**Artificial Intelligence Lab**

**LAB 8 – Implementation of knowledge representation schemes – use cases**

**Aditya Kumar**

**RA1911030010100**

**O2 Section**

**Date – 1/4/22**

**Problem Statement :**

Implementation of knowledge representation scheme.

Algorithm :

1. Create a knowledge base with identification rules
2. Create a questions and answers knowledge base
3. Ask questions to user
4. Compare inputs to the database
5. Use the inputs and compare to identification rules
6. If an animal is found print the guess.

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))).

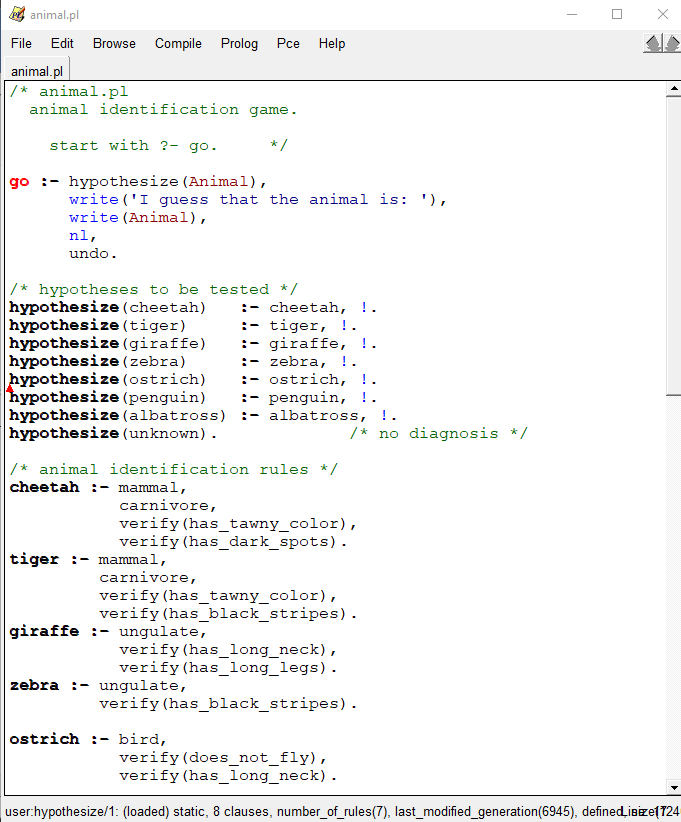
/\* undo all yes/no assertions \*/

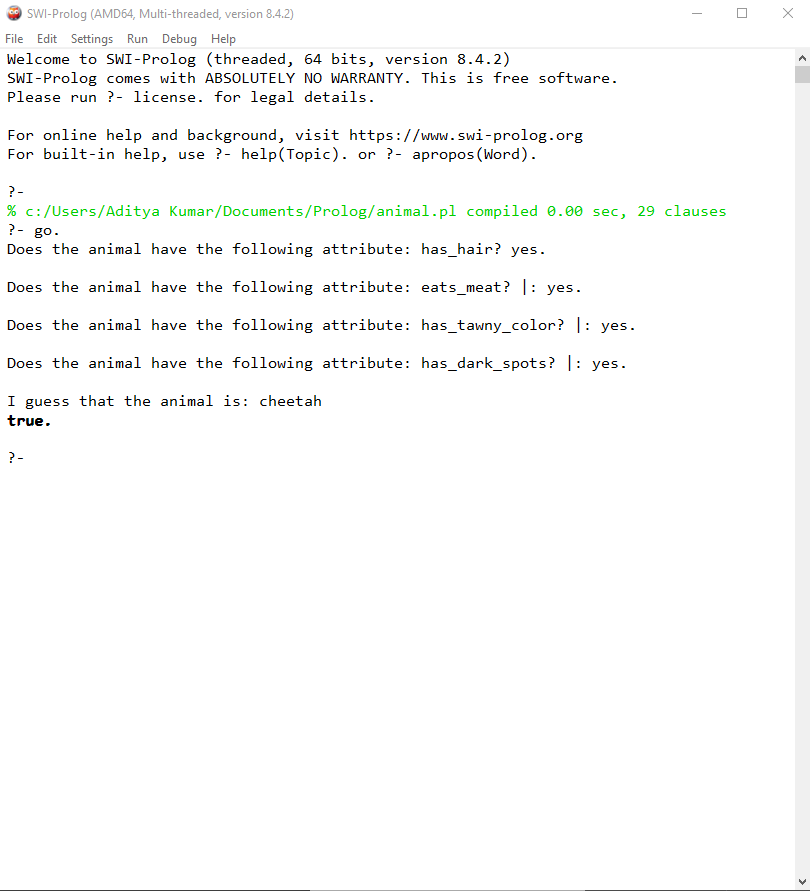
undo :- retract(yes(\_)),fail.

undo :- retract(no(\_)),fail.

undo.

Output:





Result :

Hence we created a knowledge base and tested it.