PART I UI8C0051_KRUNAL RANIK

Ans 1:

a) For the given query, we get,

A = true and A = true [Satisfying 2nd and 3rd facts]

b) For the given query, we get,

A = false [No available rule]

c) For the given query, we get,

C = mon [Because there are 6 swi nested and every time, 3 gets removed due to 4th rule]

d) For the given query, we get,

D = swi(1),

D = swi(swi(swi(swi(1))))

D = swi(swi(swi(swi(swi(swi(swi(1)))))))

and so on...

This occurs due to recursiveness of rule 4 and base rule 3.

PART 1 U18CO081 - KRUNAL RANK 2/5

Ans 2)

a) Some advantages of predicate logic over propositional logic are as follows:-

- It helps us to analyse the scope of subject over the predicate.
- We can also use some of the ~~universal~~ quantifiers to express a statement.
- It can be used to specialize statements and create meaningful clauses.
- It's truth value is definite - its completely based on the variables' value.

b) ~~Remove~~

Sentence to predicate:-

1) Rita is 21 years old ⇒ eq(Age(Rita, 21)).

2) Every ~~indig~~ indian who is more than 18 years old can vote,

$\forall x$: Indian(x) $\wedge$ gt(Age(x), 18) $\rightarrow$ Vote(x).

3) Rita & Sita belong to India.

belongsto(Rita, India) $\wedge$ belongsto(Sita, India)

4) Rita & Sita are twin sisters.

Female(Rita) $\wedge$ Female(Sita) $\wedge$ twinsister(Rita, Sita).

PART I UI8C0051_KRUNAL RANK 3/5

Ans 3:

1. print_details :-
Reservation(Travaler(Name, _), Train(TNo, TName, Source, Destination, Date of Travel(22,10,2021), Confirm(CNF)),
write("("),
write(Name),
write(","),
write(TNo),
write(","),
write(TName),
write(","),
write(Source),
write(","),
write(Destination),
write(")").

2. print_trains :-
~~(Name X) Reservation(Traveler(_, _), Train~~
findall(~~Name~~, Reservation(_, Train(_, Name, "Surat", "Mumbai", _, X, _)), LL),
unique(LL, NewList),
remove_less_than_1000(NewList, final),
write(final).
//get unique names
unique([],[]).
unique([(Name, _)|T], [H|T1]) :-
forall(member((N, _), T), N \= Name), unique(T, T1), !.
unique([_|T], T1) :- unique(T, T1).

PART I _U18CO084_ KRUNAL RANK 4/5

remove_more_than_1000([(_, Price)|T], NewList) :-
Price >= 1000,
remove_more_than_1000(T, NewList).

remove_more_than_1000([], []).
remove_more_than_1000([H|T], [H1|T1]).

3.:

print_travellers :-
forall((X, Y, Z, A, B), Reservation(Traveller(X, Y), Train(Z, A, _, _, _, _), Waiting(WL)), LL),
write(LL).

PART I U18CO081 - KRUNAL RANK 5/5

Ans 4:

a) delete(L, L1) :-
concat_lists(L, [---], L).

b) delete(L, L1) :-
concat_lists(L, [---|T], L).

Ans 4:

a) delete(L, L1) :-
conc(L1, [---], L).

b) delete(L, L2) :-
conc([---], L1, L),
conc(L2, [---], L1).