Set 10

1. Write a Prolog program find a person is Mortal or NOT with the following facts.

Socrates, Einstein, Alexander.

man(socrates), man(Einstein), man(Alexander).

1.man(socrates).

man(einstein).

man(alexander).

mortal(X) :- man(X).

% Query examples

% To check if Socrates is mortal

% Query: mortal(socrates).

% Output: true

% To check if Einstein is mortal

% Query: mortal(einstein).

% Output: true

% To check if Alexander is mortal

% Query: mortal(alexander).

% Output: true

% To check if a hypothetical person 'Plato' is mortal

% Query: mortal(plato).

% Output: false

1. 2. Write a Prolog program on Resolution Refutation for the “happy student” problem.

1. Anyone passing his history exams and winning the lottery is happy.

2. Anyone who studies or is lucky can pass all his exams.

3. John did not study, but he is lucky.

4. Anyone who is lucky wins the lottery.

Prove that John is happy!

% Rules and Facts

happy(X) :- passedHistoryExams(X), winsLottery(X).

passedExams(X) :- studies(X); lucky(X).

studies(john) :- false.

lucky(john).

winsLottery(X) :- lucky(X).

% Negation of the conclusion: ¬happy(john)

:- not(happy(john)).

Output:

?- happy(john).

Set 9:

1. Write a Prolog program to find the number of vowels in the following fact.

This is my first Degree in Saveetha School of Engineering

vowel(a). vowel(e). vowel(i). vowel(o). vowel(u).

count\_vowels([], 0).

count\_vowels([H|T], Count) :-

(vowel(H) -> count\_vowels(T, TailCount), Count is TailCount + 1 ; count\_vowels(T, Count)).

count\_vowels\_in\_sentence(S, V) :- string\_chars(S, Chars), count\_vowels(Chars, V).

output:

?- count\_vowels\_in\_sentence("This is my first Degree in Saveetha School of Engineering", Vowels).

1. Write a Prolog program to implement Inference Engine with the following facts

interpret(true) :- !.

interpret((GoalA,GoalB)) :- !,

interpret(GoalA),

interpret(GoalB).

Code:

interpret(true) :- !.

interpret((GoalA, GoalB)) :- !,

interpret(GoalA),

interpret(GoalB).

?- interpret((true, true)).

?- interpret((true, (true, true))).

?- interpret((true, (true, false))).

Set 8:

1. Write a Prolog program to find items and its location using following facts

location(desk, office).

location(apple, kitchen).

location(flashlight, desk).

Code:

location(desk, office).

location(apple, kitchen).

location(flashlight, desk).

item\_location(Item, Location) :-

location(Item, Location).

find\_location(Item) :-

item\_location(Item, Location),

write('The location of '), write(Item), write(' is '), write(Location), nl.

Output:

?- find\_location(apple).

1. 2. Write a Prolog program to predict the Causes of Disease with the following predicates and clauses.

predicates

hypothesis(name,disease)

symptom(name,indication)

clauses

symptom(amit,fever).

symptom(amit,rash).

Code:

% Predicates

hypothesis(amit, disease).

symptom(amit, fever).

symptom(amit, rash).

% Rules

causes\_disease(Name, Disease) :-

symptom(Name, Symptom),

hypothesis(Name, Disease),

symptom\_exists\_for\_disease(Disease, Symptom).

symptom\_exists\_for\_disease(disease1, fever).

symptom\_exists\_for\_disease(disease2, rash).

Output:

?- causes\_disease(amit, Disease).

Set 7:

1. Write a Prolog program to implement pattern matching with the following facts.

first\_name(tonyblair, tony).

first\_name(georgebush, georgedubya).

second\_name(tonyblair, blair).

second\_name(georgebush, bush).

Code:

first\_name(tonyblair, tony).

first\_name(georgebush, georgedubya).

second\_name(tonyblair, blair).

second\_name(georgebush, bush).

match\_name(FirstName, SecondName, FullName) :-

first\_name(FullName, FirstName),

second\_name(FullName, SecondName).

Output:

?- match\_name(FirstName, SecondName, tonyblair).

?- match\_name(FirstName, SecondName, georgebush).

1. Write a Prolog program to implement Family Relation with the following facts.

female(sarah), female(rebekah), female(hagar\_concubine).

female(milcah), female(bashemath), female(mahalath).

female(first\_daughter), female(rachel), female(labans\_wife).

male(terah), male(abraham), male(nahor).

male(haran), male(isaac), male(ismael).

male(uz), male(kemuel).\

code:

% Facts about genders

female(sarah), female(rebekah), female(hagar\_concubine).

female(milcah), female(bashemath), female(mahalath).

female(first\_daughter), female(rachel), female(labans\_wife).

male(terah), male(abraham), male(nahor).

male(haran), male(isaac), male(ismael).

male(uz), male(kemuel).

% Relationships based on facts

parent(abraham, isaac).

parent(abraham, ismael).

parent(terah, abraham).

parent(haran, lot).

parent(nahor, uz).

parent(nahor, kemuel).

parent(milcah, bashemath).

parent(milcah, mahalath).

parent(rebekah, labans\_wife).

parent(rachel, joseph).

parent(labans\_wife, first\_daughter).

% Rules for family relationships

father(Father, Child) :-

parent(Father, Child),

male(Father).

mother(Mother, Child) :-

parent(Mother, Child),

female(Mother).

sibling(X, Y) :-

parent(Z, X),

parent(Z, Y),

X \= Y.

grandparent(Grandparent, Grandchild) :-

parent(Grandparent, Parent),

parent(Parent, Grandchild).

Set 6:

1. Write a Prolog program for the forward chaining using following facts

rainy(chennai).

rainy(coimbatore).

rainy(ooty).

cold(ooty).

Code:

rainy(chennai).

rainy(coimbatore).

rainy(ooty).

cold(ooty).

cold\_city(X) :-

rainy(X), cold(X).

output:

?- cold\_city(chennai).

?- cold\_city(ooty).

1. Write a Prolog program to predict Fruits and its color using Backtracking

colour(cherry, red).

colour(banana, yellow).

colour(apple, red).

colour(apple, green).

colour(orange, orange).

colour(X, unknown).

Code:

colour(cherry, red).

colour(banana, yellow).

colour(apple, red).

colour(apple, green).

colour(orange, orange).

colour(X, unknown).

Output:

?- colour(cherry, Color).

?- colour(apple, Color).

1. Write a Prolog program with the following facts, Dog-Name, Size-Small, Medium and Big

dog (fido), dog(rover), dog(jane), dog(tom), dog(fred), dog(henry), dog(fido), cat(mary), cat(harry), cat(bill), cat(steve), small(henry), medium(harry), medium(fred), large(fido), large(mary), large(tom), large(fred), large(steve), large(jim), large(mike).

Output:

% Dogs

dog(fido).

dog(rover).

dog(jane).

dog(tom).

dog(fred).

dog(henry).

% Cats

cat(mary).

cat(harry).

cat(bill).

cat(steve).

% Sizes

size(henry, small).

size(harry, medium).

size(fred, medium).

size(fido, large).

size(mary, large).

size(tom, large).

size(fred, large).

size(steve, large).

size(jim, large).

size(mike, large).

Output:

?- size(fido, Size).

1. Write a Prolog program for a Planets Database with the following facts.

orbits(mercury, sun).

orbits(venus, sun).

orbits(earth, sun).

orbits(mars, sun).

orbits(moon, earth).

orbits(phobos, mars).

orbits(deimos, mars).

Code:

% Facts about planets and their orbits

orbits(mercury, sun).

orbits(venus, sun).

orbits(earth, sun).

orbits(mars, sun).

orbits(moon, earth).

orbits(phobos, mars).

orbits(deimos, mars).

Output:

?- orbits(mercury, Planet).

1. Write a Prolog program with Name, DOB with for following required facts.

born(jan, 20,3,1977).

born(jeroen, 2,2,1992).

born(joris, 17,3,1995).

born(jelle, 1,1,2004).

born(jesus, 24,12,2000).

born(joop, 30,4,1989).

born(jannecke, 17,3,1993).

born(jaap, 16,11,1995).

Code:

% Facts about names and dates of birth (DOB)

born(jan, 20, 3, 1977).

born(jeroen, 2, 2, 1992).

born(joris, 17, 3, 1995).

born(jelle, 1, 1, 2004).

born(jesus, 24, 12, 2000).

born(joop, 30, 4, 1989).

born(jannecke, 17, 3, 1993).

born(jaap, 16, 11, 1995).

Output:  
?- born(jan, Day, Month, Year).

1. Write a Prolog program with FName, SName, Sex, Age and Occupation

person(frances,wilson,female,28,architect).

person(fred,jones,male,62,doctor).

person(paul,smith,male,45,plumber).

person(martin,williams,male,23,chemist).

person(mary,jones,female,24,programmer).

person(martin,johnson,male,47,solicitor).

man(A):-person(A,B,male,C,D).

code:

% Facts about individuals: FName, SName, Sex, Age, Occupation

person(frances, wilson, female, 28, architect).

person(fred, jones, male, 62, doctor).

person(paul, smith, male, 45, plumber).

person(martin, williams, male, 23, chemist).

person(mary, jones, female, 24, programmer).

person(martin, johnson, male, 47, solicitor).

% Rule to identify men

man(A) :- person(A, \_, male, \_, \_).

Output:

?- man(A).

Set 3:

1. Write a Prolog program to implement the best sequence of actions for the monkey? A monkey is in a room. Suspended from the ceiling is a bunch of bananas, beyond the monkey's reach. However, in the room there are also a chair and a stick. The ceiling is just the right height so that a monkey standing on a chair could knock the bananas down with the stick. The monkey knows how to move around, carry other things around, reach for the bananas, and wave a stick in the air.

Code:

% Actions the monkey can perform

action(move(monkey, room, chair)).

action(move(chair, room, bananas)).

action(move(stick, room, monkey)).

action(use(stick, wave)).

action(use(monkey, reach)).

% Rules defining reachable conditions for bananas

reachable(bananas) :- action(move(monkey, room, chair)), action(move(chair, room, bananas)).

reachable(bananas) :- action(move(monkey, room, bananas)).

reachable(bananas) :- action(move(stick, room, monkey)), action(use(stick, wave)).

% Best sequence of actions to get the bananas

best\_sequence\_of\_actions(ActionList) :-

reachable(bananas),

findall(Action, reachable(bananas), ActionList).

Output:

?- best\_sequence\_of\_actions(ActionList).

1. Write a Prolog program for forward Chaining. **As per the law, it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert, who is an American citizen.** Prove that **"Robert is criminal."**

Code:

% Facts

american(robert).

enemy(country\_a, america).

owns(country\_a, missiles).

sells(X, missiles, country\_a) :- american(X).

criminal(X) :- sells(X, weapons, Y), enemy(Y, america).

% Rules

sells(robert, weapons, country\_a). % Given: Robert sells weapons to country A

output:

?- criminal(robert).

Set 2:

1. Write a Prolog program to print particular bird can fly or not using following facts.

Bird (eagle).

Bird (sparrow).

Bird (penguin).

Fly (penguin) :- !, fail.

Fly(X) :- bird(X).

Code:

bird(eagle).

bird(sparrow).

bird(penguin).

fly(penguin) :- !, fail.

fly(X) :- bird(X).

output:

?- fly(eagle).

1. Write a Prolog program to implement family tree with the following given facts.

Pam, Liz, Ann and Pat are female, while Tom, Bob and Jim are male persons. Using this information, define the following relations:

Define the “mother” relation:

Define the “father” relation:

Define the “grandfather” relation:

Define the “grandmother” relation:

Define the “sister” relation

Define the “brother” relation

Code:

% Facts about individuals and their genders

female(pam).

female(liz).

female(ann).

female(pat).

male(tom).

male(bob).

male(jim).

% Define mother relation

mother(pam, liz).

mother(pam, tom).

mother(bob, pat).

% Define father relation

father(jim, liz).

father(jim, tom).

father(bob, pat).

% Define grandfather relation

grandfather(X, Y) :- father(X, Z), (mother(Z, Y); father(Z, Y)).

% Define grandmother relation

grandmother(X, Y) :- mother(X, Z), (mother(Z, Y); father(Z, Y)).

% Define sister relation

sister(X, Y) :- female(X), female(Y), mother(Z, X), mother(Z, Y), X \= Y.

% Define brother relation

brother(X, Y) :- male(X), male(Y), mother(Z, X), mother(Z, Y), X \= Y.

output:

?- mother(Mother, Child).

% Output: Mother = pam, Child = liz ;

% Mother = pam, Child = tom ;

% Mother = bob, Child = pat.

?- sister(Sister, Brother).

% Output: Sister = liz, Brother = tom ;

% Sister = pat, Brother = tom ;

% Sister = liz, Brother = pat ;

% Sister = pat, Brother = liz ;

% false. % Indicates there are no more solutions.

?- grandfather(Grandfather, Grandchild).

% Output: Grandfather = jim, Grandchild = pat ;

% Grandfather = jim, Grandchild = pat ;

% false. % Indicates there are no more solutions.

Set1:

1. Write a Prolog Program for Student, Teacher and Sub-Code using following facts.

Studies (charlie, csc135).

Studies (olivia, csc135).

Studies (jack, csc131).

Studies (arthur, csc134).

Teaches (kirke, csc135).

Teaches (collins, csc131).

Teaches (collins, csc171).

Teaches (juniper, csc134).

Professor (X, Y) :- teaches(X, C),

Studies (Y, C).

Code:

% Facts about students and their courses

studies(charlie, csc135).

studies(olivia, csc135).

studies(jack, csc131).

studies(arthur, csc134).

% Facts about teachers and the courses they teach

teaches(kirke, csc135).

teaches(collins, csc131).

teaches(collins, csc171).

teaches(juniper, csc134).

% Rules to determine professors based on courses

professor(X, Y) :- teaches(X, C), studies(Y, C).

output:

?- professor(Professor, Student).

1. Write a Prolog program to solve Towers of Hanoi with mathematical puzzle where we have three rods (**A**, **B**, and **C**) and **N** disks. Initially, all the disks are stacked in decreasing value of diameter and smallest disk is placed on the top and they are on rod **A**.

Code:

hanoi(1, A, \_, C) :-

write('Move top disk from '), write(A), write(' to '), write(C), nl.

hanoi(N, A, B, C) :-

N > 1,

M is N - 1,

hanoi(M, A, C, B),

hanoi(1, A, \_, C),

hanoi(M, B, A, C).

output:

?- hanoi(3, 'A', 'B', 'C').