**Name:** Satyam Jaiswal **UID:** 2021600028 **Batch**: B **Date:** 05/11/2023

# **EXP 8:** Prolog Programming Knowledge and reasoning - II.

**Problem:** Huffman Code We suppose a set of symbols with their frequencies, given as a list of fr(S,F) terms. Example:[fr(a,45),fr(b,13),fr(c,12),fr(d,16),fr(e,9),fr(f,5)]. Our objective is to construct a list hc(S,C) terms, where C is the Huffman code word for the symbol S.In our example, the result could be

Hs =[hc(a,'0'), hc(b,'101'), hc(c,'100'), hc(d,'111'), hc(e,'1101'),

hc(f,'1100')] [hc(a,'01'),...etc.].

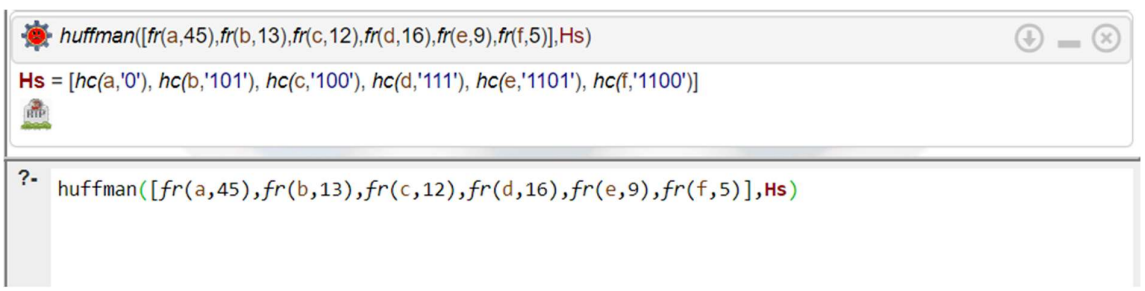
The task shall be performed by the predicate huffman/2 defined as follows: % huffman(Fs,Hs) :- Hs is the Huffman code table for the frequency table Fs. For more information check :- https://en.wikipedia.org/wiki/Huffman_coding

**Program:**

```
huffman(Fs,Cs) :-

initialize(Fs,Ns),

make_tree(Ns,T),

traverse_tree(T,Cs).

initialize(Fs,Ns) :- init(Fs,NsU), sort(NsU,Ns).

init([],[]).

init([fr(S,F)|Fs],[n(F,S)|Ns]) :- init(Fs,Ns).

make_tree([T],T).

make_tree([n(F1,X1),n(F2,X2)|Ns],T) :-

F is F1+F2,

insert(n(F,s(n(F1,X1),n(F2,X2))),Ns,NsR),

make_tree(NsR,T).

insert(N,[],[N]) :- !.

insert(n(F,X),[n(F0,Y)|Ns],[n(F,X),n(F0,Y)|Ns]) :- F < F0, !.

insert(n(F,X),[n(F0,Y)|Ns],[n(F0,Y)|Ns1]) :- F >= F0, insert(n(F,X),Ns,Ns1).

traverse_tree(T,Cs) :- traverse_tree(T,'',Cs1-[]), sort(Cs1,Cs).

traverse_tree(n(_,A),Code,[hc(A,Code)|Cs]-Cs) :- atom(A).

traverse_tree(n(_,s(Left,Right)),Code,Cs1-Cs3) :-

atom_concat(Code,'0',CodeLeft),
```

```
atom_concat(Code,'1',CodeRight),
traverse_tree(Left,CodeLeft,Cs1-Cs2),
traverse_tree(Right,CodeRight,Cs2-Cs3).
huffman(Fs) :- huffman(Fs,Hs) , nl, report(Hs,5), stats(Fs,Hs).
report([],_) :- !, nl, nl.
report(Hs,0) :- !, nl, report(Hs,5).
report([hc(S,C)|Hs],N) :- N > 0, N1 is N-1,
writef('%w %8l ',[S,C]), report(Hs,N1).
stats(Fs,Cs) :- sort(Fs,FsS), sort(Cs,CsS), stats(FsS,CsS,0,0).
stats([],[],FreqCodeSum,FreqSum) :- Avg is FreqCodeSum/FreqSum,
writef('Average code length (weighted) = %w\n',[Avg]).
stats([fr(S,F)|Fs],[hc(S,C)|Hs],FCS,FS) :-
atom_chars(C,CharList), length(CharList,N),
FCS1 is FCS + F*N, FS1 is FS + F,
 stats(Fs,Hs,FCS1,FS1).
```

**Output:**




**Conclusion:**

1) Learnt about prolog and how it is used in programming and how functions are implemented in prolog.
2) In conclusion, The Prolog code we wrote is functional implementation of the Huffman coding algorithm, which can be used to compress data efficiently by assigning variable-length codes to symbols based on their frequencies.