

```
tens(40) --> [forty].
tens(50) --> [fifty].
tens(60) --> [sixty].
tens(70) --> [seventy].
tens(80) --> [eighty].
tens(90) --> [ninety].
```

The associated program now responds to queries of the following type:

```
[user] ?- number(Value, [ two, hundred, and, twenty, three ], []).
Value = 223
yes
```

Besides performing this syntactical analysis and calculating the value, the program is also capable of providing an appropriate verbal expression for a given value (synthesis).

```
[user] ?- number(101, X, []).
X = [one,hundred,and,one]
```

CHAT-80

Can a computer understand a human? Lets test that. Let human ask questions about a domain and see if the machine can understand those questions w.r.t. the background knowledge of that domain?

It turns out this is a three-part problem:

1. Representation: of the background knowledge
2. Parsing: turning some English text into a Prolog clause
3. Planning: re-ordering the clauses

Some Parsing Tricks

```
a, some, the[singular]      exists(X,R & S)
no                          not exists(X,R & S)
every, all                  not exists(X,R & not S)
the[plural]                 exists(X, setof(X,R,X) & S)
one, two, ... numeral(N)    numberof(X,R & S,N)
which, what                 answer(X) <= R & S
how many                    answer(N) <= numberof(X,R & S,N)
```

Some birds migrate.

```
exists(X,bird(X) & migrates(X)).
```

The population of Britain exceeds 50 million.

```
exists(X,population(britain,X) & X > 50000000).
```

There are no rivers in Antarctica.

```
not exists(X,river(X) & in(X,antarctica)).
```

Man inhabits every continent.

```
not exists(X,continent(X) & not inhabits(man,X)).
```

Jupiter is the largest of the planets.

```
exists(X,setof(X,planet(X),X) & largest(X,jupiter)).
```

The Rhine flows through three countries.

```
numberof(X,country(X) & flows through(rhine,X),3).
```

Which birds migrate?

```
answer(X) <= bird(X) & migrates(X).
```

How many countries export oil?

```
answer(N) <= numberof(X,country(X) & exports(X,oil),N).
```

Which European country exports no arms to countries in Africa?

```
answer(C) <= european(C) & country(C) &
not exists(X, arm(X) &
exists(C1, country(C1) & in(C1,africa) &
exports(C,X,C1) ) )
```

"For any C, C is an answer if C is European and C is a country and it cannot be shown that there is some X such that X is an armament and there is some C1 such that C1 is a country and C1 is in Africa and C exports X to C1"

Planning

Move common tests out of inner loop. Peek at the size of each DB and match on smaller things before longer things.

```
answer(C) <= country(C) &
borders(C,mediterranean) &
exists(C1,country(C1) & asian(C1) &
borders(C,C1))
```

After planning, the logical form is transformed into:

```
answer(C) <= borders(C,mediterranean) &
{country(C)} & {borders(C,C1) &
{asian(C1) & {country(C1)}}}
```

Representation

Countries

```
...
country(uruguay,south_america,-32,55,68548,2990000,montevideo,peso).
country(venezuela,south_america,8,65,352143,11520000,caracas,bolivar).
country(vietnam,southeast_east,17,-107,126436,41850000,hanoi,dong).
country(west_germany,western_europe,52,-9,95815,61970000,bonn,deutsche_mark).
country(western_samoa,australasia,-14,172,1133,150000,apia,tala).
country(yemen,middle_east,15,-44,75289,1600000,sana_rial).
country(yugoslavia,southern_europe,44,-20,98766,21126000,belgrade,dinar).
```

```
country(zaire,central_africa,-3,-23,905063,23560000,kinshasa,zaire).
country(zambia,southern_africa,-15,-28,290724,4640000,lusaka,kwacha).
country(zimbabwe,southern_africa,-20,-30,150333,5690000,salisbury,rhodesian_dollar).
```

Rivers:

```
...
river(senegal_river,[atlantic,senegal,mali,guinea]).
river(tagus,[atlantic,portugal,spain]).
river(vistula,[baltic,poland]).
river(volga,[black_sea,soviet_union]).
river(volta,[atlantic,ghana,upper_volta]).
river(yangtze,[pacific,china]).
river(yenisei,[arctic_ocean,soviet_union,mongolia]).
river(yukon,[pacific,united_states,canada]).
river(zambesi,[indian_ocean,mozambique,zambia,angola]).
```

World:

```
flows(R,C1,C2) :- river(R,L), links(L,C2,C1).

first([X|_],X).

last([X],X).
last(_|L,X) :- last(L,X).

links([X1,X2|_],X1,X2).
links(_|L,X1,X2) :- links(L,X1,X2).
```

Borders:

```
borders(persian_gulf,saudi_arabia).
borders(persian_gulf,united_arab_emirates).
borders(red_sea,israel).
borders(red_sea,jordan).
borders(red_sea,saudi_arabia).
borders(red_sea,yemen).
borders(red_sea,egypt).
borders(red_sea,ethiopia).
borders(red_sea,sudan).
```

Contains

```
contains0(west_germany,hamburg).
contains0(west_germany,rhine).
contains0(yugoslavia,danube).
contains0(zaire,congo_river).
contains0(zambia,congo_river).
contains0(zambia,zambesi).
```

And away we go...

what rivers are there ?

```
Parse: 0.0168457sec.
whq
  $VAR
  1
  s
  np
  3+plu
  np_head
  int_det(B)
  []
  river
  []
  verb(be,active,pres+fin,[],pos)
  void
  []

Semantics: 0.0170898sec.
answer([B]) :-
  river(B)
  & exists B
  true

Planning: 0.0sec.
answer([B]) :-
  river(B)
  & exists B
  true
amazon, amu_darya, amur, brahmaputra, colorado, congo_river, cubango, danube, don, elbe, euphrates, ganges, hwang_ho, indus, irrawaddy, lena, limp

Reply: 0.166992sec.
```

what countries are there in europe ?

```
Parse: 0.0500488sec.
whq
  $VAR
  1
  s
  np
  3+plu
  np_head
  int_det(B)
  []
  country
  []
  verb(be,active,pres+fin,[],pos)
  void
  pp
  prep(in)
  np
  3+sin
  name(europe)
  []

Semantics: 0.032959sec.
answer([B]) :-
  country(B)
  & in(B,europe)

Planning: 0.0sec.
answer([B]) :-
  in(B,europe)
  & {country(B)}
```

albania, andorra, austria, belgium, bulgaria, cyprus, czechoslovakia, denmark, east_germany, eire, finland, france, greece, hungary, iceland, ital;
Reply: 0.199951sec.

which is the largest african country ?

```
Parse: 0.0500488sec.  
whq  
  $VAR  
  1  
  s  
  np  
    3+sin  
    wh(B)  
    []  
  verb(be,active,pres+fin,[],pos)  
  arg  
  dir  
  np  
    3+sin  
    np_head  
    det(the(sin))  
    sup  
    most  
    adj  
    large  
  adj  
  african  
  country  
  []
```

```
Semantics: 0.0339356sec.  
answer([B]) :-  
  exists C  
    C = setof D:E  
      country(E)  
      & area(E,D)  
      & african(E)  
      & aggregate(max,C,B)
```

```
Planning: 0.0500488sec.  
answer([B]) :-  
  exists C  
    C = setof D:E  
      african(E)  
      & {country(E)}  
      & area(E,D)  
      & aggregate(max,C,B)  
sudan.
```

Reply: 0.300049sec.

what is the ocean that borders african countries and that borders asian countries ?

```
Parse: 0.0827637sec.  
whq  
  $VAR  
  1  
  s  
  np  
    3+sin  
    wh(B)  
    []  
  verb(be,active,pres+fin,[],pos)  
  arg  
  dir  
  np  
    3+sin  
    np_head  
    det(the(sin))  
    []  
  conj  
  and  
  rel  
  $VAR  
  2  
  s  
  np  
    3+sin  
    wh(C)  
    []  
  verb(border,active,pres+fin,[],pos)  
  arg  
  dir  
  np  
    3+plu  
    np_head  
    generic  
    adj  
    african  
    country  
  []  
  rel  
  $VAR  
  3  
  s  
  np  
    3+sin  
    wh(D)  
    []  
  verb(border,active,pres+fin,[],pos)  
  arg  
  dir  
  np  
    3+plu  
    np_head  
    generic  
    adj  
    asian  
    country  
  []  
  []
```

```

Semantics: 0.100098sec.
answer([B]) :-
  ocean(B)
  & exists C
    country(C)
    & african(C)
    & borders(B,C)
  & exists D
    country(D)
    & asian(D)
    & borders(B,D)

Planning: 0.0500488sec.
answer([B]) :-
  exists C D
    ocean(B)
    & { borders(B,C)
      & {african(C)}
      & {country(C)} }
    & { borders(B,D)
      & {asian(D)}
      & {country(D)} }
  indian_ocean.

Reply: 0.25sec.

```

how many countries does the danube flow through ?

```

Parse: 0.065918sec.
whq
  $VAR
  1
  s
  np
    3+sin
    name(danube)
    []
  verb(flow,active,pres+fin,[],pos)
  []
  pp
    prep(through)
    np
      3+plu
      np_head
      quant(same,wh(B))
      []
      country
      []

```

```

Semantics: 0.0168457sec.
answer([B]) :-
  B = numberof C
  country(C)
  & flows(danube,C)

Planning: 0.0158691sec.
answer([B]) :-
  B = numberof C
  flows(danube,C)
  & {country(C)}
  6.

Reply: 0.032959sec.

```

what is the total area of countries south of the equator and not in australasia ?

```

Parse: 0.0500488sec.
whq
  $VAR
  1
  s
  np
    3+sin
    wh(B)
    []
  verb(be,active,pres+fin,[],pos)
  arg
  dir
  np
    3+sin
    np_head
    det(the(sin))
    adj
    total
    area
  pp
    prep(of)
    np
      3+plu
      np_head
      generic
      []
      country
    conj
    and
    reduced_rel
    $VAR
    2
    s
    np
      3+plu
      wh(C)
      []
      verb(be,active,pres+fin,[],pos)
      arg
      pred
      pp
      prep(southof)
      np
      3+sin
      name(equator)
      []
    []
  reduced_rel

```

```

$VAR
3
s
  np
    3+plu
    wh(D)
    []
    verb(be,active,pres+fin,[],neg)
    arg
      pred
      pp
        prep(in)
        np
          3+sin
          name(australasia)
          []
    []
  []

```

Semantics: 0.132813sec.

```

answer([B]) :-
  exists C
    C = setof D:[E]
      area(E,D)
      & country(E)
      & southof(E,equator)
      & \+in(E,australasia)
      & aggregate(total,C,B)

```

Planning: 0.0830078sec.

```

answer([B]) :-
  exists C
    C = setof D:[E]
      southof(E,equator)
      & area(E,D)
      & {country(E)}
      & {\+in(E,australasia)}
      & aggregate(total,C,B)
10228 ksqmiles.

```

Reply: 0.25sec.

is there some ocean that does not border any country ?

Parse: 0.0500488sec.

```

q
s
  there
  verb(be,active,pres+fin,[],pos)
  arg
    dir
    np
      3+sin
      np_head
      det(some)
      []
      ocean
    rel
    $VAR
    1
    s
      np
        3+sin
        wh(B)
        []
        verb(border,active,pres+fin,[],neg)
        arg
          dir
          np
            3+sin
            np_head
            det(any)
            []
            country
          []
        []
      []
    []
  []

```

Semantics: 0.032959sec.

```

answer([]) :-
  exists B
    ocean(B)
  & \+
    exists C
      country(C)
      & borders(B,C)

```

Planning: 0.032959sec.

```

answer([]) :-
  exists B
    { ocean(B)
      & { \+
          exists C
            borders(B,C)
            & {country(C)} } }

```

Yes.

Reply: 0.0158691sec.

what are the continents no country in which contains more than two cities whose population exceeds 1 million ?

Parse: 0.184082sec.

```

whq
$VAR
1
s
  np
    3+plu
    wh(B)
    []
    verb(be,active,pres+fin,[],pos)
    arg
      dir
      np
        3+plu
        np_head

```

```

det(the(plu))
[]
continent
rel
$VAR
2
s
np
3+sin
np_head
det(no)
[]
country
pp
prep(in)
np
3+plu
wh(C)
[]
verb(contain,active,pres+fin,[],pos)
arg
dir
np
3+plu
np_head
quant(more,nb(2))
[]
city
rel
$VAR
3
s
np
3+sin
np_head
det(the(sin))
[]
population
pp
poss
np
3+plu
wh(D)
[]
verb(exceed,active,pres+fin,[],pos)
arg
dir
np
3+sin
np_head
quant(same,nb(1))
[]
million
[]
[]
[]
[]
[]

```

Semantics: 0.0998535sec.

```

answer([B]) :-
  B = setof C
    continent(C)
  & \+
    exists D
      country(D)
      & in(D,C)
      & exists E
        E = numberof F
        city(F)
        & exists G
          population(F,G)
          & exceeds(G,1--million)
          & in(F,D)
          & E>2

```

Planning: 0.0500488sec.

```

answer([B]) :-
  B = setof C
    continent(C)
  & \+
    exists D
      country(D)
      & in(D,C)
      & exists E
        E = numberof F
        city(F)
        & exists G
          population(F,G)
          & exceeds(G,1--million)
          & in(F,D)
          & E>2

```

[africa,antarctica,australasia].

Reply: 25.0sec.

which country bordering the mediterranean borders a country that is bordered by a country whose population exceeds the population of india ?

Parse: 0.116943sec.

```

whq
$VAR
1
s
np
3+sin
np_head
int_det(B)
[]
country
reduced_rel
$VAR
2
s
np
3+sin
wh(C)
[]
verb(border,active,inf,[prog],pos)

```

```

    arg
    dir
    np
    3+sin
    name(mediterranean)
    []
verb(border,active,pres+fin,[],pos)
arg
dir
np
3+sin
np_head
det(a)
[]
country
rel
$VAR
3
s
np
3+sin
wh(D)
[]
verb(border,passive,pres+fin,[],pos)
[]
pp
prep(by)
np
3+sin
np_head
det(a)
[]
country
rel
$VAR
4
s
np
3+sin
np_head
det(the(sin))
[]
population
pp
poss
np
3+sin
wh(E)
[]
verb(exceed,active,pres+fin,[],pos)
arg
dir
np
3+sin
np_head
det(the(sin))
[]
population
pp
prep(of)
np
3+sin
name(india)
[]
[]
[]

```

Semantics: 0.116943sec.

```

answer([B]) :-
country(B)
& borders(B,mediterranean)
& exists C
country(C)
& exists D
country(D)
& exists E
population(D,E)
& exists F
population(india,F)
& exceeds(E,F)
& borders(D,C)
& borders(B,C)

```

Planning: 0.0830078sec.

```

answer([B]) :-
exists C D E F
population(india,F)
& borders(B,mediterranean)
& {country(B)}
& { borders(B,C) }
& {country(C)}
& { borders(D,C) }
& {country(D)}
& { population(D,E) }
& {exceeds(E,F)} } }

```

turkey.

Reply: 1.09985sec.

which countries have a population exceeding 10 million ?

Parse: 0.0500488sec.

```

whq
$VAR
1
s
np
3+plu
np_head
int_det(B)
[]
country
[]
verb(have,active,pres+fin,[],pos)
arg
dir
np
3+sin

```

```

np_head
det(a)
[]
population
reduced_rel
$VAR
2
s
np
3+sin
wh(C)
[]
verb(exceed,active,inf,[prog],pos)
arg
dir
np
3+plu
np_head
quant(same,nb(10))
[]
million
[]
[]

```

Semantics: 0.0671387sec.

```

answer([B]) :-
  country(B)
  & exists C
    exceeds(C,10--million)
    & population(B,C)

```

Planning: 0.0158691sec.

```

answer([B]) :-
  exists C
    country(B)
    & { population(B,C)
      & {exceeds(C,10--million)} }

```

afghanistan, algeria, argentina, australia, bangladesh, brazil, burma, canada, china, colombia, czechoslovakia, east_germany, egypt, ethiopia, fra

Reply: 0.466064sec.

