GROUP-20
DATE - 19/2/23

MEMBERS

RAJAN - 2019B4A70572P
YASH - 2019B4A70638P
AYUSH - 2019B4A70652P
VASU - 2019B4A70656P
SUDARSHAN - 2019B4A70744P

GRAMMAR

<start> → <program> $

1. <program> → <moduleDeclarations> <otherModules> <driverModule> <otherModules>
2. <moduleDeclarations> → <moduleDeclaration> <moduleDeclarations> | ∈
3. <moduleDeclaration> → DECLARE MODULE ID SEMICOL
4. <otherModules> → <module> <otherModules> | ∈
5. <driverModule> → DRIVERDEF DRIVER PROGRAM DRIVERENDDEF <moduleDef>
6. <module> → DEF MODULE ID ENDDEF TAKES INPUT SQBO <input_plist> SQBC SEMICOL <ret> <moduleDef>
7. <ret> → RETURNS SQBO <output_plist> SQBC SEMICOL | ∈
8. <input_plist> → ID COLON <dataType> <ip>
9. <ip> → COMMA ID COLON <dataType> <ip> | ∈
10. <output_plist> → ID COLON <type> <op>
11. <op> → COMMA ID COLON <type> <op> | ∈
12. <dataType> → INTEGER
13. <dataType> → REAL
14. <dataType> → BOOLEAN
15. <dataType> → ARRAY SQBO <range> SQBC OF <type>
16. <type> → INTEGER
17. <type> → REAL
18. <type> → BOOLEAN
19. <moduleDef> → START <statements> END
20. <statements> → <statement> <statements> | ∈
21. <statement> → <ioStmt>
22. <statement> → <simpleStmt>
23. <statement> → <declareStmt>
24. <statement> → <conditionalStmt>
25. <statement> → <iterativeStmt>
26. <ioStmt> → GET_VALUE BO ID BC SEMICOL
27. <ioStmt> → PRINT BO <var"> BC SEMICOL
28. <var"> → ID <whichID>
29. <var"> → NUM
30. <var"> → RNUM
31. <var"> → TRUE
32. <var"> → FALSE

33. <whichID> → SQBO <which'> SQBC | ε
34. <which'> → <type'>
35. <which'> → <arithmetic Expression'>
36. <type'> → NUM
37. <type'> → ID
38. <simpleStmt> → <assignmentStmt>
39. <simpleStmt> → <moduleReuseStmt>
40. <assignmentStmt> → ID <whichStmt>
41. <whichStmt> → <lvalueIDStmt>
42. <whichStmt> → <lvalueARRStmt>
43. <LValueIDStmt> → ASSIGNOP <expression> SEMICOL
44. <lvalueARRStmt> → SQBO <index> SQBC ASSIGNOP <expression> SEMICOL
45. <index> → <arithmetic Expression'>
46. <moduleReuseStmt> → <optional> USE MODULE ID WITH PARAMETERS <idList> SEMICOL
47. <optional> → SQBO <idList> SQBC ASSIGNOP | ε
48. <idList> → ID <idList'>
49. <idList'> → COMMA ID <idList'> | ε
50. <expression> → <arithmetic Expression>
51. <expression> → <boolean Expression>
52. <arithmeticExpression> → <term> <ae>
53. <arithmeticExpression> → PLUS <newterm>
54. <arithmeticExpression> → MINUS <newterm>
55. <ae> → <op'> <term> <ae> | ε
56. <term> → <factor> <te>
57. <te> → <op''> <factor> <te> | ε
58. <factor> → BO <arithmetic Expression> BC
59. <factor> → <var>
60. <var> → ID <whichID>
61. <var> → NUM
62. <var> → RNUM
63. <op'> → PLUS
64. <op'> → MINUS
65. <op''> → MUL
66. <op''> → DIV
67. <arithmeticExpression'> → <term'> <ae'>
68. <arithmeticExpression'> → PLUS <newterm'>
69. <arithmeticExpression'> → MINUS <newterm'>

<ae'> → <op'> <term'> <ae'> | ε

<term'> → <factor'> <te'>

<te'> → <op''> <factor'> <te'> | ε

<factor'> → BO <arithmetic expression'> BC

<factor'> → <var''>

<var'''> → ID

<var'''> → NUM

<var'''> → RNUM

<newterm> → ID

<newterm> → NUM

<newterm> → RNUM

<newterm> → BO <arithmeticExpression> BC

<newterm'> → ID

<newterm'> → NUM

<newterm'> → RNUM

<newterm'> → BO <arithmetic Expression'> BC

<booleanExpr> → <arithmeticExpression> <booleanExpr'>

<booleanExpr'> → <relationalOp> <arithmeticExpression> <bE> | ε

<bE> → <logical Op> <arithmetic Expression> <relational Op> <arithmeticExpression> <bE> | ε

<booleanExpr> → BO <boolean Expr.> BC <be'>

<be'> → <logicalOp> BO <booleanExpr> BC <be'> | ε

<be'> → TRUE

<be'> → FALSE

<be'> → ID

<logicalOp> → AND | OR

<relational Op> → LT

<relational Op> → LE

<relational Op> → GT

<relational Op> → GE

<relational op> → EQ

<relational op> → NE

<declareStmt> → DECLARE <idList> COLON <dataType> SEMICOL

<conditional Stmt> → SWITCH BO ID BC START <caseStmt> <default> END

<caseStmt> → CASE <int.bool>

<int_bool> → <int>

<int_bool> → <bool>

<int> → NUM COLON <statements> BREAK SEMICOL <r_int>

<r_int> → CASE <int>

<r_int> → DEFAULT

<bool> → <value> COLON <statements> BREAK SEMICOL <r_bool>

<r_bool> → CASE <bool> | ε

<value> → TRUE

<value> → FALSE

<default> → DEFAULT COLON <statements> BREAK SEMICOL | ε

<iterativeStmt> → FOR BO ID IN <range> BC START <statements> END.

<iterativeStmt> → WHILE BO <booleanExpr> BC START <statements> END.

<range> → NUM RANGEOP NUM

<range'> → <typ'> RANGEOP <typ'>

Assumptions:

1. a := a + 2 + 2 and similar statements are not allowed.
2. a++a , --a are not allowed.
3. -(-(a+b)) and similar statements are allowed.
4. A[a+(2+2)] indexing is allowed

FIRST SETS

FIRST (<stmt>) = {DECLARE, DEF, DRIVERDEF}

FIRST (<program>) = {DECLARE, DEF, DRIVERDEF}

FIRST (<moduleDeclerationS>) = {DECLARE, ϵ}

FIRST (<moduleDeclaration>) = {DECLARE}

FIRST (<otherModuel>) = {DEF, ϵ}

FIRST (<driverModule>) = {DRIVERDEF}

FIRST (<module>) = {DEF}

FIRST (<ret>) = {RETURNS, ϵ}

FIRST (<input_plist>) = {ID}

FIRST (<ip>) = {COMMA, ϵ}

FIRST (<output_plist>) = {ID}

FIRST (<op>) = {COMMA, ϵ}

FIRST (<dataType>) = {INTEGER, REAL, BOOLEAN, ARRAY}

FIRST (<type>) = {INTEGER, REAL, BOOLEAN}

FIRST (<moduleDef>) = {START}

FIRST (<statements>) = {GET_VALUE, PRINT, ID, SQBO, USE, DECLARE, SWITCH, FOR, WHILE, ϵ}

FIRST (<statement>) = {GET_VALUE, PRINT, ID, SQBO, USE, DECLARE, SWITCH, FOR, WHILE}

FIRST (<ioStmt>) = {GET_VALUE, PRINT}

FIRST (<var>) = {ID, NUM, RNUM, TRUE, FALSE}

FIRST (<whichID>) = {SQBO, ϵ}

FIRST (<whichStmt>) = {NUM, ID, BO, RNUM, PLUS, MINUS}

FIRST (<lvalue>) = {NUM, ID}

FIRST (<simpleStmt>) = {ID, SQBO, USE}

FIRST (<assignmentStmt>) = {ID}

FIRST (<whichStmt>) = {ASSIGNOP, SQBO}

FIRST (<lValueIdStmt>) = {ASSIGNOP}

FIRST (<lValueArrStmt>) = {SQBO}

FIRST (<index>) = {NUM, ID, BO, RNUM, PLUS, MINUS}

FIRST (<moduleReuseStmt>) = {SQBO, USE}

FIRST(<optional>) = {SQBO, ε}

FIRST(<idList>) = {ID}

FIRST(<idList'>) = {COMMA, ε}

FIRST(<expression>) = {BO, ID, NUM, RNUM, PLUS, MINUS, LT, LE, GT, GE, EQ, NE, ε, ~}

FIRST(<arithmeticExpression>) = {BO, ID, NUM, RNUM, PLUS, MINUS}

FIRST(<ae>) = {PLUS, MINUS, ε}

FIRST(<term>) = {BO, ID, NUM, RNUM}

FIRST(<te>) = {MUL, DIV, ε}

FIRST(<factor>) = {BO, ID, NUM, RNUM}

FIRST(<var>) = {ID, NUM, RNUM}

FIRST(<op'>) = {PLUS, MINUS}

FIRST(<op''>) = {MUL, DIV}

FIRST(<arithmeticExpression'>) = {BO, ID, NUM, RNUM, PLUS, MINUS}

FIRST(<ae'>) = {PLUS, MINUS, ε}

FIRST(<term'>) = {BO, ID, NUM, RNUM}

FIRST(<te'>) = {MUL, DIV, ε}

FIRST(<factor'>) = {ID, NUM, RNUM, BO}

FIRST(<var'''>) = {ID, NUM, RNUM}

FIRST(<newTerm>) = {ID, NUM, RNUM, BO}

FIRST(<newTerm'>) = {ID, NUM, RNUM, BO}

FIRST(<booleanExpr>) = {LT, LE, GT, GE, EQ, NE, ε}

FIRST(<booleanExpr'>) = {LT, LE, GT, GE, EQ, NE, ε}

FIRST(<bE>) = {AND, OR, ε}

FIRST(<bE'>) = {AND, OR, ε, TRUE, FALSE, ID}

FIRST(<logicalOp>) = {AND, OR}

FIRST(<relationalOp>) = {LT, LE, GT, GE, EQ, NE}

FIRST(<declareStmt>) = {DECLARE}

FIRST(<conditionalStmt>) = {SWITCH}

FIRST(<caseStmt>) = {CASE}

FIRST(<int_bool>) = {NUM, TRUE, FALSE}

FIRST(<int>) = {NUM}

FIRST(<rint>) = {CASE, DEFAULT}

FIRST (<bool>) = { TRUE, FALSE }

FIRST (<rbool>) = { CASE, ε }

FIRST (<value>) = { TRUE, FALSE }

FIRST (<default>) = { DEFAULT, ε }

FIRST (<iterativeStmt>) = { FOR, WHILE }

FIRST (<range>) = { NUM }

FIRST (<range'>) = { NUM, ID }

FOLLOW SETS

1. FOLLOW (<moduleDeclarations>) = {DEF, DRIVERDEF}
2. FOLLOW (<otherModules>) = {DRIVERDEF, $}
3. FOLLOW (<ret>) = {START}
4. FOLLOW (<ip>) = {SQBC}
5. FOLLOW (<program>) = {$}
6. FOLLOW (<input_plist>) = {SQBC}
7. FOLLOW (<op>) = {SQBC}
8. FOLLOW (<output_plist>) = {SQBC}
9. FOLLOW (<statements>) = {END, BREAK}
10. FOLLOW (<whichId>) = {MUL, DIV, PLUS, MINUS, SEMICOL, BC, LT, LE, GT, GE, EQ, NE, AND, OR}
11. FOLLOW (<var'>) = {BC}
12. FOLLOW (<optional>) = {USE}
13. FOLLOW (<idList'>) = {COLON, SEMICOL, SQBC, COMMA}
14. FOLLOW (<idList>) = {COLON, SEMICOL, SQBC, COMMA}.
15. FOLLOW (<arithmeticExpression>) = {SEMICOL, BC, LT, LE, GT, GE, EQ, NE, AND, OR}
16. FOLLOW (<ae>) = {SEMICOL, BC, LT, LE, GT, GE, EQ, NE, AND, OR}
17. FOLLOW (<be'>) = {BC, SEMICOL}
18. FOLLOW (<booleanExpr>) = {BC, SEMICOL}
19. FOLLOW (<t'>) = {PLUS, MINUS, SEMICOL, BC, LT, LE, GT, GE, EQ, NE, AND, OR}
20. FOLLOW (<term>) = {PLUS, MINUS, SEMICOL, BC, LT, LE, GT, GE, EQ, NE, AND, OR}
21. FOLLOW (<var>) = {MUL, DIV, PLUS, MINUS, SEMICOL, BC, LT, LE, GT, GE, EQ, NE, AND, OR}
22. FOLLOW (<factor>) = {MUL, DIV, PLUS, MINUS, SEMICOL, BC, LT, LE, GT, GE, EQ, NE, AND, OR}

23. FOLLOW(<ae'>) = {BC, $ BC}
24. FOLLOW(<arithmeticExpression'>) = {BC, $ BC}
25. FOLLOW(<which'>) = {$ BC}
26. FOLLOW(<index>) = {$ BC}
27. FOLLOW(<te'>) = {PLUS, MINUS, $ BC, BC}
28. FOLLOW(<term'>) = {PLUS, MINUS, $ BC, BC}
29. FOLLOW(<booleanExpr'>) = {BC, SEMICOL}
30. FOLLOW(<be>) = {BC, SEMICOL}
31. FOLLOW(<r_bool>) = {DEFAULT, END}
32. FOLLOW(<ind_bool>) = {DEFAULT, END}
33. FOLLOW(<caseStmt>) = {DEFAULT, END}
34. FOLLOW(<bool>) = {DEFAULT, END}
35. FOLLOW(<default>) = {COLON}