whose
|: who ?

[who,?]
[who]
Next possible word:
<a body>        <number of km>  a               an              are
around          astronomers     at              bodies          diameters
did             discover        discovered      discovers       do
does            exactly         five            four            is
moons           no              of              one             planets
satellites      than            that            the             three
to              two             up              what            which
who             whom            whose
|: discovered?

[who,discovered,?]
[who,discovered]
Next possible word:
<a body>        a               an              at              bodies
exactly         five            four            moons           no
one             planets         satellites      that            the
three           two             up              what            which
whose
|: two?

[who,discovered,two,?]
[who,discovered,two]
Next possible word:
bodies          moons           planets         satellites
|: satellites?
```

```
[who,discovered,two,satellites,?]
Query: [hum,wh(#0,and(astronomer(#0),number(exactly,2,#1,and(satellite(#1),discoveredby(#1,#0)))))]
Answer:
>> Hall
>> Kuiper
|: Stop!

[stop,!]
Bye.
```

12 Annexe : exemples pour tester les grammaires de l'anglais et du français

Les accolades { }, les commentaires Prolog et les lignes commençant par un caractère % ne sont pas pris en compte.

```
/*
GRAMMAIRE DU FRANCAIS

(1) Proposition principale
(2) Proposition interrogative et relative
(3) Groupe nominal
(4) Groupe nominal incomplet
(5) Article
(6) Pronom interrogatif
(7) Pronom relatif
(8) Cas et preposition
(9) Propositions moins un argument
(10) Propositions moins deux arguments
(11) Conjonction de qualicatifs
(12) Entrees lexicales
(13) Autre exemples

(1) Proposition principale
<phrase> -> stoppe ! | <prop> ?
*/

quel est l astre dont le satellite gravite autour de Mars ?

/*
(2) Proposition interrogative et relative
<prop> -> <art> <etre> <gn> | <pron rel> <etre> <gn> |
          <gn> <estceq> <prop 1> | <gn> <gn moins> <verbe compl> |
          <gn> <suj verbe> <gn moins>
<estceq> -> <vide| est ce <quei>
<quei> -> que | qui
*/

{quels sont les satellites qui gravitent autour de Saturne} ?
% {quels sont les satellites {qui sont des planetes}} ?
{qui {est ce {qui}} decouvrit Europe} ?
quel est l'astre {dont le diametre est inferieur a 1000 km} ?
{de quel astre Cassini a il decouvert un satellite} ?

/*
(3) Groupe nominal
<gn> -> <gn> | <pron rel> | <gn p>
<gn p -> <pron intero> | <artnombre> de <gn p| <cas> <nom 0> |
          <cas> <nom km 0> km | <art> <nom 1> <qual ev> |
          <art> <nom 2> <gn>
```

```
*/

{autour de quel astre} tourne {la lune} ?
{qui} decouvrit {au moins deux satellites {qui} tournent {autour de
  Saturne}} ?
{qui} decouvrit {au moins un satellite} ?
quel est {le nombre de satellites de Jupiter} ?
quelles sont les lunes qui tournent autour de la terre ?
quel nombre de satellites tournent autour de Saturne ?
{autour de combien d astres} gravite {} {Io} ?
{quels sont les astres} dont le diametre est superieur {a 100 km} ?
{autour de quoi} est ce que tourne {un satellite {de la terre}} ?

/*
(4) Groupe nominal incomplet
<gn moins> -> <art> <nom 2> <gn moins suite>
<gn moins suite> -> <vide> | <gn moins>
*/

quel est l'astre dont {le diametre {}} est superieur a
  10000 km ?
quel est l'astre dont {le diametre {du satellite{}}} est superieur a
  10000 km ?

/*
(5) Article
<art> -> <art p> | <cas> <art sc>
<art sc> -> <vide| combien de | <art sc p> | <exact> <art sc num>
<exact> -> au moins | au plus | exactement
<art p> -> au | aux | du | de | des
<art sc p> -> aucun | aucune | des | l | la | le | les | quel |
              quelle | quelles | quels
<art sc num> -> cinq | deux | quatre | trois | un | une | 0 | 1 | 2
*/

qu est ce que est superieur {au} diametre {du} Soleil ?
{autour de combien de} astres gravite Io ?
{a quel} diametre est inferieur {le} diametre de Venus ?
Autour de qui gravitent {exactement trois} satellites ?
{quel} diametre est inferieur {a 100} km ?

/*
(6) Pronom interrogatif
<pron intero> -> <pronom intero> | <prep> <pronom intero sc>
<pronom intero> -> que | qui | quoi
<pronom intero sc> -> qui | quoi
*/

{que} est ce que a decouvert Cassini ?
{autour de quoi} tourne Io ?

/*
(7) Pronom relatif
<pron rel> -> <pronom rel> | <prep<pronom rel sc>
<pronom rel> -> dont | que | qui
<pronom rel sc> -> qui | quoi
*/
```

```
quel est le astre {que} a decouvrit Cassini ?
quels sont les astres {autour de quoi} tourne Io ?

/*
(8) Cas et preposition
<cas> -> <vide| <prep>
<prep> -> <prep ? >| autour <prep>
<prep ? > -> a | de
*/

qui decouvrit {} Mars ?
que est ce qui tourne {autour {de}} Mars ?

/*
(9) Propositions moins un argument
<prop 1> -> <verbe compl> | <suj verbe> | <verbe suj>
<verbe compl> -> <prop 2> <gn>
<suj verbe> -> <gn> <prop 2>
<verbe suj> -> <prop 2> <gn>
*/

qui {decouvrit une planete} ?
a quoi {le diametre du Soleil est il egal} ?
a quoi {est egal le diametre de la lune} ?

/*
(10) Proposition moins deux arguments
<prop 2> -> <ne> <aux verbe> <til> <pp adj>
<pp adj> -> <vide| <ppad>
<aux verbe> -> <avoir> | <etre> | <verbe 2>
<ppad> -> <ppavoir 2> | <ppavoir obj 2> | <adjectif 2>
<ne> -> <vide> | ne
<til> -> <vide> | <pronom pers>
<etre> -> est | sont
<avoir> -> a | ont
<pronom pers> -> elle | elles | il | ils
*/

que {{tourne} {il}} autour de Saturne ?
quel planete est ce que {{a} {decouverte}} Hall ?
qui est ce qui {{a} {decouvert}} Deimos ?
Qu'est-ce qui {{est} {superieur}} 10 km ?
qui est-ce qui {{ne} {a} {decouvert}} aucun satellite de Jupiter ?
qui {{a} {il} {decouvert}} Ganymede ?

/*
(11) Conjonction de qualicatifs
<quals ev> -> <quals> | <rels ev>
<quals> -> <adjf> <et quals>
<et quals> -> <vide> | et <quals> | et <rels>
<rels ev> -> <vide> | <rels>
<rels> -> <prop> <et rels>
<et rels> -> <vide> | et <rels>
<adjf> -> <adjectif 2> <gn>
*/

quels sont les astres {dont le diametre est superieur a 100 km {et dont
  le diametre est inferieur a 1000 km}} ?
```

```
quels astres {dont le diametre est superieur a 100 km {et dont le
  diametre est inferieur a 1000 km}} gravitent autour de Jupiter ?
%quels astres dont le diametre est un diametre {superieur a 100 km {et
%inferieur a 1000 km}} gravitent autour d'un astre ?

/*
(12) Entrees lexicales
<nom 0? > -> 0 | 1 | 2 | 3
<nom 0> -> <astronome> | <astre>
<astronome> -> Galilee  | <uk/fr astronomer>
<astre> ->
   Adrastee | Amalthee | Encelades | Europe | Japet | Lysithee |
   Mercure | Nereide | Pluton | Saturne | la lune | la terre|
   le soleil | <uk/fr star>
<nom 1> -> astre | astres | astronome | astronomes | lune | planete |
          planetes | soleil | terre
<nom 2> -> diametre | diametres | satellite | satellites
<verbe 2> -> decouvrit | geravite | gravitent | tourne | tournent
<pp avoir 2> -> decouvert
<pp avoir obj 2> -> | decouverte | decouvertes | decouverts
<adjectif 2> ->
   egal | egale | egales | egaux | inferieure | inferieures |
   inferieurs | superieur | superieure | superieures | superieurs
*/

% que est ce qui est {superieur} a {800} km ?
quels {astronomes} ont {decouvert} {Hyperion} ?
% quel est le {diametre} de la {lune} ?
autour de quoi {tournent} les {planetes} ?
quelles {planetes} a {decouvertes} Hall ?
que est ce que {Hall} a {decouvert} ?
% quelle {lune} est un {satellite} de la {terre} ?
qui a {decouvert} {Leda} ?

/*
ENGLISH GRAMMAR

(1) Main clause
(2) Interrogative and relative clause
(3) Argument
(4) Noun phrase
(5) Right possesive adjunction
(6) Interogative pronouns
(7) Relative pronouns
(8) Determiner
(9) Clauses minus one argument
(10) Conjunction of relative clauses
(11) Lexical entries
(11) Other examples

(1) Main clause
<english sentence> -> stop ! | <clause> ?
*/

Who discovered at most two satellites?

/*
(2) Interrogative and relative clause
```

```
<clause> -> <arg> <verb compl> | <arg> <do> <suj verb prep>
<do> -> did | do | does
*/

{who discovered one satellite}?
{around what does a satellite revolve}?
{who discovered a satellite which revolves around Mars}?
{what satellite around which Leda revolves did Hall discover}?

/*
(3) Argument
<arg> -> <case> <np> | <whose> <noun 2>
<case> -> <empty> | <prep>
<prep> -> around | of | than | to
*/

{around what} planet does Deimos revolve?
{whose diameter} is less than 100 km?
{} what moon {whose diameter is greater {than} 100 km did {} Hall
  discoverd?

/*
(4) Noun phrase
<np> -> <pron> | <np p>
<np p> -> <noun 0> <radj> | <number of km> km | <det> <noun 1> <radj> |
          <det> <noun 1> <rels> | <det> <noun 2> <arg>
*/

{who} discovered exactly one satellite?
what is the diameter of Saturn?
what is less than 1000 km?
what is the diameter of a planet?
what is a body whose diameter is less than 1000 km?
what body diameter s is less to 100 km?

/*
(5) Right possesive adjunction
<radj> -> <empty> | ' s <radj>
*/

who discovered Mars {s satellite}?
what is Mars {s satellite s diameter}?

/*
(6) Interogative  pronouns
<pron> -> that | what | which | who | whom
*/

what is the moon of Saturn?

/*
(7) Relative pronouns
*/



{who} discovered a body which revolves around Saturn?
```

```
/*
(8) Determiner
<det> -> <empty> | <a> | <the> | <what>
*/

who discovers {} planets?
who discovers {a} planet?
Who discovered {three} satellite?
who discovered {at most one} satellite?

/*
(9) Clauses minus one argument
<verb compl> -> <verb 2> <arg> | <be> <arg> | <be> <adjv>
<suj verb prep> -> <arg> <verb 2> <prep dangle>
<adjv> -> <adjecvtive 2> <arg>
<prep dangle> -> <empty> | <prep
*/

who {discovered a body} which {revolves around the satellites of Saturn}?
what {are bodies whose diameters {are less than 1000 km}}?
around what {a satellite turns}?
around what {a satellite turns around}?

/*
(10) Conjunction of relative clauses
<rels> -> <clause> <and rels>
<and rels> -> <empty> | and <rels
*/

what is the planet {whose diameter exceeds the diameter of Mars {and
  whose diameter is less than the diameter of the Earth{}}}?
what is the moon {which revolves around Neptune {and which Lassel
  discovered {and whose diameter is 5000 km{}}}}?

/*
(11) Lexical entries
<number> -> 0 | 1 | 2 |
<number of km> ->
   15 | 20 | 27 | 30| 140 | 200 | 270 | 300 | 400 | 500 | 700 |
   900 | 1100 | 1200 | 1400| 1500 | 1600 | 1700 | 3000 | 3126 |
   3473 | 3638 | 4848 | 5000 | 5150 | 5276 | 6787 | 12100 |
   12756 | 49500 | 51800 | 120600 | 142800| 1392000 |
   1 | 10 | 100 | 1000 | 10000 | 100000 | 1000000 | 10000000 |
   100000000
<noun 0> -> <astronomer| <celestial body>
<astronomer> -> Galileo | <uk/fr astronommer>
<star> ->
   Adrastea | Amalthea | Enceladus | Europa |
   Iapetus | Lysithea | Mercury | Nereid | Pluto | Saturn |
   the Earth | the Moon | the Sun | <uk/fr star>
<noun 1> ->
   astronomer | astronomers | bodies | body | diameter | diameters |
   Earth | Moon | moon | moons | planet | planets | satellite |
   satellites | Sun
<noun 2> -> diameter | diameters | moon | moons | satellite | satellites
<verb 2> ->
   discover | discovers | discovered | exceed | exceeds | revolve |
   revolves
```

```
<adjective 2> -> equal | greater | less
<uk/fr astronomer> ->
   Bond | Cassini | Dollfus | Fountain |
   Hall | Herschel | Huygens | Kuiper | Lacques |
   Lassel | Melotte | Nicholson | Perrine | Pickering
   Smith | Tombaugh
<uk/fr star>
   1979J2 | 1979J3 | 1980S1 | 1980S13 | 1980S25 | 1980S26
   1980S27 | 980S28 | 1980S3 | 1980S6 |
   Ananke | Ariel | Callisto | Carme | Charon | Deimos | Dione
   Elara |  Ganymede | Himalia | Hyperion
   Io | Jupiter | Leda |  Mars
   Mimas | Miranda | Neptune | Oberon | Pasiphae
   Phobos | Phoebe | Rhea | Sinope | Tethys
   Titan | Titania | Triton | Umbriel | Uranus | Venus
*/

what are the {bodies} whose {diameters} are {less} than {1000 km}?
what did {Kuiper} {discover}?

stop!
```