Solutions for Logic Programming Qualifying Examination – Fall 1998

1. *Tracing Meta-interpreter:*

```
interpret(true,Level) :- !.
interpret((Goal1,Goal2), Level) :-
        interpret(Goal1,Level),
        interpret(Goal2,Level).
interpret(Goal, Level) :-
        tab(Level), write('Call:'),write(Goal), nl,
        clause(Goal,Body),
        Level1 is Level + 4,
        interpret(Body, Level1),
        tab(Level), write('Exit:'),write(Goal),nl,
        redo_message(Goal,Level).
interpret(Goal,Level) :-
        tab(Level), write('Fail:'),write(Goal),nl,
        fail.

redo_message(Goal,Level).
redo_message(Goal,Level) :-
        tab(Level),write('Redo:'),write(Goal),nl,
        fail.
```

2. *Meta-interpreter for SLD-Tree:*

```
go(true,true,true) :- !.
go(true,Cont,Tree) :- !,
        go(Cont,true,Tree).
go((true,G),Cont,Tree) :- !,
        go(G,Cont,Tree).
go((G1,G2),Cont,Tree) :-!,
        current_goal((G1,G2),Cont,Goal),
        findall(Body,clause(G1,Body),List),
        (List = [] -> Tree = tree(Goal,[fail]);
                        generate_trees(List,(G2,Cont),Trees),
                        Tree = tree(Goal,Trees)
        ).
go(G,Cont,Tree) :-
```

```
        current_goal(G,Cont,Goal),
        findall(Body,clause(G,Body),List),
        (List = [] -> Tree = tree(Goal,[fail]);
                        generate_trees(List,Cont,Trees),
                        Tree = tree(Goal,Trees)).

current_goal((true,G),G2,Goal) :- !,
        current_goal(G,G2,Goal).
current_goal((Ga,Gb),G2,(Ga,Goal)) :-!,
        current_goal(Gb,G2,Goal).
current_goal(true,G,G) :- !.
current_goal(G,Ga,(G,Ga)).



generate_trees([],_,[]).
generate_trees([X|Y],Cont,[T|Rest]) :-
        go(X,Cont,T),
        generate_trees(Y,Cont,Rest).
```

3. *Fair Computation Rule:* meta-interpretation can be used to avoid infinite computation rule, for example, by introducing a breadth-first strategy in the exploration of the resolution tree. In the meta-interpreters above, the generation of the tree is depth-first, since the expansion of a goal is a recursive call to the interpreter itself. The meta-interpreter can be modified to implement breadth-first by introducing a queue of goals; at each stage of the execution, the queue contains a representation of the current leaves—true, fail, or a goal which has not been expanded yet—of the resolution tree.