Seminar 3 – Liste eterogene in Prolog

%verif\_muntecresc(L:lista liniara)

verif\_muntecresc([H1,H2|T]):-

H1<H2,

verif\_muntecresc([H2,T]).

verif\_muntecresc([H1,H2|T]):-H1>H2,

verif\_munte\_descresc([H2|T]).

verif\_munte\_descresc([\_]).

verif\_munte\_descresc([H1,H2|T]):-

H1>H2,

verif\_munte\_descresc([H2|T]).

verif\_munte([H1,H2,H3|T]):-

H1<H2,

verif\_munte\_cresc([H1,H2,H3|T]).

verif(l1…n, f)=

true, n = 1 si f = 0

fals, n = 1 si f != 0

fals, n = 0

verif(l2…n,1), l1<l2, n>=2, f=-1

verif(l2…n,1),l1<l2, n>=2, f=1

verif(l2…n,0),l1>l2,n>=2, f=1

verif(l2…n,0),l1>l2,n>=2, f=0

fals, altfel

%verif(L:lista, F:intreg)

%model flux (i,i)

verif([\_],0).

verif([H1,H2|T],F):-

H1<H2,

F\=0,

verif([H2|T],1).

verif([H1,H2|T],F):-

H1>H2,

F>=-1,

verif([H2|T],0).