**Batch: A1 Roll No.: 1911004**

**Experiment / assignment / tutorial No. 4**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title:** Implementation of medical diagnosis in prolog. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Objective:** To use the concepts of knowledge engineering to design and solve moderate complex problem.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **Course Outcome** | **After successful completion of the course students should be able to** |
| **CO3** | Represent the knowledge and reason through inference |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. **www.csupomona.edu/~jrfisher/www/prolog\_tutorial/contents.html**
2. **www.csupomona.edu/~jrfisher/www/prolog\_tutorial/pt\_framer.html**
3. **www.doc.gold.ac.uk/~mas02gw/prolog\_tutorial/prologpages/**
4. **classes.soe.ucsc.edu/cmps112/Spring03/languages/prolog/PrologIntro.pdf**
5. **“Prolog: Programming for Artificial Intelligence” by Ivan Bratko, Pearson education Publications**
6. **“Artificial Intelligence: a Modern Approach” by Russel and Norving, Pearson education Publications**
7. **“Artificial Intelligence” By Rich and knight, Tata Mcgraw Hill Publications**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

Agents, Agent Architecture, Programming with PROLOG

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Historical Profile:**

In [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence), an expert system is a computer system that emulates the decision-making ability of a human expert Expert systems are designed to solve complex problems by [reasoning](https://en.wikipedia.org/wiki/Automated_reasoning) through bodies of knowledge, represented mainly as [if–then rules](https://en.wikipedia.org/wiki/Rule-based_system) rather than through conventional [procedural code](https://en.wikipedia.org/wiki/Procedural_programming). The first expert systems were created in the 1970s and then proliferated in the 1980s.Expert systems were among the first truly successful forms of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) (AI) software

An expert system is divided into two subsystems: the [inference engine](https://en.wikipedia.org/wiki/Inference_engine) and the [knowledge base](https://en.wikipedia.org/wiki/Knowledge_base). The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned:**

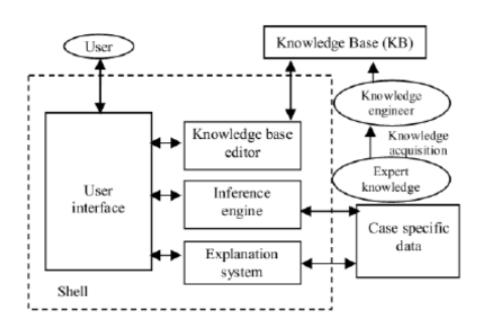
Knowledge engineering, implementing complex agent architecture, uncertainty in knowledge, Expert System.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Medical Diagnosis Chosen Statement: The medical expert system will tell the patient the type of disease he/she has on the basis of the symptoms given by the patient.**

**Expert System Architecture:**

An expert system is an example of a knowledge-based system. Expert systems were the first commercial systems to use a knowledge-based architecture. A knowledge-based system is essentially composed of two sub-systems: the knowledge base and the inference engine. Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if–then rules rather than through conventional procedural code. The first expert systems were created in the 1970s and then proliferated in the 1980s. An expert system is divided into two subsystems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities. The architecture of expert system has the following major components:



Knowledge Base (KB): repository of special heuristics or rules that direct the use of knowledge, facts (productions). It contains the knowledge necessary for understanding, formulating, & problem solving.

* Case specific data (Blackboard): if forward chaining used It describes the current problem & record intermediate results Records Intermediate Hypothesis & Decisions: 1. Plan, 2. Agenda, 3. Solution
* Inference Engine: the deduction system used to infer results from user input & KB. It is the brain of the ES, the control structure (rule interpreter). It provides methodology for reasoning
* Explanation Subsystem (Justifier): Traces responsibility & explains the ES behavior by interactively answering question: Why?, How?, What?, Where?, When?, Who?
* User Interface: interfaces with user through Natural Language Processing (NLP), or menus & graphics. Acts as Language Processor for friendly, problem- oriented communication

Shell = Inference Engine + User Interface

The Human Elements in ESs

* Expert: Has the special knowledge, judgement, experience and methods to give advice and solve problems. Provides knowledge about task performance
* Knowledge Engineer: Usually also the System Builder Helps the expert(s) structure the problem area by interpreting and integrating human answers to questions, drawing analogies, posing counter examples, and bringing to light conceptual difficulties. The Expert & the knowledge Engineer should Anticipate Users’ needs & Limitations when designing Expert Systems
* User: Possible Classes of Users can be:
* A non-expert client seeking direct advice (ES acts as a Consultant or Advisor)
* A student who wants to learn (ES acts as an Instructor)
* An ES builder improving or increasing the knowledge base(ES acts as a Partner)
* An Expert (ES acts as a Colleague or an Assistant).

For medical diagnosis system, I have designed an expert system where the medical system is applicable for kidney diseases. The expert system diagnoses the following diseases:

* + - 1. Chronic Kidney Disease
      2. Glomerulonephritis
      3. Polycystic kidney disease
      4. Kidney Cancer
      5. Kidney Stone

The expert system also makes simple treatment recommendations based on the disease diagnosed.

**Program Implementation**

**Code**

notice:-

write('This medical system diagnoses for Kideny Disease.').

hypothesis(kidney\_stone):- kidney\_stone, !.

hypothesis(kidney\_cancer):- kidney\_cancer, !.

hypothesis(polycystic\_kidney\_disease):- polycystic\_kidney\_disease, !.

hypothesis(glomerulonephritis):- glomerulonephritis, !.

hypothesis(chronic\_kidney\_disease):- chronic\_kidney\_disease, !.

hypothesis(unknown):- write('Sorry could not diagnose the disease.'). %no diagnosis%

kidney\_stone:-

verify(pain\_side\_back\_of\_body),

verify(fever),

verify(cloudy\_urine),

verify(unusual\_urine\_color),

verify(urge\_to\_urinate\_but\_urinate\_only\_small\_amount).

kidney\_cancer:-

verify(upset\_stomach),

verify(vomitting),

verify(food\_tastes\_metal),

verify(loss\_of\_appetite),

verify(unusual\_urine\_color),

verify(urge\_to\_urinate\_but\_urinate\_only\_small\_amount),

verify(nausea),

verify(pain\_side\_back\_of\_body).

polycystic\_kidney\_disease:-

verify(food\_tastes\_metal),

verify(upset\_stomach),

verify(vomitting),

verify(fever),

verify(nausea),

verify(blue\_skin),

verify(rapid\_breath).

glomerulonephritis:-

verify(swelling\_of\_limbs),

verify(fever),

verify(fatigue),

verify(itching),

verify(food\_tastes\_metal),

verify(upset\_stomach),

verify(vomitting),

verify(nausea).

chronic\_kidney\_disease:-

verify(headache),

verify(fever),

verify(fatigue),

verify(itching),

verify(pain\_side\_back\_of\_body),

verify(diarrhoea),

verify(high\_blood\_pressure).

treatment(kidney\_stone):-

write('Kindly drink more water, have more water food. Do consult doctor.').

treatment(kidney\_cancer):-

write('Kindly drink more water & watery food. Do consult doctor as soon as possible.').

treatment(polycystic\_kidney\_disease):-

write('Currently, consult doctor and stop other drugs with having more water & watery food.').

treatment(glomerulonephritis):-

write('Kindly take bed rest and plenty of water.It can risky for self treatment so visit the kidney specialist as soon as possible').

treatment(chronic\_kidney\_disease):-

write('Avoid fatty food, Kindly drink more water, have more water food. Do consult doctor and do certian tests.').

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.

verify(S) :- (yes(S)->true ;(no(S)->fail ;ask(S))).

start:-

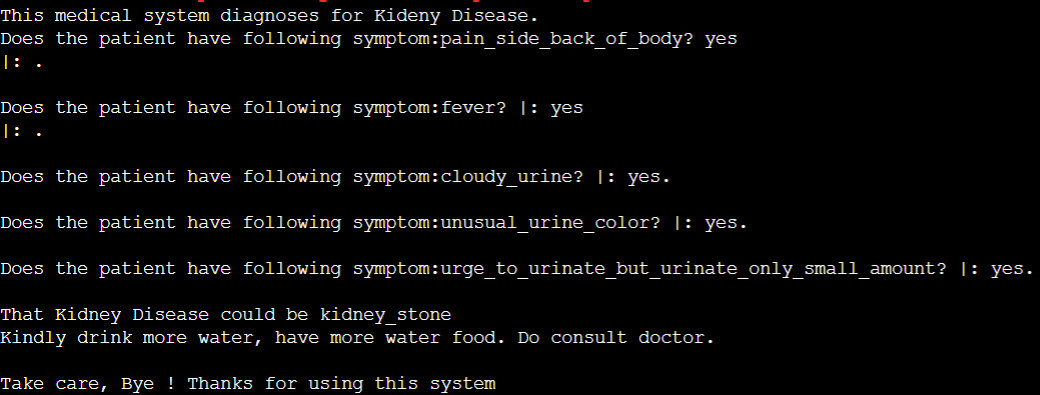
notice,nl, hypothesis(Disease), write('That Kidney Disease could be '),write(Disease),nl, treatment(Disease),nl,nl, write('Take care, Bye ! Thanks for using this system'), undo.

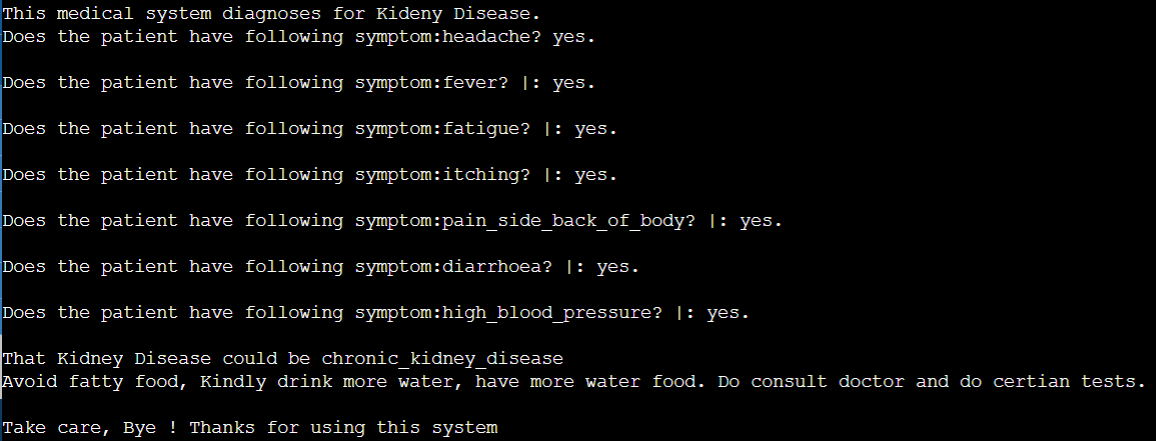
undo :- retract(yes(\_)),fail.

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

undo.

**Output**





**Conclusion:** The medical expert system for diseases related to Kidney was successfully understood & implemented in PROLOG.