EXPERIMENT NO: 8
DATE: 19/04/2021

A N V SREEVISHNU
RA1811003010333

Implementation of Knowledge Representation Schemes

Aim: To implement knowledge representation scheme in SWI prolog.

Procedure/Algorithm:

- Retrieve a value V for an attribute A of an instance object O.
- Find object O in the knowledge base.
- If there is a value for the attribute A then report that value.
- Else, if there is a value for the attribute instance; If not, then fail.
- Else, move to the node corresponding to that value and look for a value for the attribute A; If one is found, report it.
- Else, do until there is no value for the "isa" attribute or
- until an answer is found :
- Get the value of the "isa" attribute and move to that node.
- See if there is a value for the attribute A; If yes, report it.

Code:

```
/* animal.pl

animal identification game.

   start with ?- go.     */

go :- hypothesize(Animal),
      write('I guess that the animal is: '),
      write(Animal),
      nl,
      undo.
```

```
/* hypotheses to be tested */
hypothesize(cheetah)   :- cheetah, !.
hypothesize(tiger)     :- tiger, !.
hypothesize(giraffe)   :- giraffe, !.
hypothesize(zebra)     :- zebra, !.
hypothesize(ostrich)   :- ostrich, !.
hypothesize(penguin)   :- penguin, !.
hypothesize(albatross) :- albatross, !.
hypothesize(unknown).             /* no diagnosis */

/* animal identification rules */
cheetah :- mammal,
           carnivore,
           verify(has_tawny_color),
           verify(has_dark_spots).
tiger :- mammal,
         carnivore,
         verify(has_tawny_color),
         verify(has_black_stripes).
giraffe :- ungulate,
           verify(has_long_neck),
           verify(has_long_legs).
zebra :- ungulate,
         verify(has_black_stripes).

ostrich :- bird,
           verify(does_not_fly),
           verify(has_long_neck).
penguin :- bird,
           verify(does_not_fly),
           verify(swims),
           verify(is_black_and_white).
albatross :- bird,
            verify(appears_in_story_Ancient_Mariner),
            verify(flys_well).

/* classification rules */
mammal    :- verify(has_hair), !.
mammal    :- verify(gives_milk).
bird      :- verify(has_feathers), !.
bird      :- verify(flys),
            verify(lays_eggs).
carnivore :- verify(eats_meat), !.
carnivore :- verify(has_pointed_teeth),
            verify(has_claws),
            verify(has_forward_eyes).
ungulate :- mammal,
           verify(has_hooves), !.
ungulate :- mammal,
           verify(chews_cud).

/* how to ask questions */
ask(Question) :-
```

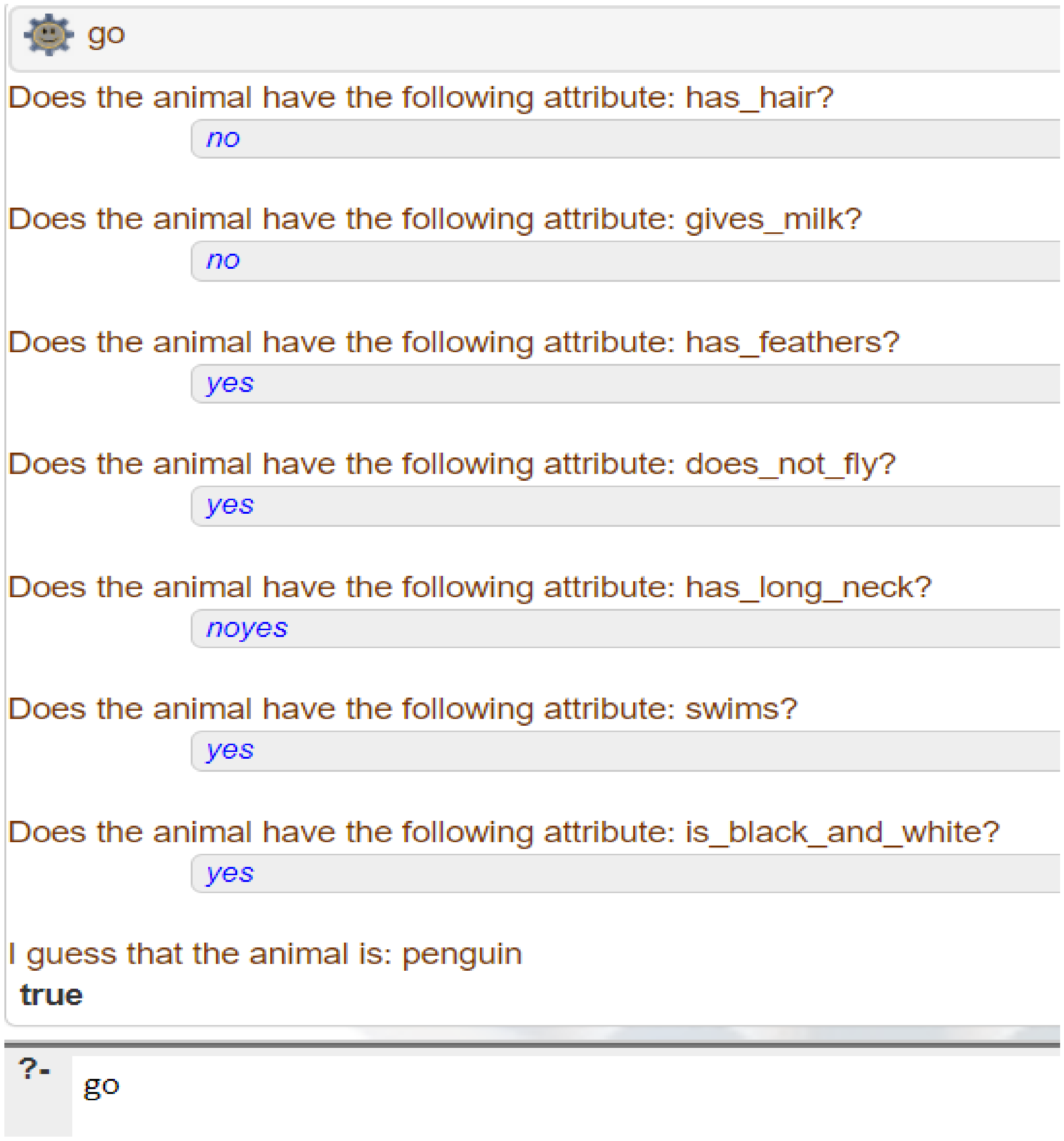
```
    write('Does the animal have the following attribute: '),
    write(Question),
    write('? '),
    read(Response),
    nl,
    ( (Response == yes ; Response == y)
      ->
       assert(yes(Question)) ;
       assert(no(Question)), fail).

:- dynamic yes/1,no/1.

/* How to verify something */
verify(S) :-
   (yes(S)
    ->
    true ;
    (no(S)
     ->
     fail ;
     ask(S))).

/* undo all yes/no assertions */
undo :- retract(yes(_)),fail.
undo :- retract(no(_)),fail.
undo.
```

Output:



```
go
Does the animal have the following attribute: has_hair?
        no
Does the animal have the following attribute: gives_milk?
        no
Does the animal have the following attribute: has_feathers?
        yes
Does the animal have the following attribute: does_not_fly?
        yes
Does the animal have the following attribute: has_long_neck?
        noyes
Does the animal have the following attribute: swims?
        yes
Does the animal have the following attribute: is_black_and_white?
        yes
I guess that the animal is: penguin
 true
?-  go
```

Result: Thus, the implementation of knowledge representation schemes in SWI prolog is successfully completed.