Register Number

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Department of Computer Science and Engineering

Continuous Assessment Test – II

Answer Key

Degree & Branch	B.E & CSE				**Semester**	V
Subject Code & Name	UCS1524 – Logic Programming				**Regulation:**	**2018**
Academic Year	2022-2023 (Odd)	**Batch**	2020-2024	**Date**	**20-10-22**	**FN**
Time: 8.15 to 9.45	**Answer All Questions**				**Maximum: 50 Marks**	

Part – A (6×2 = 12 Marks)

<table>
<tr><th>K Level</th><th>Questions</th><th>COs</th><th>PIs</th></tr>
<tr><td>K1</td><td>1. What is swapping lemma?
The evaluation order of procedure calls is not relevant, and can be swapped without changing the computation result.</td><td>CO2</td><td>1.4.1</td></tr>
<tr><td>K2</td><td>2. Compare procedural semantics and declarative semantics.
<table>
<tr><td>Procedural</td><td>Declarative</td></tr>
<tr><td>To do a, first do b1, then do b2, …, then do bm</td><td>a is true if b1 and b2 and … and bm are all true</td></tr>
<tr><td>does depend on the order of goals and clauses</td><td>does not depend on the order of the clauses and the order of the goals in clauses</td></tr>
</table></td><td>CO2</td><td>1.4.1</td></tr>
<tr><td>K2</td><td>3. Explain type-1 non-determinism with an example.
Suppose, a particular literal is selected already (i.e. a procedure call) in the goal clause which is to be unified with some procedure head of some program clause. If there are several such program clauses which can be used to produce resolvents, we call this <i>type 1 nondeterminism.</i>
Goal clause ?- <i>A,B,C</i>
Program clauses
<i>B :-D.</i>
<i>B.</i>
<i>B :-E,F</i>

3 SLD resolvents
<i>?- A,D,C.</i>
<i>?- A,C.</i>
<i>?- A,E,F,C.</i></td><td>CO2</td><td>1.4.1</td></tr>
<tr><td>K1</td><td>4. Consider the following operator directives
op(100, xfx, +).
op(200, xfx, -).
What is the precedence of (b + c) in an expression (a – b) + (b + c)?
0</td><td>CO3</td><td>1.4.1</td></tr>
</table>

K1	5. What are the outputs of the following goals in Prolog? a. X is *(-(4, 3), +(2, 3)). b. X = *(-(4, 3), +(2, 3)). a. X = 5 b. X = (4-3)*(2+3)	CO3	1.4.1
K3	6. Identify the functionality of ‘fun’ in the Prolog code given below. Find the answer for the query “fun(6,R)”. fun(0,0). fun(N,R) :- N > 0, N1 is N-2, fun(N1,R1) , R is R1+N. to find the sum of even numbers answer=12	CO3	2.1.2 2.1.3 2.2.3 13.3.1

Part – B (3×6 = 18 Marks)

<table>
<tr><td>K2</td><td>7. Explain Model theoretic semantics with an example.
Proud_parent(X) :- parent(X, Y), newborn(Y)
parent(X, Y) :- father (X, Y)
parent(X, Y) :- mother (X, Y)
father(adam, mary)
newborn(mary)

Herbrand Universe <b>is the set of all ground</b> terms that can be formed from the constants and function symbols
{adam, mary}
Berbrand Base <b>is the set of all ground goals</b> that can be formed from the predicates in P and the terms in the Herbrand universe
{proud_parent(adam, mary), proud_parent(mary, adam), ------}
Minimal model: What is implied as true by the program is true; everything else is false
father(adam, mary)
newborn(mary)
parent(adam,mary)
proud_parent(adam)</td><td>CO2</td><td>1.4.1</td></tr>
<tr><td>K2</td><td>8. Explain the following list operations with Prolog rules.
a. deletes an element from the list
b. permutation
Illustrate these operations for a character list.

del(X, [X| Tail], Tail).
del(X, [Y| Tail], [Y|Tail1]) :- del(X, Tail, Tail1).

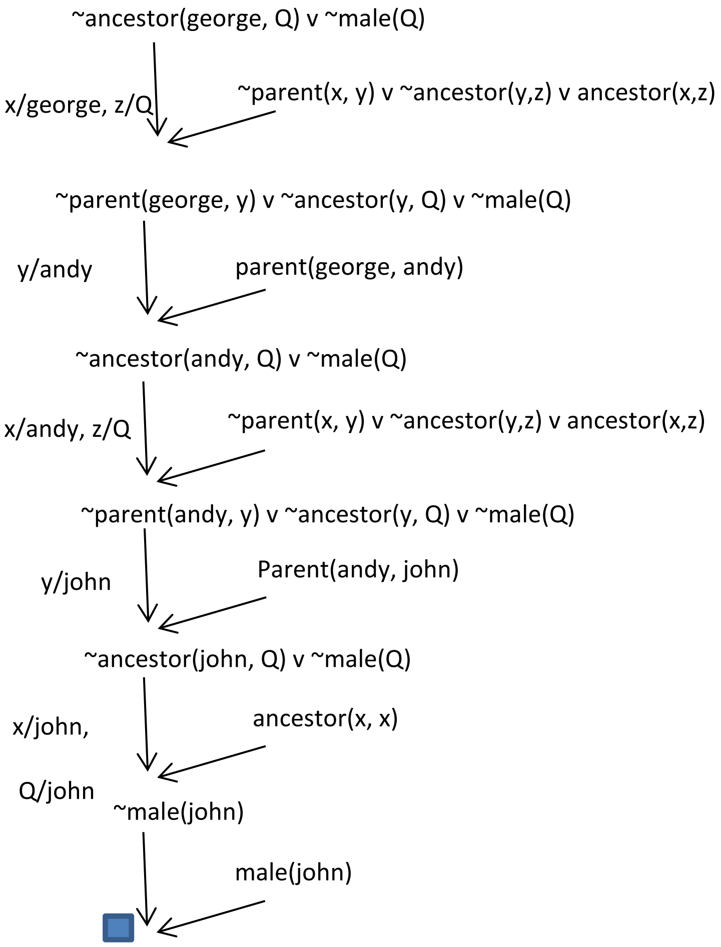
?del(b, [a,b,c], R).

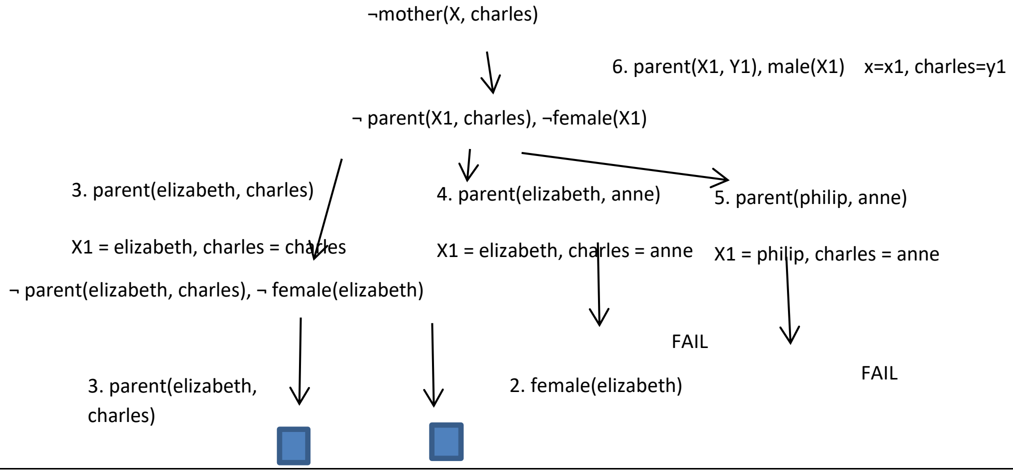
permutation1([],[]).
permutation1([X| L], P):- permutation1(L, L1), insert(X, L1, P).
insert(X, List, BiggerList) :- del(X, BiggerList, List).

?permutation1([a,b,c,d], P)</td><td>CO3</td><td>1.4.1</td></tr>
<tr><td>K3</td><td>9. Draw the structure of a family knowledge base. Write facts with two unary relations: ‘male’ and ‘female’ and a binary relation: ‘parent’ for the given knowledge base. Construct Prolog rules to define the following relations.
a. uncle
b. aunty</td><td>CO3</td><td>1.4.1
2.1.2
13.3.1</td></tr>
</table>

	Illustrate these relations for the family knowledge base with appropriate queries. brother(X, Y) :- parent(Z, X), parent(Z, Y), male(X), X \= Y. sister(X, Y) :- parent(Z, X), parent(Z, Y), female(X), X \= Y. uncle(X,Y) :- parent(Z, Y), brother(X, Z). aunty(X,Y) :- parent(Z, Y), sister(X, Z).		

Part – C (2×10 = 20 Marks)

K 3	10. Consider the following knowledge base. Apply horn clause programming logic to find the male descendant of George. ancestor(X,X). ancestor(X,Z) :- parent(X,Y), ancestor(Y,Z). parent(george,sam). parent(george,andy). parent(andy,mary). parent(andy,john). male(george). male(sam). male(andy). male(john). female(mary).  ~ancestor(george, Q) v ~male(Q) x/george, z/Q ~parent(x, y) v ~ancestor(y,z) v ancestor(x,z) ~parent(george, y) v ~ancestor(y, Q) v ~male(Q) y/andy parent(george, andy) ~ancestor(andy, Q) v ~male(Q) x/andy, z/Q ~parent(x, y) v ~ancestor(y,z) v ancestor(x,z) ~parent(andy, y) v ~ancestor(y, Q) v ~male(Q) y/john Parent(andy, john) ~ancestor(john, Q) v ~male(Q) x/john, ancestor(x, x) Q/john ~male(john) male(john)	CO 2	1.4.1 2.1.2 13.1. 2
	(OR)		
K 3	11. Consider the following logic program. Apply procedural semantics to find the mother’s name of Charles and show the semantics using a tree structure. male(philip).	CO 2	1.4.1 2.1.2

	female(elizabeth). parent(elizabeth, charles). parent(elizabeth, anne). parent(philip, anne). father(X, Y) :- parent(X, Y), male(X). mother(X, Y) :- parent(X, Y), female(X).  ¬mother(X, charles) 6. parent(X1, Y1), male(X1) x=x1, charles=y1 ¬ parent(X1, charles), ¬female(X1) 3. parent(elizabeth, charles) 4. parent(elizabeth, anne) 5. parent(philip, anne) X1 = elizabeth, charles = charles X1 = elizabeth, charles = anne X1 = philip, charles = anne ¬ parent(elizabeth, charles), ¬ female(elizabeth) FAIL FAIL 3. parent(elizabeth, charles) 2. female(elizabeth)		13.1.2
K 3	12. Given four facts with ‘point’ relation: point(X1,Y1), point(X2, Y2), point(X3,Y3) and point(X4,Y4). Construct Prolog rules to define following structures. Illustrate the same with example queries. a. Square b. Rectangle c. Isosceles trapezoid d. Parallelogram e. Rhombus square(point(X1,Y1), point(X2, Y2), point(X3,Y3), point(X4,Y4)) :- Y1=:=Y2, X2=:=X3, X4=:=X1, Y3=:=Y4. square(point(1,1), point(2, 1), point(2,2), point(1,2)). rectangle(point(X1,Y1), point(X2, Y2), point(X3,Y3), point(X4,Y4)) :- Y1=:=Y2, X2=:=X3, X4=:=X1, Y3=:=Y4, Y4 < X2. rectangle(point(1,1), point(3, 1), point(3,2), point(1,2)). trapezoid (point(X1,Y1), point(X2, Y2), point(X3,Y3), point(X4,Y4)) :- Y1=:=Y2, X3 < X2, X4 > X1, Y3=:=Y4. trapezoid(point(1, 1), point(9,1), point(7,6), point(3,6)). parallelogram(point(X1,Y1), point(X2, Y2), point(X3,Y3), point(X4,Y4)) :- Y1=:=Y2, X3 > X2, X4 > X1, Y3=:=Y4. parallelogram(point(1, 1), point(9,1), point(10,6), point(3,6)). rhombus(X1,Y1), point(X2, Y2), point(X3,Y3), point(X4,Y4)) :- Y2=:=Y4, X1=:=X3, X3<X2, Y2<Y3. rhombus(point(1,2), point(2, 2), point(3,2), point(2,3)).	CO 3	1.4.1 2.1.2 2.2.3 13.3.1
	(OR)		
K 3	13. Develop Prolog programs for the following and illustrate the same with appropriate queries. a. Let L1 = [1,2,3] and L2 = [a,b]. Concatenate L1 and L2 to get L3 = [a,b,1,2,3] b. Let L = [m, a, d, a, m]. Use concatenation function to reverse the list and check to whether L is palindrome or not. c. Let L = [5,7,9,1,3]. Find sum of elements in the list. d. Let L = [5,7,9,1,3]. Find the maximum number in the list.	CO 3	1.4.1 2.1.2 2.2.3 13.3.1

	a. conc(L, [], L). conc(L1, [X|L2], [X| L3]) :- conc(L1, L2, L3). b. conc([], L, L). conc([X| L1], L2, [X| L3]) :- conc(L1, L2, L3). reverse1([], []). reverse1([First | Rest], Reversed) :- reverse1(Rest, ReversedRest), conc(ReversedRest, [First], Reversed). palin(L) :- reverse1(L, L). c. sumlist([], 0). sumlist([First | Rest], Sum) :- sumlist(Rest, SumRest), Sum is First + SumRest. d. max(X, Y, X) :- X >= Y. max(X, Y, Y) :- X < Y. maxlist([X], X). maxlist([X, Y | Rest], Max):- maxlist([Y | Rest], MaxRest), max(X, MaxRest, Max).		

Prepared By D. Thenmozhi	Reviewed By	Approved By
Course Coordinator	PAC Team	HOD