GRAMMATICA:

Quando nel follow hai un epsilon, devi guardare il first delle produzioni che seguono il non-terminale

ESEMPIO DI FOLLOW:

1. Follow (Exp) =

end,then,else,First(X), First(Seq\_exp)

1. Follow(Exp) =

end,then,else, (and, Follow(Bind)),(First(Exp), Follow(Seq\_Exp))

1. Follow(Exp) =

end, then,else,(and, in), (First(Prog),lambda,First(ExpA), First(OPP), if, Follow(Seq\_Exp))

1. Follow(Exp) =

end,then,else, (and, in), ((let, letrec), lambda,(First(T)), (cons,car,cdr,eq,leq,atom),if, Follow(Seq\_Exp))

1. Follow(Exp) =

end, then,else,(and, in), ((let, letrec), lambda,(First(F)), (cons,car,cdr,eq,leq,atom),if, Follow(Seq\_Exp))

1. Follow(Exp) =

end,then,else, (and, in), ((let, letrec), lambda,(First(F)), (cons,car,cdr,eq,leq,atom),if, Follow(Seq\_Exp))

1. Follow(Exp) =

end, then,else,(and, in), ((let, letrec), lambda,( var, exp\_const, ( ), (cons,car,cdr,eq,leq,atom),if, Follow(Seq\_Exp))

1. Follow(Exp) =

end, then,else,(and, in), ((let, letrec), lambda,( var, exp\_const, ( ), (cons,car,cdr,eq,leq,atom),if, ))

TABELLA:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Non terminale** | **First** | **Follow** | **First finale** | **Follow finale** |
|  | var, epsilon | ) | var,epsilon | ) |
|  | First(Exp), epsilon | ) | var, let, letrec, lambda, exp\_const, (, cons, car, cdr, eq, leq, atom, if, epsilon | ) |
|  | cons, car, cdr, eq, leq, atom | ( | cons, car, cdr, eq, leq, atom | ( |
|  | \*, / | First(F) | \*, / | var, exp\_const, ( |
|  | +, - | First(T) | +, - | var, exp\_const ( |
|  | (, epsilon | Follow(F) | (, epsilon | \*,/,+,-,),letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, |
|  | var, exp\_const, ( | { First(T1) }  U  { Follow(T) } | var, exp\_const, ( | \*,/,+,-,),letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, |
|  | First(OPM), epsilon | Follow(T) | \*, /, epsilon | +,-,),letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, |
|  | First(F) | { Fisrt(E1) }  U  { Follow(E1) } | var, exp\_const, ( | +,-,),letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, |
|  | First(OPA) | Follow(ExpA) | +, -, epsilon | ),end,and,then, else, in,letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, |
|  | First(T) | { ) }  U  { Follow(Exp) } | var, exp\_const, ( | ),end,and,then, else, in,letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, |
|  | First(Prog) U {lambda} U  First(ExpA) U First(OPP) U {if} | end, and, then, else, in, let, letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, ) | let, letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, ) | end,and,then, else, in, let, letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, ) |
|  | and, epsilon | Follow(Bind) | and, epsilon | in |
|  | var | in | var | in |
|  | let, letrec | { $ }  U  { Follow(Exp) } | let, letrec | $, end,and,then, else, in, let, letrec, lambda, var, exp\_const, ( , cons,car,cdr,eq,leq,atom,if, ) |