**Assignment 4**

**1. (10 Marks) Describe briefly the working principles of PROLOG.**

Ans: Prolog is a logical programming language which is also said to be a declarative language. All the logic in Prolog is defined using relations such as Facts and Rules. Then these logic (Facts and Rules) are used to perform Computation.

Working Principles

**Resolution**: A mechanism of proving something is equal to something based on the inner relation is known as Resolution. This can be understanded more easily using an example; Let’s say all women like shopping and Becky is a woman that means Becky likes shopping. More over this can be simplified using symbolic representation, a = b and b = c that means a = c. This is resolution.

**Develop and Test**: When a fact is tested and based on that fact other facts are tested to see the relation that the bounding of a variable might change. This state of change is very useful to solve compound constraints. For example, part(X, katija) is called it will give us a value of X and then when further part(Y, X) is called, X will be used to find the value of Y. So at the time of First call of ‘part’ X was unbounded but at the time of second ‘part’ call X was bounded.

**Backtracking**: A statement or computation which might have more than one answer to it. For example, children(X, salica). This relation might satisfy more than one Fact. This is known as Nondeterminism. Using Backtracking, Prolog solves the issue of nondeterminism. Undoing all the things done to try alternative choice since a hesitant choice was made.

**Debugger**: For a debugger in Prolog there are four performance steps, Calling of an entry in the goal, second exiting of an entry upon successful compilation, third retrying or backtracking into the goal, and last failing if the entry was unsuccessful.

**2. (10 Marks) Describe briefly the working principles of LISP.**

LISP is known as a meta programming language or symbolic programming language.

Some of the working principle of LISP:

Local Environment Setup

**3. (20 Marks, 5/each) Write both a PROLOG program and LISP program for the following problems**

**a) Find the length of a list.**

size([], 0).

size([\_|T], N) :-

size(T, N1), N is N1 + 1.

**b) Find the average of a list of numbers.**

sum([], 0).

sum([H|T], N) :- sum(T, N1), N is N1 + H.

size([], 0).

size([\_|T], N) :-

size(T, N1), N is N1 + 1.

average([], 0).

average(L, N) :- sum(L, S), size(L, N1), N is S / N1.

**c) Merge sort a list of integers.**

sort([],[]).

sort([H],[H]).

sort([H, H1|T], S) :-

split([H, H1|T], L1, L2),

sort(L1, S1),

sort(L2, S2),

merge(S1, S2, S).

split([], [], []).

split([H], [H], []).

split([H1, H2|T], [H1|T1], [H2|T2]) :-

split(T, T1, T2).

merge(H, [], H).

merge([], H, H).

merge([H1|T1], [H2|T2], [H1|R]) :-

H1 =< H2, merge(H1, [H2|T2], R).

merge([H1|T1], [H2|T2], [H2|R]) :-

H1 > H2, merge([H1|T1], T2, R).

**d) Reverse a list.**

reverse([], []).

reverse([H], [H]).

reverse([H|T], R) :-

reverse(T, R1), append(R1, [H], R).

**4. (15 Marks, 5/each) Page 676 (Programming exercises)**

**questions 5, 6, 13**

**5. (15 Marks, 5/each) Page 715 (Programming exercises)**

**questions 3**

maxlist([], \_).

maxlist([H], H).

maxlist([H, H1|T], M) :-

maxlist([H1|T], M\_Rest),

max(H, M\_Rest, M).

max(H1, H2, H1) :- H1 >= Y.

max(H1, H2, H2) :- H1 < Y.

**questions 5**

union([H|T], L2, U) :-

list\_member(H, L2), union(T, L2, U).

union([H|T], L2, [H|U]) :-

\+ list\_member(H, L2), union(T, L2, U).

union([], L2, L2).

list\_member(H, [H|\_]).

list\_member(H, [\_|T]) :-

list\_member(H, T).

**questions 7**

sort([],[]).

sort([H|T], R) :-

separate(H, T, S, L),

sort(S, SS),

sort(L, SL),

append(SS, [H|SL], R).

separate(M, [], [] , []).

separate(M, [H|T], [H|S] , L) :-

H =< M,

separate(M, Y, S, L).

separate(M, [H|T], S, [H|L]) :-

H > M,

separate(M, T, S, L).

**6. (10 marks) Give a short definition/description of the following concepts:**

**- sextuple of attributes of variables**

**- scope (static and dynamic) of a variable**

**- binding (static and dynamic)**

**- lifetime of a variable**

The scope of any variable is the area of the program where the use of x is its declaration.

Scoping is of two classes: 1.Static Scoping & 2.Dynamic Scoping

Static Scoping :Static Scoping or lexical scoping makes it a lot simpler to create a measured code as the developer can sort out the programmer just by taking a gander at the code.

Dynamic Scoping : With dynamic scope, each identifier has a global stack of bindings and the event of an identifier is searched in the latest binding.

The relationship of technique call to the strategy body is known as binding. There are two sorts of binding: Static Binding and Dynamic Binding.

Static restricting refers to the execution of the program where the object type is resolved during the compiling time.

Dynamic Binding is the execution of the program where the object type is resolved during the runtime.

The lifetime of a variable is the time during which the variable stays in memory and is therefore accessible during program execution.