1. The grammar which describe the mini program language that we've got has some left recursion and common left prefixes which we need to eliminate before coding the parser. The corrections are:

* VARIABLES\_LIST has a left recursion.
  + Correction:   
    VARIABLE\_LIST -> VARIABLE VARIABLE\_LIST'  
    VARIABLE\_LIST' -> ,VARIABLE VARIABLE\_LIST |
* VARIABLE has a common left prefixes.
  + Correction:   
    VARIABLE -> id VARIABLE'.

VARIABLE' -> [int\_number] |

* FUNC\_DEFINITIONS has a left recursion.
  + Correction:   
    FUNC\_DEFINITIONS -> FUNC\_DEFINITION FUNC\_DEFINITIONS'.

FUNC\_DEFINITIONS' -> FUNC\_DEFINITION FUNC\_DEFINITION' |

* STATEMENT has a common left prefixes.
  + Correction:  
    STATEMENT -> VARIABLE=EXPRESSION | BLOCK | return STATEMENT' | FUNCTION\_CALL

STATEMENT' -> EXPRESSION |

* EXPRESSION has a common left prefixes.
  + Correction:   
    EXPRESSION -> int\_number | real\_number | id EXPRESSION'

EXPRESSION' -> VARIABLE' | ar\_op EXPRESSION

Therefore, the fixed grammar which suitable for LL(1) parser is:

PROGRAM -> program VAR\_DEFINITIONS; STATEMENTS end FUNC\_DEFINITIONS

DEFINITIONS -> VAR\_DEFINITIONS; FUNC\_DEFINITION.

VAR\_DEFINITIONS -> VAR\_DEFINITION | VAR\_DEFINITION; VAR\_DEFINITIONS

VAR\_DEFINITION -> TYPE VARIABLES\_LIST.

TYPE -> real | integer.

VARIABLE\_LIST -> VARIABLE VARIABLE\_LIST'

VARIABLE\_LIST' -> ,VARIABLE VARIABLE\_LIST |

VARIABLE -> id VARIABLE'.

VARIABLE' -> [int\_number] |

FUNC\_DEFINITIONS -> FUNC\_DEFINITION FUNC\_DEFINITIONS'

FUNC\_DEFINITIONS' -> FUNC\_DEFINITION FUNC\_DEFINITION' |

FUNC\_DEFINITION -> RETURNED\_TYPE id (PARAM\_DEFINITIONS)

BLOCK

RETURNED\_TYPE -> void | TYPE

PARAM\_DEFINITIONS -> VAR\_DEFINITIONS |

STATEMENTS -> STATEMENT; | STATEMENT; STATEMENTS

STATEMENT -> VARIABLE=EXPRESSION | BLOCK | return STATEMENT' | FUNCTION\_CALL

STATEMENT' -> EXPRESSION |

BLOCK -> { VAR\_DEFINITIONS; STATEMENTS }

FUNCTION\_CALL -> id (PARAMETERS\_LIST)

PARAMETERS\_LIST -> VARIABLES\_LIST |

EXPRESSION -> int\_number | real\_number | id EXPRESSION'

EXPRESSION' -> VARIABLE' | ar\_op EXPRESSION

1. Calculation of attributes Nullable, First and Follow.

|  |  |  |  |
| --- | --- | --- | --- |
| **The variable / attribute** | **Nullable** | **First** | **Follow** |
| **PROGRAM** | - | {**program**} | {EOF} |
| **DEFINITIONS** | - |  |  |
| **VAR\_DEFINITIONS** | - | {**real**, **integer**} |  |
| **VAR\_DEFINITION** | - | {**real**, **integer**} |  |
| **TYPE** | - | {**real**, **integer**} | {id} |
| **VARIABLE\_LIST** | - | {id} |  |
| **VARIABLE\_LIST'** | + | {id} |  |
| **VARIABLE** | - | {id} | {,} |
| **VARIABLE'** | + | {[int\_number]} | {,} |
| **FUNC\_DEFINITIONS** | - | {**void**, **real**, **integer**} | {**void**, **real**, **integer**} |
| **FUNC\_DEFINITIONS'** | - |  |  |
| **FUNC\_DEFINITION** | + | { **void**, **real**, **integer**} |  |
| **RETURNED\_TYPE** | - | { **void**, **real**, **integer**} | {id} |
| **PARAM\_DEFINITIONS** | + | {**real**, **integer**} |  |
| **STATEMENTS** | - |  | {**void**, **real**, **integer**, **end**} |
| **STATEMENT** | - | {**return**, id} |  |
| **STATEMENT'** | + | {int\_number, real\_number, id} |  |
| **BLOCK** | - |  | {**void**, **real**, **integer**} |
| **FUNCTION\_CALL** | - | {id} |  |
| **PARAMETERS\_LIST** | + |  |  |
| **EXPRESSION** | - | {int\_number, real\_number, id} |  |
| **EXPRESSION'** | + | {ar\_op, int\_number} |  |

Parsing table:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **program** | **end** | **real** | **integer** | **void** | **return** | **Int\_number** | **real\_number** | **id** | **ar\_op** | **EOF** |
| **The variable** |  |  |  |  |  |  |  |  |  |  |  |
| **PROGRAM** |  |  |  |  |  |  |  |  |  |  |  |
| **DEFINITIONS** |  |  |  |  |  |  |  |  |  |  |  |
| **VAR\_DEFINITIONS** |  |  |  |  |  |  |  |  |  |  |  |
| **VAR\_DEFINITION** |  |  |  |  |  |  |  |  |  |  |  |
| **TYPE** |  |  |  |  |  |  |  |  |  |  |  |
| **VARIABLE\_LIST** |  |  |  |  |  |  |  |  |  |  |  |
| **VARIABLE\_LIST'** |  |  |  |  |  |  |  |  |  |  |  |
| **VARIABLE** |  |  |  |  |  |  |  |  |  |  |  |
| **VARIABLE'** |  |  |  |  |  |  |  |  |  |  |  |
| **FUNC\_DEFINITIONS** |  |  |  |  |  |  |  |  |  |  |  |
| **FUNC\_DEFINITIONS'** |  |  |  |  |  |  |  |  |  |  |  |
| **FUNC\_DEFINITION** |  |  |  |  |  |  |  |  |  |  |  |
| **RETURNED\_TYPE** |  |  |  |  |  |  |  |  |  |  |  |
| **PARAM\_DEFINITIONS** |  |  |  |  |  |  |  |  |  |  |  |
| **STATEMENTS** |  |  |  |  |  |  |  |  |  |  |  |
| **STATEMENT** |  |  |  |  |  |  |  |  |  |  |  |
| **STATEMENT'** |  |  |  |  |  |  |  |  |  |  |  |
| **BLOCK** |  |  |  |  |  |  |  |  |  |  |  |
| **FUNCTION\_CALL** |  |  |  |  |  |  |  |  |  |  |  |
| **PARAMETERS\_LIST** |  |  |  |  |  |  |  |  |  |  |  |
| **EXPRESSION** |  |  |  |  |  |  |  |  |  |  |  |
| **EXPRESSION'** |  |  |  |  |  |  |  |  |  |  |  |