€ == Epsilon

Quitada recursividad por la izquierda

1. S -> **program id pyc** Vsp Bloque
2. Vsp -> Unsp Vsp_prima
3. Vsp_prima -> Unsp Vsp_prima
4. Vsp_prima -> €
5. Unsp -> **function id dosp** Tipo **pyc** Vsp Bloque **pyc**
6. Unsp -> **var** LV
7. LV -> V LV _prima
8. LV _prima -> V LV _prima
9. LV _prima -> €
10. V -> **id** Lid **dosp** Tipo **pyc**
11. Lid -> **coma id** Lid
12. Lid -> €
13. Tipo -> **integer**
14. Tipo ->**real**
15. Bloque -> **begin** SInstr **end**
16. SInstr -> Instr SInstrp
17. SInstrp -> **pyc** Instr SInstrp
18. SInstrp -> €
19. Instr -> Bloque
20. Instr -> **id asig** E
21. Instr -> **if** E **then** Instr **endif**
22. Instr -> **if** E **then** Instr **else** Instr **endif**
23. Instr -> **while** E **do** Instr
24. Instr -> **writeln pari** E **pard**
25. E -> Expr **relop** Expr
26. E -> Expr
27. Expr -> Term Expr_prima
28. Expr_prima -> **addop** Term Expr_prima
29. Expr_prima -> €
30. Term -> Factor Term_prima
31. Term_prima -> **mulop** Factor Term_prima
32. Term_prima -> €
33. Factor -> **id**
34. Factor -> **nentero**
35. Factor -> **nreal**
36. Factor -> **pari** Expr **pard**

Quitados factores comunes

1. S -> **program id pyc** Vsp Bloque
2. Vsp -> Unsp Vsp_prima
3. Vsp_prima -> Unsp Vsp_prima
4. Vsp_prima -> €
5. Unsp -> **function id dosp** Tipo **pyc** Vsp Bloque **pyc**
6. Unsp -> **var** LV
7. LV -> V LV_prima
8. LV_prima -> V LV_prima
9. LV_prima -> €
10. V -> **id** Lid **dosp** Tipo **pyc**
11. Lid -> **coma id** Lid
12. Lid -> €
13. Tipo -> **integer**
14. Tipo -> **real**
15. Bloque -> **begin** SInstr **end**
16. SInstr -> Instr SInstrp
17. SInstrp -> **pyc** Instr SInstrp
18. SInstrp -> €
19. Instr -> Bloque
20. Instr -> **id asig** E
21. Instr -> **if** E **then** Instr Instr_prima
22. Instr_prima -> **endif**
23. Instr_prima -> **else** Instr **endif**
24. Instr -> **while** E **do** Instr
25. Instr -> **writeln pari** E **pard**
26. E -> Expr E_prima
27. E_prima -> **relop** Expr
28. E_prima -> €
29. Expr -> Term Expr_prima
30. Expr_prima -> **addop** Term Expr_prima
31. Expr_prima -> €
32. Term -> Factor Term_prima
33. Term_prima -> **mulop** Factor Term_prima
34. Term_prima -> €
35. Factor -> **id**
36. Factor -> **nentero**
37. Factor -> **nreal**
38. Factor -> **pari** Expr **pard**

Conjuntos de predicción

PRIMEROS

PRIMEROS(S) = {program}
PRIMEROS(Vsp) = {PRIMEROS(Unsp)} = {function,var}
PRIMEROS(Vsp_prima) = {PRIMEROS(Unsp) U €} = {function, var, €}
PRIMEROS(Unsp) = {function, var}
PRIMEROS(LV) = {Primeros(V)} = {id}
PRIMEROS(LV_prima) = {PRIMEROS(V) U €} = {id, €}
PRIMEROS(V) = {id}
PRIMEROS(Lid) = {coma, €}
PRIMEROS(Tipo) = {real,integer}
PRIMEROS(Bloque) = {begin}
PRIMEROS(SInstr) = {PRIMEROS(Instr)} = {begin, id, if,while, writeln}
PRIMEROS(SInstrp) = {pyc, €}
PRIMEROS(Instr) = {PRIMEROS(BLOQUE) U {id, if, while, writeln}} = {begin, id, if, while, writeln}
PRIMEROS(Instr_prima) = {endif, else}
PRIMEROS(E) = {PRIMEROS(Expr)} = {id, nentero, nreal, pari}
PRIMEROS(E_prima) = {relop, €}
PRIMEROS(Expr) = { PRIMEROS(Term)} = {id, nentero, nreal, pari}
PRIMEROS(Expr_prima) = {addop, €}
PRIMEROS(Term) = {PRIMEROS(Factor)-{€}} = {id, nentero, nreal, pari}
PRIMEROS(Term_prima) = {mulop, €}
PRIMEROS(Factor) = {id, nentero, nreal, pari}

SIGUIENTES

SIGUIENTES(S) = {$}

SIGUIENTES(Vsp) = {PRIM(Bloque) - {€}} = {begin}

SIGUIENTES(Vsp_prima) = {SIGUIENTES(Vsp)} = {begin}

SIGUIENTES(Unsp) = {PRIMEROS(Vsp_prima) - {€} U SIGUIENTES(Vsp_prima)} = {function, var, begin}

SIGUIENTES(LV) = {SIGUIENTES(Unsp)} = {function, var, begin}

SIGUIENTES(LV_prima) = {SIGUIENTES(LV)} = {function, var, begin}

SIGUIENTES(V) = {PRIMEROS(LV_prima) - {€} U SIGUIENTES(LV_Prima)} = {id, function, var, begin}

SIGUIENTES(Lid) = {dosp}

SIGUIENTES(Tipo) = {pyc}

SIGUIENTES(Bloque) = {SIGUIENTES(S) U SIGUIENTES(Instr) U pyc} = {$,pyc,endif, else,end}

SIGUIENTES(SInstr) = {end}

SIGUIENTES(SInstrp) = {SIGUIENTES(SInstr)} = {end}

SIGUIENTES(Instr) = {PRIMEROS(SInstrp) - {€} U PRIMEROS(Instr_prima) - {€} U endif U SIGUIENTES(SInstrp)} = {pyc, endif, else, end}

SIGUIENTES(Instr_prima) = {SIGUIENTES(Instr)} = {pyc, endif, else, end}

SIGUIENTES(E) = {SIGUIENTES(Instr) U {then, do, pard}} = {pyc, endif, else, end, then, do, pard}

SIGUIENTES(E_prima) = {Siguientes(E)} = {pyc,endif,else,end,then,do,pard}

SIGUIENTES(Expr) = {Primeros(E_prima) - {€} U Siguientes(E) U Siguientes(E_prima) U {pard} } = {relop, pyc, endif, else, end, then, do, pard}

SIGUIENTES(Expr_prima) = {SIGUIENTES(Expr)} = {relop, pyc, endif, else, end, then, do, pard}

SIGUIENTES(Term) = {PRIMEROS(Expr_prima) - {€} U SIGUIENTES(Expr) U SIGUIENTES(Expr_Prima)} = {addop, relop, pyc, endif, else, end, then, do, pard}

SIGUIENTES(Term_prima) = {SIGUIENTES(Term)} = {addop, relop, pyc, endif, else, end, then, do, pard}

SIGUIENTES(Factor) = {PRIMEROS(Term_prima) - {€} U SIGUIENTES(Term_prima) U SIGUIENTES(Term)} = {mulop, addop, relop, pyc, endif, else, end, then, do, pard}

Conjuntos de predicción

PRED(S -> **program id pyc** Vsp Bloque) = {program}
PRED(Vsp -> Unsp Vsp_prima) = {PRIM(Unsp)} = {function,var}
PRED(Vsp_prima -> Unsp Vsp_prima) = {PRIM(Unsp)} = {function, var}
PRED(Vsp_prima -> €) = {{PRIM(€) - {€} U SIGUIENTES(Vsp_prima)} = {begin}
PRED(Unsp -> **function id dosp** Tipo **pyc** Vsp Bloque **pyc**) = {function}
PRED(Unsp -> **var** LV) = {var}
PRED(LV -> V LV_prima) = {PRIM(V)} = {id}
PRED(LV_prima -> V LV_prima) = {PRIM(V)} = {id}
PRED(LV_prima -> €) = {PRIM(€) -{€} U SIGUIENTES(LV_PRIMA} = {fuction, var, begin}
PRED(V -> **id** Lid **dosp** Tipo **pyc**) = {id}
PRED(Lid -> **coma id** Lid) = {coma}
PRED(Lid -> €) = {PRIM(€) -{€} U SIGUIENTES(Lid)} = {dosp}
PRED(Tipo -> **integer**) = {integer}
PRED(Tipo -> **real**) = {real}
PRED(Bloque -> **begin** SInstr **end**) = {begin}
PRED(SInstr -> Instr SInstrp) = {PRIM(Instr)} = {begin, id, if, while, writeln}
PRED(SInstrp -> **pyc** Instr SInstrp) = {pyc}
PRED(SInstrp -> €) = {PRIM(€) -{€} U SIGUIENTES(SImstrp)} = {end}
PRED(Instr -> Bloque) = {PRIM(Bloque)} = {begin}
PRED(Instr -> **id asig** E) = {id}
PRED(Instr -> **if** E **then** Instr Instr_prima) = {if}
PRED(Instr_prima -> **endif**) = {endif}
PRED(Instr_prima -> **else** Instr **endif**) = {else}
PRED(Instr -> **while** E **do** Instr) = {while}
PRED(Instr -> **writeln pari** E **pard**) = {writeln}
PRED(E -> Expr E_prima) = {PRIM(Expr)} = {id, nentero, nreal, pari}
PRED(E_prima -> **relop** Expr) = {relop}
PRED(E_prima -> €) = {PRIM(€) -{€} U SIGUIENTES(E_PRIMA)} =
{pyc,endif,else,end,then,do,pard}
PRED(Expr -> Term Expr_prima) = {PRIM(Term)} = {id, nentero, nreal, pari}
PRED(Expr_prima -> **addop** Term Expr_prima) = {addop}
PRED(Expr_prima -> €) = {PRIM(€) -{€} U SIGUIENTES(Expr_PRIMA)} = {relop, pyc,
endif, else, end, then, do, pard}
PRED(Term -> Factor Term_prima) = {PRIM(Factor)} = {id, nentero, nreal, pari}
PRED(Term_prima -> **mulop** Factor Term_prima) = {mulop}
PRED(Term_prima -> €) = {PRIM(€) -{€} U SIGUIENTES(Term_PRIMA)} = {addop, relop,
pyc, endif, else, end, then, do, pard}
PRED(Factor -> **id**) = {id}
PRED(Factor -> **nentero**) = {nentero}

PRED(Factor -> **nreal**) = {nreal}
PRED(Factor -> **pari** Expr **pard**) = {pari}