|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prod # | Variable | First | Follows | Follows |
| 1.1 | *program* | **program** | **{$}** | **{$}** |
| 1.2.1 | *program’* | **procedure** | Follow(program) | **{$}** |
| 1.2.2 | *program’* | **begin** | Follow(program) | **{$}** |
| 1.2.3 | *program’* | **var** | Follow(*program)* | **{$}** |
| 1.3.1 | *program’’* | **procedure** | Follow(*program’)* | **{$}** |
| 1.3.2 | *program’’* | **begin** | Follow(*program’)* | **{$}** |
| 2.1 | *idlst* | **id** | **{)}** | **{)}** |
| 2.2.1 | *idlst’* | **,** | Follow(*idlst) U* Follow(*idlst’)* | **{)}** |
| 2.2.2 | *idlst’* | ϵ | Follow(*idlst) U* Follow(*idlst’)* | **{)}** |
| 3.1 | *declarations* | **var** | **{procedure, begin}** | **{procedure, begin}** |
| 3.2.1 | *declarations’* | **var** | Follow(*declarations) U Follow(declarations’)* | **{procedure, begin}** |
| 3.2.2 | *declarations’* | ϵ | Follow(*declarations) U Follow(declarations’)* | **{procedure, begin}** |
| 4.1 | *type* | **integer real** | **{;}** U Follow(*parameter\_list’*) | **{;, )}** |
| 4.2 | *type* | **array** | **{;}** U Follow(*parameter\_list’*) | **{;, )}** |
| 5.1 | *standard\_type* | **integer** | Follow(*type*) | **{;, )}** |
| 5.2 | *standard\_type* | **real** | Follow(*type*) | **{;, )}** |
| 6.1 | *subdeclarations* | **procedure** | **{begin}** | **{begin}** |
| 6.2.1 | *subdeclarations’* | **procedure** | Follow(*subdeclarations*) U Follow(*subdeclarations’*) | **{begin}** |
| 6.2.2 | *subdeclarations’* | ϵ | Follow(*subdeclarations*) U Follow(*subdeclarations’*) | **{begin}** |
| 7.1 | *subdeclaration* | **procedure** | **{;}** | **{;}** |
| 7.2.1 | *subdeclaration’* | **procedure** | Follow(*subdeclaration*) | **{;}** |
| 7.2.2 | *subdeclaration’* | **begin** | Follow(*subdeclaration*) | **{;}** |
| 7.2.3 | *subdeclaration’* | **var** | Follow(*subdeclaration*) | **{;}** |
| 7.3.1 | *subdeclaration’’* | **procedure** | Follow(*subdeclaration’*) | **{;}** |
| 7.3.2 | *subdeclaration’’* | **begin** | Follow(*subdeclaration’*) | **{;}** |
| 8.1 | *subprogram\_head* | **procedure** | **{procedure, begin, var}** | **{procedure, begin, var}** |
| 8.2.1 | *subprogram\_head’* | **;** | Follow(*subprogram\_head*) | **{procedure, begin, var}** |
| 8.2.2 | *subprogram\_head’* | **(** | Follow(*subprogram\_head* | **{procedure, begin, var}** |
| 9.1 | *arguments* | **(** | **{;}** | **{;}** |
| 10.1 | *parameter\_list* | **id** | **{)}** | **{)}** |
| 10.2.1 | *parameter\_list’* | **;** | Follow(*parameter\_list) U Follow(parameter\_list’)* | **{)}** |
| 10.2.2 | *parameter\_list’* | ϵ | Follow(*parameter\_list) U Follow(parameter\_list’)* | **{)}** |
| 11.1 | *compound\_statement* | **begin** | **{.}** U Follow(*subdeclaration’) U* Follow(*subdeclaration’’) U Follow(statement)* | **{., ;, end, else}** |
| 11.2.1 | *compound\_statement’* | **id call begin while if** | Follow(*compound\_statement)* | **{., ;, end, else}** |
| 11.2.2 | *compound\_statement’* | **end** | Follow(*compound\_statement)* | **{., ;, end, else}** |
| 12.1 | *optional\_statements* | **id call begin while if** | **{end}** | **{end}** |
| 13.1 | *statement\_list* | **id call begin while if** | Follow(*optional\_statements)* | **{end}** |
| 13.2.1 | *statement\_list’* | **;** | Follow(*statement\_list) U* Follow(*statement\_list’)* | **{end}** |
| 13.2.2 | *statement\_list’* | ϵ | Follow(*statement\_list) U* Follow(*statement\_list’)* | **{end}** |
| 14.1 | *statement* | **id** | **{;, else}** U Follow(*statement\_list) U* Follow(*statement) U* Follow(*statement’)* | **{;, end, else}** |
| 14.2 | *statement* | **call** | **{;, else}** U Follow(*statement\_list) U* Follow(*statement) U* Follow(*statement’)* | **{;, end, else}** |
| 14.3 | *statement* | **begin** | **{;, else}** U Follow(*statement\_list) U* Follow(*statement) U* Follow(*statement’)* | **{;, end, else}** |
| 14.4 | *statement* | **while** | **{;, else}** U Follow(*statement\_list) U* Follow(*statement) U* Follow(*statement’)* | **{;, end, else}** |
| 14.5 | *statement* | **if** | **{;, else}** U Follow(*statement\_list) U* Follow(*statement) U* Follow(*statement’)* | **{;, end, else}** |
| 14.6.1 | *statement’* | **else** | Follow(*statement)* | **{;, end, else}** |
| 14.6.2 | *statement’* | ϵ | Follow(*statement)* | **{;, end, else}** |
| 15.1 | *variable* | **id** | **{assignop}** | **{assignop}** |
| 15.2.1 | *variable’* | **[** | Follow(*variable*) | **{assignop}** |
| 15.2.2 | *variable’* | ϵ | Follow(*variable*) | **{assignop}** |
| 16.1 | *procedure\_statement* | **call** | Follow(s*tatement*) | **{;, end, else}** |
| 16.2.1 | *procedure\_statement’* | **(** | Follow(*procedure\_*s*tatement*) | **{;, end, else}** |
| 16.2.2 | *procedure\_statement’* | ϵ | Follow(*procedure\_*s*tatement*) | **{;, end, else}** |
| 17.1 | *expression\_list* | **id num ( not + -** | **{)}** | **{)}** |
| 17.2.1 | *expression\_list’* | **,** | Follow(*expression\_list*) U Follow(*expression\_list’*) | **{)}** |
| 17.2.2 | *expression\_list’* | ϵ | Follow(*expression\_list*) U Follow(*expression\_list’* | **{)}** |
| 18.1 | *expression* | **id num ( not + -** | **{], do, then,** **‘,’,)}** UFollow(*statement)* U Follow(*expression\_list’*) | **{], do, then,** **‘,’, ), ;, end, else}** |
| 18.2.1 | *expression’* | **relop** | Follow(*expression*) | **{], do, then,** **‘,’, ), ;, end, else}** |
| 18.2.2 | *expression’* | ϵ | Follow(*expression*) | **{], do, then,** **‘,’, ), ;, end, else}** |
| 19.1 | *simple\_expression* | **id num ( not** | **{relop}** U Follow(*expression’)* | **{relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 19.2 | *simple\_expression* | **+ -** | **{relop}** U Follow(*expression’)* | **{relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 19.3.1 | *simple\_expression’* | **addop** | Follow(*simple\_expression’) U* Follow(*simple\_expression)* | **{relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 19.3.2 | *simple\_expression’* | ϵ | Follow(*simple\_expression’) U* Follow(*simple\_expression)* | **{relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 20.1 | *term* | **id num ( not** | **{addop}** U Follow(*simple\_expression’)* | **{addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 20.2.1 | *term’* | **mulop** | Follow(*term’) U* Follow(*term)* | **{addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 20.2.2 | *term’* | ϵ | Follow(*term’) U* Follow(*term)* | **{addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 21.1 | *factor* | **id** | **{mulop}**  U Follow(term) U Follow(*term’) U* Follow(*factor)* | **{mulop, addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 21.2 | *factor* | **num** | **{mulop}**  U Follow(term) U Follow(*term’) U* Follow(*factor)* | **{mulop, addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 21.3 | *factor* | **(** | **{mulop}**  U Follow(term) U Follow(*term’) U* Follow(*factor)* | **{mulop, addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 21.4 | *factor* | **not** | **{mulop}**  U Follow(term) U Follow(*term’) U* Follow(*factor)* | **{mulop, addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 21.5.1 | *factor’* | **[** | Follow(*factor)* | **{mulop, addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 21.5.2 | *factor’* | ϵ | Follow(*factor)* | **{mulop, addop, relop, ], do, then,** **‘,’, ), ;, end, else}** |
| 22.1 | *sign* | **+** | **{id, num, (, not}** | **{id, num, (, not}** |
| 22.2 | *sign* | **-** | **{id, num, (, not}** | **{id, num, (, not}** |