ABSTRACT SYNTAX TREE CREATION RULES

Nithin Benny Myppan
2016A7PS0014P

Swarup Natraj
2016A7PS0080P

GROUP 26

Adhitya Mamallan
2016A7PS0028P

Naveen Unnikrishnan
2016A7PS0111P

Grammar

1. <program> ===> <otherFunctions> <mainFunction>
2. <mainFunction> ===> TK_MAIN <stmts> TK_END
3. <otherFunctions> ===> <function><otherFunctions>
4. <otherFunctions> ===> e
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> ===> e
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> ===> e
17. <stmts> ===> <typeDefinitions> <declarations> <otherStmts><returnStmt>
18. <typeDefinitions> ===> <typeDefinition><typeDefinitions>
19. <typeDefinitions> ===> e
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> ===> e
25. <declarations> ===> <declaration><declarations>
26. <declarations> ===> e
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> ===> e
30. <otherStmts> ===> <stmt><otherStmts>
31. <otherStmts> ===> e
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> ===> e
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> ===> e
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 <allVar_1>
52. <allVar> ===> TK_NUM
53. <allVar> ===> TK_RNUM
54. <allVar_1> ===> TK_DOT TK_FIELDID
55. <allVar_1> ===> e
56. <arithmeticExpression> ===> <term> <expPrime>
57. <expPrime> ===> <lowPrecedenceOperators> <term> <expPrime>
58. <expPrime> ===> e
59. <term> ===> <factor> <termPrime>
60. <termPrime> ===> <highPrecedenceOperators><factor> <termPrime>
61. <termPrime> ===> e
62. <factor> ===> TK_OP <arithmeticExpression> TK_CL
63. <factor> ===> <all>
64. <highPrecedenceOperators> ===> TK_MUL
65. <highPrecedenceOperators> ===> TK_DIV
66. <lowPrecedenceOperators> ===> TK_PLUS
67. <lowPrecedenceOperators> ===> TK_MINUS
68. <all> ===> TK_NUM
69. <all> ===> TK_RNUM
70. <all> ===> TK_ID <temp>
71. <temp> ===> e
72. <temp> ===> TK_DOT TK_FIELDID

73. <booleanExpression> ===> TK_OP <booleanExpression> TK_CL <logicalOp> TK_OP <booleanExpression> TK_CL
74. <booleanExpression> ===> <var> <relationalOp> <var>
75. <booleanExpression> ===> TK_NOT TK_OP <booleanExpression>TK_CL
76. <var> ===> TK_ID
77. <var> ===> TK_NUM
78. <var> ===> TK_RNUM
79. <logicalOp> ===> TK_AND
80. <logicalOp> ===> TK_OR
81. <relationalOp> ===> TK_LT
82. <relationalOp> ===> TK_LE
83. <relationalOp> ===> TK_EQ
84. <relationalOp> ===> TK_GT
85. <relationalOp> ===> TK_GE
86. <relationalOp> ===> TK_NE
87. <returnStmt> ===> TK_RETURN <optionalReturn> TK_SEM
88. <optionalReturn> ===> TK_SQL <idList> TK_SQR
89. <optionalReturn> ===> e
90. <idList> ===> TK_ID <more_ids>
91. <more_ids> ===> TK_COMMA <idList>
92. <more_ids> ===> e

Semantic Rules

1. program.addr = makeNode(PROGRAM, otherFunctions.addr, mainFunction.addr)
2. mainFunction.addr = makeNode(MAIN, stmts.addr)
3. otherFunctions.addr = makeNode(OTHER_FN, function.addr, otherFunctions.addr)
4. otherFunctions.addr = NULL
5. function.addr = makeNode(FN, TK_FUNID.addr, input_par.addr, output_par.addr, stmts.addr)
6. input_par.addr = parameter_list.addr, free(parameter_list)
7. output_par.addr = parameter_list.addr, free(parameter_list)
8. output_par.addr = NULL
9. parameter_list.addr = makeNode(PAR_LIST, 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 = makeLeaf(TK_INT.addr)
13. primitiveDatatype.addr = makeLeaf(TK_REAL.addr)
14. constructedDatatype.addr = makeLeaf(TK_RECORDID.addr)
15. remaining_list.addr = parameter_list.addr, free(parameter_list)
16. remaining_list.addr = NULL
17. stmts.addr = makeNode(STMTS, typeDefinitions.addr, declarations.addr, otherStmts.addr, returnStmt.addr)
18. typeDefinitions.addr = makeNode(TYPE_DEFS, typeDefinition.addr, typeDefinitions.addr)

19. typeDefinitions.addr = NULL
20. typeDefinition.addr = makeNode(TYPE_DEF, TK_RECORDID.addr, fieldDefinitions.addr)
21. fieldDefinitions.addr = makeNode(FIELD_DEFS, fieldDefinition1.addr, fieldDefinition2.addr, moreFields.addr)
22. fieldDefinition.addr = makeNode(FIELD_DEF, primitiveDatatype.addr, TK_FIELDID.addr)
23. moreFields.addr = makeNode(MORE_FIELDS, fieldDefinition.addr, moreFields.addr)
24. moreFields.addr = NULL
25. Declarations.addr = makeNode(DECLARATIONS, declaration.addr, declarations.addr)
26. declarations.addr = NULL
27. declaration.addr = makeNode(DECLARATION, dataType.addr, TK_ID.addr, global_or_not.addr)
28. global_or_not.addr = makeLeaf(TK_GLOBAL.addr)
29. global_or_not.addr = NULL
30. otherStmts.addr = makeNode(OTHER_STMTS, stmt.addr, otherStmts.addr)
31. otherStmts.addr = NULL
32. stmt.addr = assignmentStmt.addr, free(assignmentStmt)
33. stmt.addr = iterativeStmt.addr, free(iterativeStmt)
34. stmt.addr = conditionalStmt.addr, free(conditionalStmt)
35. stmt.addr = ioStmt.addr, free(ioStmt)
36. stmt.addr = funCallStmt.addr, free(funCallStmt)
37. assignmentStmt.addr = makeNode(ASSIGNMENT_STMT, singleOrRecId.addr, arithmeticExpression.addr)
38. singleOrRecId.addr = makeNode(SINGLE_OR_REC_ID, TK_ID.addr, new_24.addr)
39. new_24.addr = NULL
40. new_24.addr = makeLeaf(TK_FIELDID.addr)
41. funCallStmt.addr = makeNode(FUN_CALL_STMT, outputParameters.addr, TK_FUNID.addr, inputParameters.addr)
42. outputParameters.addr = idList.addr, free(idList)
43. outputParameters.addr = NULL
44. inputParameters.addr = idList.addr, free(idList)
45. iterativeStmt.addr = makeNode(ITERATIVE_STMT, booleanExpression.addr, stmt.addr, otherStmts.addr)
46. conditionalStmt.addr = makeNode(CONDITIONAL_STMT, booleanExpression.addr, stmt.addr, otherStmts.addr, elsePart.addr)
47. elsePart.addr = makeNode(ELSE, stmt.addr, otherStmts.addr)
48. elsePart.addr = NULL
49. ioStmt.addr = makeNode(IO_STMT_READ, singleOrRecId.addr)
50. ioStmt.addr = makeNode(IO_STMT_WRITE, allVar.addr)
51. allVar.addr = makeNode(ALL_VAR, TK_ID.addr, allVar_1.addr)
52. allVar.addr = makeLeaf(TK_NUM.addr)
53. allVar.addr = makeLeaf(TK_RNUM.addr)
54. allVar_1.addr = makeLeaf(TK_FIELDID.addr)
55. allVar_1.addr = NULL

56. arithmeticExpression.addr = makeNode(ARITHMETIC, term.addr, expPrime.addr)
57. expPrime.addr = makeNode(EXP_PRIME, lowPrecedenceOperators.addr, term.addr, expPrime.addr)
58. expPrime.addr = expPrime.inh_addr
59. term.addr = makeNode(TERM, factor.addr, termPrime.addr)
60. termPrime.addr = makeNode(TERM_PRIME, highPrecedenceOperators.addr, factor.addr, termPrime.addr)
61. termPrime.addr = termPrime.inh_addr
62. Factor.addr = arithmeticExpression.addr
63. factor.addr = all.addr, free(all)
64. highPrecedenceOperators.addr = makeLeaf(TK_MUL.addr)
65. highPrecedenceOperators.addr = makeLeaf(TK_DIV.addr)
66. lowPrecedenceOperators.addr = makeLeaf(TK_PLUS.addr)
67. lowPrecedenceOperators.addr = makeLeaf(TK_MINUS.addr)
68. all.addr = makeLeaf(TK_NUM.addr)
69. all.addr = makeLeaf(TK_RNUM.addr)
70. all.addr = makeNode(ALL, TK_ID.addr, temp.addr)
71. temp.addr = NULL
72. temp.addr = makeLeaf(TK_FIELDID.addr)
73. booleanExpression.addr = makeNode(BOOLEAN, booleanExpression1.addr, logicalOp.addr, booleanExpression2.addr)
74. booleanExpression.addr = makeNode(BOOLEAN, var1.addr, relationalOp.addr, var2.addr)
75. booleanExpression.addr = makeNode(BOOLEAN, TK_NOT.addr, booleanExpression.addr)
76. var.addr = makeLeaf(TK_ID.addr)
77. var.addr = makeLeaf(TK_NUM.addr)
78. var.addr = makeLeaf(TK_RNUM.addr)
79. logicalOp.addr = makeLeaf(TK_AND.addr)
80. logicalOp.addr = makeLeaf(TK_OR.addr)
81. relationalOp.addr = makeLeaf(TK_LT.addr)
82. relationalOp.addr = makeLeaf(TK_LE.addr)
83. relationalOp.addr = makeLeaf(TK_EQ.addr)
84. relationalOp.addr = makeLeaf(TK_GT.addr)
85. relationalOp.addr = makeLeaf(TK_GE.addr)
86. relationalOp.addr = makeLeaf(TK_NE.addr)
87. returnStmt.addr = makeNode(RETURN, TK_RETURN.addr, optionalReturn.addr)
88. optionalReturn.addr = idList.addr, free(idList)
89. optionalReturn.addr = NULL
90. idList.addr = makeNode(RETURN, TK_ID.addr, more_ids.addr)
91. more_ids.addr = idList.addr, free(idList)
92. more_ids.addr = NULL