|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Applied Artificial Intelligence  Practical # 5 | | |
|  |  |  |  |
|  |  |  |  |
| **Name** | Ninad Karlekar | **Roll Number** | 22306A1012 |
| **Subject/Course:** | Applied Artificial Intelligence | **Class** | M.Sc. IT – Sem III |
| **Topic** | Rule Based System. | **Batch** | 1 |
|  |  |  |  |
| **A program to implement Rule Based System.** | | | |
| **AIM: Write a program which contains three predicates: male, female, parent. Make rules for following family relations: father, mother, grandfather, grandmother, brother, sister, uncle, aunt, nephew and niece, cousin.**   1. **Draw Family Tree.**      1. **Define: Clauses, Facts, Predicates and Rules with conjunction and disjunction**   a) Clauses: Statements or assertions in Prolog that can be facts, rules, or queries used for logical reasoning.  b) Facts: Simple statements or data in Prolog that describe relationships or properties, typically represented as predicate terms.  c) Predicates: Symbols or functions in Prolog that represent relationships or properties, typically followed by one or more arguments.  d) Rules: Logical constructs in Prolog that define relationships or properties based on conditions and can be used for inference and reasoning.  e) Conjunction: In Prolog, a logical operator (',') used to connect conditions, requiring all conditions to be true for a statement to be true.  f) Disjunction: In Prolog, a logical operator (';') used to connect conditions, allowing a statement to be true if at least one condition is true.  **DESCRIPTION:**  **Code:**  male(vijay).  male(mahadev).  male(gaurihar).  male(omkar).  male(bajrang).  male(chaitanya).  female(vasanti).  female(indubai).  female(ashwini).  female(gayatri).  female(sangita).  parent(vijay,chaitanya).  parent(vasanti,chaitanya).  parent(vijay,gaurihar).  parent(vasanti,gaurihar).  parent(vijay,ashwini).  parent(vasanti,ashwini).  parent(mahadev,vijay).  parent(indubai,vijay).  mother(X,Y):-parent(X,Y),female(X).  father(X,Y):- parent(X,Y), male(X).  grandmother(GM,X):- mother(GM,Y) ,parent(Y,X).  grandfather(GF,X):- father(GF,Y) ,parent(Y,X).  greatgrandmother(GGM,X):- mother(GGM,GM) ,parent(GM,F),parent(F,Y),parent(Y,X).  greatgrandfather(GGF,X):- father(GGF,GF) ,parent(GF,F),parent(F,Y),parent(Y,X).  sibling(X,Y):-mother(M,X), mother(M,Y),X\=Y, father(F,X), father(F,Y).  brother(X,Y):-sibling(X,Y), male(X).  sister(X,Y):-sibling(X,Y), female(X).  uncle(U,X):- parent(Y,X), brother(U,Y).  aunt(A,X):- parent(Y,X), sister(A,Y).  nephew(N,X):- sibling(S,X),parent(S,N),male(N).  niece(N,X):-sibling(S,X), parent(S,N), female(N).  cousin(X,Y):-parent(P,Y),sibling(S,P),parent(S,X).  **Output:** | | | |
| /\* Facts \*/  male(jack).  male(oliver).  male(ali).  male(james).  male(simon).  male(harry).  female(helen).  female(sophie).  female(jess).  female(lily).  parent\_of(jack, jess).  parent\_of(jack, lily).  parent\_of(helen, jess).  parent\_of(helen, lily).  parent\_of(oliver, james).  parent\_of(sophie, james).  parent\_of(jess, simon).  parent\_of(ali, simon).  parent\_of(lily, harry).  parent\_of(james, harry).  /\* Rules \*/  father\_of(X, Y):- male(X), parent\_of(X, Y).  mother\_of(X, Y):- female(X), parent\_of(X, Y).  grandfather\_of(X, Y):- male(X), parent\_of(X, Z), parent\_of(Z, Y).  grandmother\_of(X, Y):- female(X), parent\_of(X, Z), parent\_of(Z, Y).  sister\_of(X, Y):- female(X), father\_of(F, Y), father\_of(F, X), X \= Y.  sister\_of(X, Y):- female(X), mother\_of(M, Y), mother\_of(M, X), X \= Y.  aunt\_of(X, Y):- female(X), parent\_of(Z, Y), sister\_of(Z, X), !.  brother\_of(X, Y):- male(X), father\_of(F, Y), father\_of(F, X), X \= Y.  brother\_of(X, Y):- male(X), mother\_of(M, Y), mother\_of(M, X), X \= Y.  uncle\_of(X, Y):- parent\_of(Z, Y), brother\_of(Z, X).  ancestor\_of(X, Y):- parent\_of(X, Y).  ancestor\_of(X, Y):- parent\_of(X, Z), ancestor\_of(Z, Y). | | | |