Batch 1
PA 17 Ketaki Patil

AI Lab Assignment 5

Title : Write a program to develop mini expert system using Prolog.

Aim : Implementation of expert system

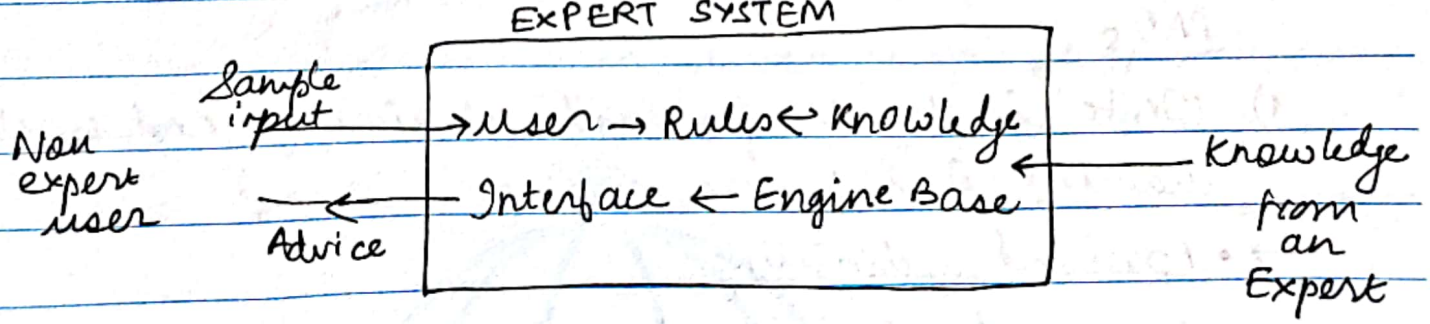
Requirements : SWI Prolog, Turbo Prolog

Objective : To study the concepts of expert system and inference engine.

Theory : Write in brief

1) **Architecture of Expert System**

An expert system is a computer program that is designed to solve complex problems and to provide decision making ability like a human expert. It performs this by extracting knowledge base using the reasoning and inference rules according to the user queries.



2) Main players of expert system

There are 5 members of the expert system.

(i) Domain Expert - He is a knowledgeable and skilled person capable of solving problems in a specific area of domain.

(ii) Knowledge engineer - Someone who is capable of designing, building and testing an expert system.

(iii) Programmer - He is the person responsible for the actual programming, describing the domain knowledge in terms that a computer can understand.

(iv) Project Manager - He is the leader of the expert system development team, responsible for keeping the project on track.

(v) End User - Often called the user.

Input : Run the program on SWI Prolog.

Output : Give decisions based on the rules provided in the program.

Platform : Linux / Windows

FAQs

1) Write in brief forward chaining and backward chaining of inference engine.

→ • Forward chaining

Forward chaining is also known as a forward deduction or forward reasoning method when using

an inference engine. It is a form of reasoning which start with atomic sentences in the knowledge base and applies inference rules in the forward direction to extract more data until a goal reached. The forward chaining algorithm starts from known facts, triggers all rules whose premises are satisfied, and add their conclusion to the known facts. This process repeats until the problem is solved.

Properties :

(i) It is a down up approach

(ii) It is also called as data driven as we reach to the goal using available data.

(iii) Commonly used in an expert systems, such as CLIPS, business etc.

- Backward Chaining :

It is also known as a backward deduction or backward reasoning method when using an inference engine. A backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through the rules to find known facts that support the goal.

Properties :

(i) It is also known as top down approach.

(ii) It is called as goal driven approach, as a list of goals decides which rules are selected and used.

(iii) The backward chaining method mostly uses a depth first search strategy.

2) List down the applications of expert system.

→ (i) Different types of medical diagnosis like internal medicine, blood diseases etc.

(ii) Forecasting crop damage.

(iii) Diagnosis of a software development project.

(iv) Teaching students specialize task.

AI LAB ASSIGNMENT NO.5
MEDICAL EXPERT SYSTEM USING PROLOG

NAME : KETAKI PATIL
ROLL NO : PA-17
BATCH : A1

CODE :

```
/*
Disease Diagnosis expert system
AI LAB Assignment 5
Name: Ketaki Patil
Roll No. PA 17
Batch 1
*/

go:-
hypothesis(Disease),
write('The patient might be having '),
write(Disease),
nl,
undo.

/*Hypothesis that should be tested*/
hypothesis(cold) :- cold, !.
hypothesis(flu) :- flu, !.
hypothesis(typhoid) :- typhoid, !.
hypothesis(measles) :- measles, !.
hypothesis(german_measles) :- german_measles, !.
hypothesis(malaria) :- malaria, !.
hypothesis(mumps) :- mumps, !.
hypothesis(chicken_pox) :- chicken_pox, !.
hypothesis(tuberculosis) :- tuberculosis, !.
hypothesis(pneumonia) :- pneumonia, !.
hypothesis(byssinosis) :- byssinosis, !.
hypothesis(pertusis) :- pertusis, !.
hypothesis(pneumoconiosis) :- pneumoconiosis, !.
hypothesis(sarcoidosis) :- sarcoidosis, !.
hypothesis(asbestosis) :- asbestosis, !.
hypothesis(asthma) :- asthma, !.
hypothesis(bronchiolitis) :- bronchiolitis, !.
hypothesis(influenza) :- influenza, !.
hypothesis(lung_cancer) :- lung_cancer, !.
```

```
hypothesis(covid) :- covid, !.
hypothesis(diabetes) :- diabetes, !.
hypothesis(epilepsy) :- epilepsy, !.
hypothesis(glaucoma) :- glaucoma, !.
hypothesis(heart_stroke) :- heart_stroke, !.
hypothesis(hyperthyroidism) :- hyperthyroidism, !.
hypothesis(hypothermia) :- hypothermia, !.
hypothesis(jaundice) :- jaundice, !.
hypothesis(sinusitis) :- sinusitis, !.
hypothesis(unknown). /* no diagnosis*/
/*Hypothesis Identification Rules*/

cold :-
verify(headache),
verify(runny_nose),
verify(sneezing),
verify(sore_throat),
nl.

flu :-
verify(fever),
verify(headache),
verify(chills),
verify(body_ache),
nl.

typhoid :-
verify(headache),
verify(abdominal_pain),
verify(poor_appetite),
verify(fever),
nl.

measles :-
verify(fever),
verify(runny_nose),
verify(rash),
verify(conjunctivitis),
nl.

german_measles :-
verify(fever),
verify(runny_nose),
verify(rash),
verify(headhache),
nl.
```

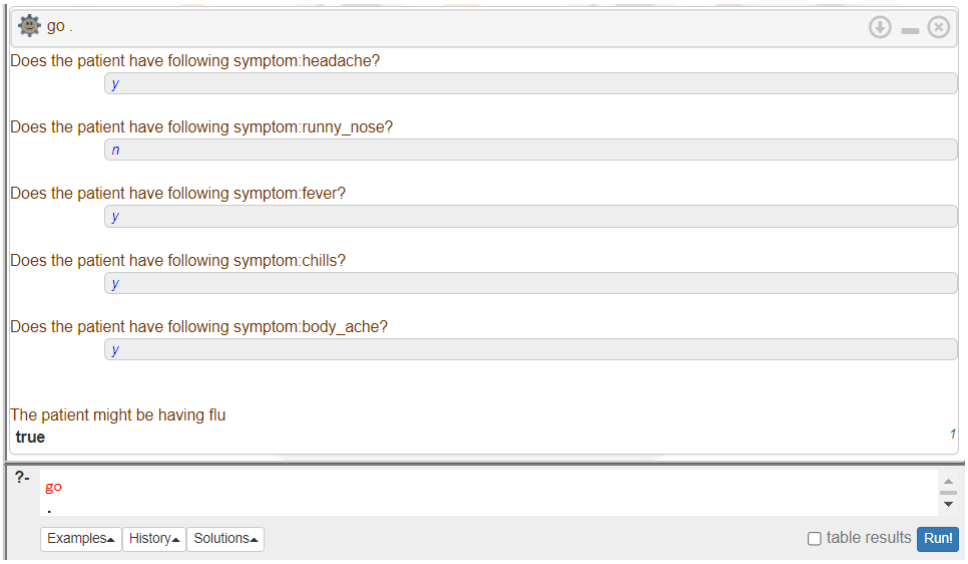
```
malaria :-
verify(fever),
verify(sweating),
verify(headache),
verify(nausea),
verify(vomiting),
verify(diarrhea),
nl.

mumps :-
verify(fever),
verify(swollen_glands),
nl.

chicken_pox :-
verify(fever),
verify(chills),
verify(bodyache),
nl.

tuberculosis :-
verify(fever),
verify(persistent_cough),
verify(constant_fatigue),
verify(weight_loss),
verify(coughing_blood),
verify(night_sweats),
nl.

pneumonia :-
verify(cough),
verify(fever),
verify(shaking_chills),
verify(shortness_of_breath),
nl.

byssinosis :-
verify(chest_tightness),
verify(cough),
verify(wheezing),
nl.

pertusis :-
verify(runny_nose),
verify(mild_fever),
nl.
```

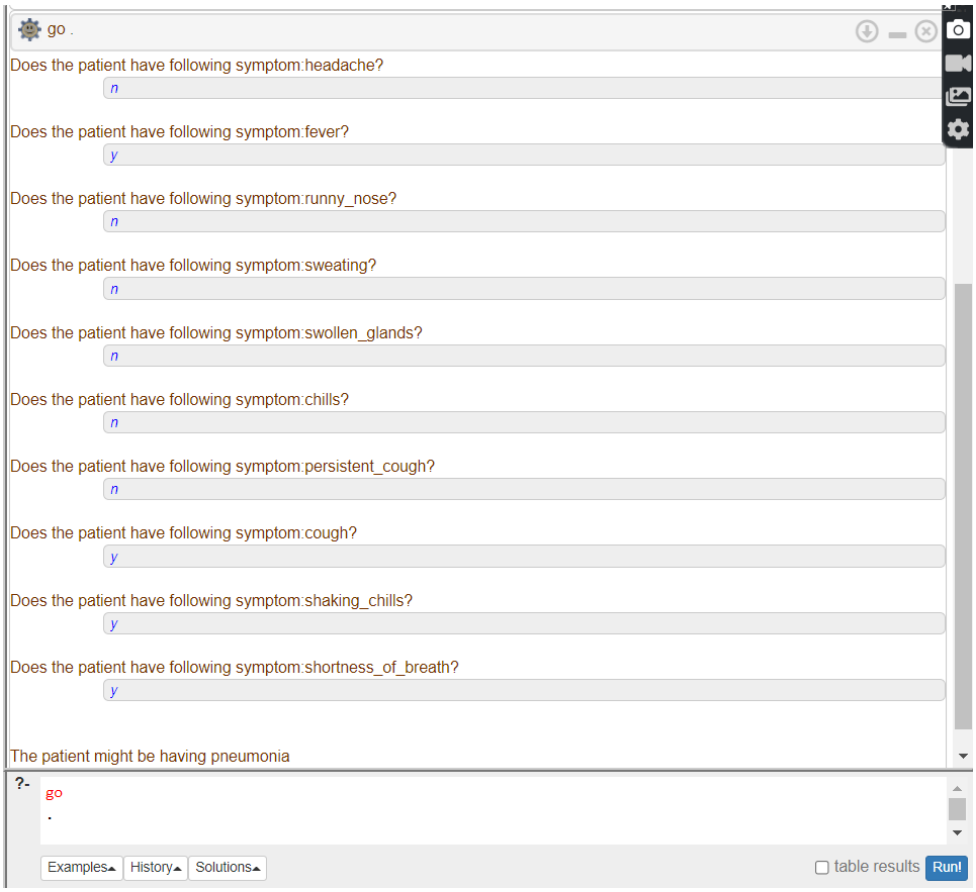
```
pneumoconiosis :-
verify(chronic_cough),
verify(shortness_of_breath),
nl.

sarcoidosis :-
verify(dry_cough),
verify(shortness_of_breath),
verify(mild_chest_pain),
verify(scaly_rash),
verify(fever),
verify(red_bumps_on_legs),
verify(sore_eyes),
verify(swollen_ankles),
nl.

asbestosis :-
verify(chest_tightness),
verify(shortness_of_breath),
verify(chest_pain),
verify(lack_of_appetite),
nl.

asthma :-
verify(wheezing),
verify(cough),
verify(chest_tightness),
verify(shortness_of_breath),
nl.

bronchiolitis :-
verify(wheezing),
verify(fever),
verify(blue_skin),
verify(rapid_breath),
nl.

influenza :-
verify(headache),
verify(fever),
verify(shaking_chills),
verify(nasal_congestion),
verify(runny_nose),
verify(sore_throat),
nl.

lung_cancer :-
```

```
verify(cough),
verify(fever),
verify(hoarseness),
verify(chest_pain),
verify(wheezing),
verify(weight_loss),
verify(lack_of_appetite),
verify(coughing_blood),
verify(headache),
verify(shortness_of_breath),
nl.

covid :-
verify(fever),
verify(cough),
verify(headache),
verify(bodyache),
nl.


diabetes :-
verify(fatigue),
verify(restlessness),
nl.


epilepsy :-
verify(headache),
verify(sunken_eyes),
nl.

glaucoma :-
verify(fever),
verify(sore_throat),
nl.

heart_stroke :-
verify(fainting),
verify(fever),
nl.

hyperthyroidism :-
verify(restlessness),
verify(sunken_eyes),
nl.

hypothermia :-
```

```
verify(fatigue),
verify(chest_pain),
verify(sore_throat),
nl.

jaundice :-
verify(cough),
verify(headache),
verify(fever),
nl.

sinusitis :-
verify(sore_throat),
verify(headache),
verify(fever),
nl.


/* how to ask questions */
ask(Question) :-
write('Does the patient have following symptom:'),
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 patient have following symptom:headache?
y
Does the patient have following symptom:runny_nose?
n
Does the patient have following symptom:fever?
y
Does the patient have following symptom:chills?
y
Does the patient have following symptom:body_ache?
y
The patient might be having flu
true
1
?-
go
.
Examples
History
Solutions
table results
Run!


go .
Does the patient have following symptom:headache?
n
Does the patient have following symptom:fever?
y
Does the patient have following symptom:runny_nose?
n
Does the patient have following symptom:sweating?
n
Does the patient have following symptom:swollen_glands?
n
Does the patient have following symptom:chills?
n
Does the patient have following symptom:persistent_cough?
n
Does the patient have following symptom:cough?
y
Does the patient have following symptom:shaking_chills?
y
Does the patient have following symptom:shortness_of_breath?
y
The patient might be having pneumonia
?-
go
.
Examples
History
Solutions
table results
Run!