Name : Tejas Redkar
PRN : 1032210937
Panel-C
RollNo : PC-44

AIES Lab Assignment-7

* 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 & inference engine.

* Theory

- Architecture of Expert System

The architecture of an expert system typically consists of three main components:

1) Knowledge Base (KB): This is the repository where domain-specific information, rules, & facts are stored. It includes both factual knowledge & heuristic rules that the expert system uses to make decisions or solve problems.

2) Inference Engine (IE): The inference engine is responsible for reasoning & drawing conclusions based on the information stored in the knowledge base. It uses various inference mechanics, such as forward chaining (data-driven) or backward chaining (goal-driven), to derive new knowledge or solve problems.

3) User Interface (UI): The user interface provides a means for users to interact with the expert system. It can be a text-based interface, a graphical-user interface (GUI), or a natural language interface, depending on the design. The UI allows users to input queries, receive advice, & understand the system's output.

- Main players of expert system

The main players of expert system in today's world are:

1) IBM (International Business Machines Corporation): IBM has been a major player in the development & application of expert systems. They have implemented expert systems in various domains, including finance, healthcare & customer support.

2) Oracle: Oracle has provided expert system tools & solutions for different industries. Their products often integrate with databases

& other enterprise systems.

3) Microsoft : Microsoft has been involved in the development of expert systems & AI technologies. Product like ~~Azu~~ Azure Machine learning & other AI services contribute to building intelligent systems.

4) Google : Google, with its expertise in machine learning & AI, plays a significant role in advancing technologies related to expert systems. Google AI cloud services offer solutions for building intelligent applications.

5) Expert System S.p.A : This company specializes in cognitive computing & AI technologies, providing solutions for natural language understanding & text analytics.

6) Cognitive Scale : A company focused on enterprise AI solutions, Cognitive Scale provides AI-powered software that includes ~~provid~~ capabilities for building expert systems.

* Input : Run the program on SWI Prolog
* Output : Give decisions based on the rules provided in the program.
* Platform : Linux / Windows
* FAQ's

Q1) Write in brief forward chaining & backward chaining of inference engine.

Ans Forward chaining:

In forward chaining, the inference engine starts with available data & uses rules to derive new conclusions. It begins with known facts & iteratively applies rules to reach a goal or conclusion. This approach is often used when the system is designed to solve a specific problem or make predictions based on existing data.

Backward Chaining:

In backward chaining, the inference engine starts with a goal or desired outcome & works backward through the rules & knowledge base to determine the necessary conditions or facts. It is a goal-driven approach where the system explores the rules & facts to find the information needed to satisfy the given goal. Backward chaining is commonly employed in diagonstic systems & troubleshooting scenarios.

Q2) List down the applications of expert system

Ans 1) Medical Diagnosis
2) Financial Advisoury
3) Customer Support
4) Manufacturing Process Control.
5) Education Tutoring
6) Natural Language Processing
7) Fault Diagnosis in Engineering
8) Computer Network Management
9) Quality Control
10) Agricultural Decision Support.

11/12/23

CODE:

```
% Define facts and rules
mammal(dog).
mammal(cat).
mammal(human).

has_fur(dog).
has_fur(cat).
has_fur(human).

gives_birth_to_live_young(dog).
gives_birth_to_live_young(cat).
gives_birth_to_live_young(human).

% Define the rule for determining if an animal is a mammal
is_mammal(Animal) :-
    mammal(Animal),
    has_fur(Animal),
    gives_birth_to_live_young(Animal).

% Sample Input and Output
% Query: is_mammal(dog).
% Output: true

% Query: is_mammal(snake).
% Output: false
```

INPUT

```
?- is_mammal(dog).
```

OUTPUT

```
true.
```