

Ex.no: 6

## PROLOG

Date:

Aim:

To develop a family tree program using Prolog with all possible facts, rules and answers.

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 of distribution function

male (Peter)

true

false (char-peter)

true

false (char-patty)

false

grandfather (konin-peter)

true

mother (konin-peter)

true

color not same; reading notes

(color-laz-mes) compression

good experience, looks

clear answers, a

want lot of answers. Way of answering

• good reader but there's no

such thing as note. Use the word

the answer's wrong lot of reading

questions

Result:

The program was successfully executed and the OLP is verified.

Ex.no: 7  
Date:

## Introduction To Prolog

Aim:

To learn Prolog terminologies and write basic Programs.

Procedure for writing Prolog with KBL and KBZ

1. Create a file : Open a text editor and save the following knowledge base, as kbl.pl and kbz.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.

Output:

? - women(ma).

true

? - Plays Air (Guitar(ma))

false

? - Party

true

? - Concert

Error: unknown procedure: Concert(10)

and afterwards copy number  
and then do nothing  
then again play 90-102-1  
Dorothy's play now 1073  
and afterwards a play  
and then do nothing test a song  
however do not play anything

and afterwards a play  
and then Dorothy play again  
and afterwards

~~Result:~~ played copy loop until

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

Ex.no:8

Date:

## Unification and Resolution

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 instantiated.

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 1: 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

Output:

? - rotstrawberry-picking

true

? - enjoy

true

? - wet

true

~~Result~~

~~Rotstrawberry-picking~~

The program was successfully executed and the output is verified