

AIM:

To develop a family tree program using PROLOG with all possible facts, rules and queries.

Procedure:

1. Create a file: Write the knowledge base in a text file and save it as family-relations.pl
2. Open prolog: Launch your prolog interpreter
3. Load the knowledge base: In prolog, load the file using
?- [family-relations]
4. Query the database: Ask queries based on the facts and rules.
5. View results: Prolog will return results for each query. Use the semicolon ; to see more answers.
6. Exit prolog: Type:
?- halt,
to exit the interpreter.

Output :

male (peter)

true

father (chris - peter)

true

father (chris - betty)

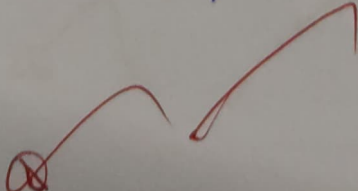
false

grandfather (kevin - peter)

true

Result:

The program was successfully executed and the O/P is verified.



EX-NO : 7

INTRODUCTION TO PROLOG

Aim :

To learn PROLOG terminologies and write basic programs

Procedure for using prolog with KB1 and KB 2

1. Create a file : Open a text editor and save the following knowledge bases as kb1.pl and kb2.pl respectively
2. Open prolog : launch your prolog interpreter (e.g., SWI-Prolog)
3. Load Knowledge Base ;
4. Execute Queries
5. View results : Prolog will provide answers to each query. If the query is true, it will return true ; if false, it will return false you can press ; to see more answers if available
6. Exit Prolog : Type :
?- halt
to exit the prolog interpreter

O/P : 1

? - woman (mia)

true

? - plays Air Guitar (mia)

false

? - party

true

? - concert

Error: Unknown procedure: concert / 0

Result:

The program was successfully executed
and the O/P is verified

EX.NO: 8

Unification and Resolution

Date:

Aim :

To execute programs based on unification and resolution. Deduction in prolog is based on the unification and instantiation. Matching terms are unified and variables get instantiation.

Procedure for executing prolog programs on Unification and resolution

1. Set Up Prolog Environment :

Open your prolog interpreter

2. Creating a knowledge base file :

Open a text editor and save the following prolog code as resolution - kb. pl :

3. Load the Knowledge Base :

In your prolog interpreter, load the knowledge base

4. Define goals for Refutation :

For Goal 1: Prove not-strawberry, picking

For Goal 2: Prove enjoy

5. Execute Queries for each Goal :

For Goal 1: check if strawberry - picking is true

For Goal 2: check if not (enjoy) is true.

6. Review results

7. Conclusion.

8. Exit Prolog:

after testing all goals.type:

? - halt

O/P:

? - notstrawberry - picking

true

? - enjoy

true

? - wet

true

Result:

The program was successfully executed
and the O/P is verified

AIM:

The aim of implementing fuzzy logic for edge detection is to enhance the robustness and accuracy of edge detection in images by handling uncertainties in pixel intensity transitions.

Procedure for Fuzzy Logic Edge detection.

Step 1: Setup the environment

1. Open MATLAB: Ensure you have access to MATLAB with the image processing toolbox and fuzzy logic toolbox installed.

Step 2: Import and convert image to grayscale

1. Read the RGB Image
2. Convert to Grayscale

Step 3: Convert image to double-precision data

1. Convert to double

Step 4: Obtain image gradient-

1. Define Gradient filters:

2. Calculate Gradients

3. Plot image gradients

Step 5: Define fuzzy inference system (FIS) for edge detection

1. Create FIS
2. Add inputs
3. Add Define Membership function for input
4. Add output
5. Define Membership functions for output
6. Plot Membership functions

Step 6: Specify FIS rules

1. Add rules for FIS

Step 7: Evaluate FIS

1. Evaluate edge detection

Step 8: Plot results

1. Plot original Grayscale Image
2. Plot detected edges

~~Step 9:~~
Result:

The program was successfully executed and the O/P is verified.