GROUP 20

Amit Bansal	**2016A7PS0140P**
Vedant Patwary	**2016A7PS0031P**
Abhimanyu Singh Shekhawat	**2016A7PS0112P**
Abhilash Neog	**2016A7PS0004P**

Grammar Rules

1. program ⇒ otherFunctions mainFunction
2. mainFunction ⇒TK_MAIN stmts TK_END
3. otherFunctions⇒ function otherFunctions
4. otherFunctions ⇒eps
5. function ⇒ TK_FUNID input_par output_par TK_SEM stmts TK_END
6. input_par ⇒TK_INPUT TK_PARAMETER TK_LIST TK_SQL parameter_list TK_SQR
7. output_par ⇒ TK_OUTPUT TK_PARAMETER TK_LIST TK_SQL parameter_list TK_SQR
8. Output_par ⇒ eps
9. parameter_list ⇒ dataType TK_ID remaining_list
10. dataType ⇒ primitiveDatatype
11. dataType ⇒ constructedDatatype
12. primitiveDatatype ⇒ TK_INT
13. primitiveDatatype ⇒ TK_REAL
14. constructedDatatype ⇒ TK_RECORD TK_RECORDID
15. remaining_list ⇒TK_COMMA parameter_list
16. remaining_list⇒ eps
17. stmts ⇒ typeDefinitions declarations otherStmts returnStmt
18. typeDefinitions ⇒ typeDefinition typeDefinitions
19. typeDefinitions ⇒ eps
20. typeDefinition ⇒ TK_RECORD TK_RECORDID fieldDefinitions TK_ENDRECORD TK_SEM
21. fieldDefinitions ⇒ fieldDefinition fieldDefinition moreFields
22. fieldDefinition ⇒TK_TYPE primitiveDatatype TK_COLON TK_FIELDID TK_SEM
23. moreFields ⇒ fieldDefinition moreFields
24. moreFields ⇒ eps
25. declarations ⇒ declaration declarations
26. declarations ⇒ eps

27. declaration ⇒TK_TYPE dataType TK_COLON TK_ID global_or_not TK_SEM
28. global_or_not ⇒ TK_COLON TK_GLOBAL
29. Global_or_not ⇒ eps
30. otherStmts ⇒ stmt otherStmts
31. otherStmts ⇒ eps
32. stmt ⇒ assignmentStmt
33. stmt ⇒ iterativeStmt
34. stmt ⇒conditionalStmt
35. stmt ⇒ ioStmt
36. stmt ⇒funCallStmt
37. assignmentStmt⇒singleOrRecId TK_ASSIGNOP arithmeticExpression TK_SEM
38. singleOrRecId ⇒ TK_ID new_24
39. New_24 ⇒ eps
40. new_24 ⇒ TK_DOT TK_FIELDID
41. funCallStmt ⇒outputParameters TK_CALL TK_FUNID TK_WITH TK_PARAMETERS inputParameters TK_SEM
42. outputParameters ⇒ TK_SQL idList TK_SQR TK_ASSIGNOP
43. outputParameters ⇒ eps
44. inputParameters ⇒ TK_SQL idList TK_SQR
45. iterativeStmt ⇒ TK_WHILE TK_OP booleanExpression TK_CL stmt otherStmts TK_ENDWHILE
46. conditionalStmt ⇒ TK_IF TK_OP booleanExpression TK_CL TK_THEN stmt otherStmts elsePart
47. elsePart ⇒ TK_ELSE stmt otherStmts TK_ENDIF
48. elsePart ⇒ TK_ENDIF
49. ioStmt ⇒ TK_READ TK_OP singleOrRecId TK_CL TK_SEM
50. ioStmt ⇒ TK_WRITE TK_OP allVar TK_CL TK_SEM
51. allVar ⇒ TK_ID temp
52. allVar ⇒ TK_NUM
53. allVar ⇒ TK_RNUM
54. arithmeticExpression ⇒ term expPrime
55. expPrime ⇒ lowPrecedenceOperators term expPrime
56. expPrime ⇒ eps
57. term ⇒ factor termPrime
58. termPrime ⇒ highPrecedenceOperator factor termPrime
59. termPrime ⇒ eps
60. factor ⇒ TK_OP arithmeticExpression TK_CL
61. factor ⇒ all
62. highPrecedenceOperator ⇒ TK_MUL
63. highPrecedenceOperator ⇒ TK_DIV
64. lowPrecedenceOperators ⇒ TK_PLUS
65. lowPrecedenceOperators ⇒ TK_MINUS
66. all ⇒ TK_NUM

67. all ⇒ TK_RNUM
68. all ⇒ TK_ID temp
69. temp ⇒ eps
70. temp ⇒ TK_DOT TK_FIELDID
71. booleanExpression ⇒ TK_OP booleanExpression TK_CL logicalOp TK_OP booleanExpression TK_CL
72. booleanExpression ⇒ var relationalOp var
73. booleanExpression ⇒ TK_NOT TK_OP booleanExpression TK_CL
74. var ⇒ TK_ID
75. var ⇒ TK_NUM
76. var ⇒ TK_RNUM
77. logicalOp ⇒ TK_AND
78. logicalOp ⇒ TK_OR
79. relationalOp ⇒ TK_LT
80. relationalOp ⇒ TK_LE
81. relationalOp ⇒ TK_EQ
82. relationalOp ⇒ TK_GT
83. relationalOp ⇒ TK_GE
84. relationalOp ⇒ TK_NE
85. returnStmt ⇒ TK_RETURN optionalReturn TK_SEM
86. optionalReturn ⇒ TK_SQL idList TK_SQR
87. optionalReturn ⇒ eps
88. idList ⇒ TK_ID more_ids
89. more_ids ⇒ TK_COMMA idList
90. more_ids ⇒ eps

Semantic Rules for AST

1. program.addr = newNode(P_Node, otherFunctions.addr, mainFunction.addr)
2. mainFunction.addr = stmts.addr,free(stmts)
3. otherFunctions.addr = newNode(function.addr,otherFunctions.addr)
4. otherFunctions.addr = NULL
5. function.addr = newNode(TK_FUNID.addr, input_par.addr, output_par.addr, stmts.addr)
6. input_par.addr = parameter_list.addr
7. output_par.addr = parameter_list.addr
8. output_par.addr = NULL
9. parameter_list.addr = newNode(dataType.addr, TK_ID.addr, remaining_list.addr)
10. dataType.addr = primitiveDatatype.addr,free(primitiveDatatype)
11. dataType.addr = constructedDatatype.addr,free(constructedDatatype)
12. primitiveDatatype.addr = newLeaf(TK_INT.addr)
13. primitiveDatatype.addr = newLeaf(TK_REAL.addr)
14. constructedDatatype.addr = newLeaf(TK_RECORDID.addr)
15. remaining_list.addr = parameter_list.addr
16. remaining_list.addr = NULL
17. stmts.addr = newNode(typeDefinitions.addr, declarations.addr, otherStmts.addr, returnStmt.addr)
18. typeDefinitions.addr = newNode(typeDefinition.addr, typeDefinitions.addr)
19. typeDefinitions.addr = NULL
20. typeDefinition.addr = newNode(TK_RECORDID.addr, fieldDefinitions.addr)
21. fieldDefinitions.addr = newNode(fieldDefinition.addr, fieldDefinition.addr, moreFields.addr)
22. fieldDefinition.addr = newNode(TK_TYPE.addr, primitiveDatatype.addr, TK_FIELDID.addr)
23. moreFields.addr = newNode(fieldDefinition.addr, moreFields.addr)
24. moreFields.addr = NULL
25. declarations.addr = newNode(declaration.addr, declarations.addr)
26. declarations.addr = NULL
27. declaration.addr = newNode(TK_TYPE.addr, dataType.addr, TK_ID.addr, global_or_not.addr)
28. global_or_not.addr = TK_GLOBAL.addr
29. global_or_not.addr = NULL
30. otherStmts.addr = stmt.addr otherStmts.addr
31. otherStmts.addr = NULL
32. stmt.addr = assignmentStmt.addr
33. stmt.addr = iterativeStmt.addr
34. stmt.addr = conditionalStmt.addr

35. stmt.addr = ioStmt.addr
36. stmt.addr = funCallStmt.addr
37. assignmentStmt.addr = newNode(singleOrRecId.addr, arithmeticExpression.addr)
38. singleOrRecId.addr = newNode(TK_ID.addr, new_24.addr)
39. new_24.addr = NULL
40. new_24.addr = newLeaf(TK_FIELDID.addr)
41. funCallStmt.addr = newNode(outputParameters.addr, TK_FUNID.addr, inputParameters.addr)
42. outputParameters.addr = idList.addr
43. outputParameters.addr = NULL
44. inputParameters.addr = idList.addr, free(idList.addr)
45. iterativeStmt.addr = newNode(booleanExpression.addr, stmt.addr , otherStmts.addr)
46. conditionalStmt.addr = newNode(booleanExpression.addr, stmt.addr, otherStmts.addr, elsePart.addr)
47. elsePart.addr = newNode(stmt.addr, otherStmts.addr)
48. elsePart.addr = NULL
49. ioStmt = singleOrRecId.addr
50. ioStmt = allVar.addr
51. allVar = newNode(TK_ID.addr, temp.addr)
52. allVar = newLeaf(TK_NUM.addr)
53. allVar = newLeaf(TK_RNUM.addr)
54. arithmeticExpression.addr = newNode(term.addr,expPrime.addr)
55. expPrime.addr = newNode(lowPrecedenceOperators.addr, term.addr, expPrime.addr)
56. expPrime.addr = NULL
57. term.addr = newNode(factor.addr, termPrime.addr)
58. termPrime.addr = newNode(highPrecedenceOperator.addr, factor.addr, termPrime.addr)
59. termPrime.addr = NULL
60. Factor.addr = arithmeticExpression.addr
61. Factor.addr = all.addr
62. highPrecedenceOperator.addr = TK_MUL.addr
63. highPrecedenceOperator.addr = TK_DIV.addr
64. lowPrecedenceOperators.addr = TK_PLUS.addr
65. lowPrecedenceOperators.addr = TK_MINUS.addr
66. all.addr = newLeaf(TK_NUM.addr)
67. all.addr = newLeaf(TK_RNUM.addr)
68. all.addr = newNode(TK_ID.addr, temp.addr)
69. temp = NULL
70. temp = TK_FIELDID.addr
71. booleanExpression.addr = newNode(booleanExpression.addr, logicalOp.addr, booleanExpression.addr)
72. booleanExpression.addr = newNode(var.addr, relationalOp.addr, var.addr)
73. booleanExpression.addr = booleanExpression.addr

74. var.addr = TK_ID.addr
75. var.addr = newLeaf(TK_NUM.addr)
76. var.addr = newLeaf(TK_RNUM.addr)
77. logicalOp.addr = TK_AND.addr
78. logicalOp.addr = TK_OR.addr
79. relationalOp.addr = TK_LT.addr
80. relationalOp.addr = TK_LE.addr
81. relationalOp.addr = TK_EQ.addr
82. relationalOp.addr = TK_GT.addr
83. relationalOp.addr = TK_GE.addr
84. relationalOp.addr = TK_NE.addr
85. returnStmt.addr = optionalReturn.addr
86. optionalReturn.addr = idList.addr
87. optionalReturn.addr = NULL
88. idList.addr = newNode(TK_ID.addr, more_ids.addr)
89. more_ids.addr = idList.addr
90. More_ids.addr = NULL