New grammar after Elimination left recursion and common left prefixes

PROGARAM 🡪 program DECLARATIONS ; STATMENTS end

DECLARATIONS 🡪 DECLARATION DECLARATIONS\_A

DECLARATIONS\_A 🡪 epsilon

DECLARATIONS\_A 🡪 ; DECLARATIONS

DECLARATION 🡪 VARIABLE\_LIST : TYPE

TYPE 🡪 real | integer

VARIABLE\_LIST 🡪 V VARIABLE\_LIST\_A

VARIABLE\_LIST \_A🡪 epsilon

VARIABLE\_LIST \_A🡪 , VARIABLE VARIABLE\_LIST\_A

VARIABLE 🡪 id VARIABLE \_A

VARIABLE \_A🡪 epsilon

VARIABLE \_A🡪 [int\_number]

STATMENTS 🡪 STATMENT STATEMENTS\_A

STATEMENTS\_A 🡪 epsilon

STATEMENTS\_A 🡪 ; STATMENTS

STATMENT 🡪RECIEVER= EXPRESSION

STATMENT 🡪 if CONDITION then STATMENTS else STATMENTS end\_if

STATMENT 🡪 loop STATMENTS until C end\_loop

STATMENT 🡪 start DECLARATIONS ; STATMENTS end

RECEIVER 🡪 id RECEIVER1

RECEIVER\_A 🡪 epsilon

RECEIVER\_A 🡪 [EXPRESSION]

EXPRESSION 🡪 number

EXPRESSION 🡪 id EXPRESSION1

EXPRESSION\_A 🡪 epsilon

EXPRESSION\_A 🡪 [EXPRESSION]

EXPRESSION\_A 🡪 ar\_op EXPRESSION

CONDITION 🡪EXPRESSION rel\_op EXPRESSION

Table of Nullable, First, Follow

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nullable | First | Follow |
| PROGARAM |  | Program | $ |
| DECLARATIONS |  | Id | ; |
| STATEMENTS |  | If Loop Start id | End Else End\_if  Until |
| DECLARATION |  | Id | ; |
| DECLARATIONS\_A | Yes | ; | ; |
| VARIABLE\_LIST |  | Id | : |
| TYPE |  | Real Integer | ; |
| VARIABLE |  | Id | , : |
| VARIABLE\_LIST\_A | Yes | , | : |
| VARIABLE\_A | Yes | [ | , : |
| STATEMENT |  | If Loop Start Id | ; End Else End\_if  Until |
| STATEMENTS\_A | Yes | ; | End Else End\_if  Until |
| RECEIVER |  | Id | = |
| EXPRESSION |  | Number Id | ] rel\_op ; Then  End\_loop End  Else End\_if Until |
| CONDITION |  | Number Id | Then End\_loop |
| RECEIVER\_A | Yes | [ | = |
| EXPRESSION\_A | yes | [ ar\_op | ] rel\_op ; Then  End\_loop End  Else End\_if Until |