2020A7PS0072 Archal Join Siddhauth Khandelneal 2020A7PS 0098 2020A7 PS0092 Sarthak Shah Richi Rakuh ShrivastuazozoA7PSQ108 Brown pratap Rathore 2020ATPS 1675

Modified Gramman with first & Pollow

< module Reclarations > < other Modules > < driver Module > < other Modules > Start Symbol (FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)

FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)

FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)

FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)

FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)

FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)

FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)

FIRST (module De clarations) U (FIRST (athurmodules) - Y FIRST (driver Module)) progroum> -FIRST (program) =

FOLLOW (program) = 545

- <module Declaration> < module Declarations > / È < mo dule Declarations> FIRST (moduleDeclarations) = { DECLARE), E3 (2) FOLLOW (module Declarations) = {DEF, f, DRIVER DEF}
- DECLARE MODULE ID SEMICOL < module Declaration) ____ FIRST (module Deducation) = \ \ DECLAREZ 3 FOLLOW (module Declaration) = SDECLARE, DEF, DRIVER DEF, \$3
- cother Modules > ____ cmodule > < other Modules > | G FIRST (office Modules) = { DEF }, E} Follow (alher Modules) = { DRIVER DEF, \$}
 - DRIVERENDEF (Module Def) PRO GILAM <driver Module) - DRIVER DEF DRIVER (3) FIRST (driver, Madule) = { DRIVERDEFY FOLLOW (driver Module) = & DEF, \$3
- 6 < module) -> DEF MODULE ID ENDDEF TAKES INPUT SORD < input_plist> SORC SEMICOL < ret> cmodule FIRST (module) = {DEF} FOLLOW (module) = { DEF, DRIVER DEF, \$}
- SQBO Content-list > SQBC SEMICOL C RETURNS < ret> -FIRST (ret) = { RETURNS, E3 Facour(red) = { START3
- (8) cinput-plist> -> ID colon < dataType> <NI> FIRST (input-plist) = { ID3

FOLLOW (input-plist) = { 59BC}

COMMA ID COLON < data Type> <NI> | E (9) <NI> FIRST - { COMMA, E3

FOLLOWY 259BC3 <type> <N2> COLON <output_plist> FIRST + \$ IQ3

FOLLOW + & SPBC3

```
< type> <N2> / E
                                             COLON
      FIRST - ¿ COMMA, ES
                      FOLLOWIZ SPBC3
    (2) < data Type>
                            NTEGER REAL BOOLEAN ARRAY SOBO < VAY-runge SOBC OF
                      FIRST+ {INTEGER, REAL, BOOLEAN, ARRAY}
                       FOLLOW + & COMMA, S&BC& , SENECOLS
                              <index > RANGE- OP < index>
       < var-range >
                        FIRST 1 & ID, NUM }
                         FOLLOW + { SABC3
                            INTEGER | REAL BOOLEAN
  (14)
        <type)
                       First > 2 INTEGER, REAL, BOOLEANG
                       FOllow -> 2 COMMA, SOB & , SEMELOLG
                           START (statements) END
       < module Def) ->
 (15)
                 FINE - 1 STARTS
                  Follow -> 20EF, ORIVERDEF, $5
                             <statement> <statemente> / E
(16)
      < statements) ->
                 FINE -> {GET-VALUE, PRINT, IO, SQOO, USE, DECKARE, SWITCH, FOR, WHILE 1, ES
                  FOLL OW 4 & ENDS
                          <10 Stat> | < simple Stat> | < declarestant> | < conditional Start> | <itexative Start>
(17)
                 FINE -> CAET-VALUE, PRINT, ID, SOBO, USE, DECLARE, SWITCH, FOR, WHILES
                 FOLLOWS ENOU {FIRST (statements) - E}
                                                   BC SEMICOL | PRINT BO EVANCES BC SEMICOL
       <iosfmt>-
 (18)
                       GET_VALUE
                                        BO
                                              ID
                     FINE -> 2GET_VALUE, PRINT S
                     FOLLOW (Statement) 
<arithematic-Vax / (bool - Vare)
(13)
                    FINE > ? ID, NUM, RNUM, TRUE, FALSES
                     FOLLOW + 2BC
                                    FALSE
                          TRUE
     < bool - var)
                    FINE ZTYUE, FALSES
                   FOLLOW + 2BC, AND, OR, & ENELOL &
                                     < which ID> | NUM | RNUM
    <arithematic _var>
                               ID
                   FIRE - LID, NUM, ENUMS
                   60110W-> 2 MUL, DIV, PLUS, NUNUS, LT, G,T, LE, GEI ER, NE, FAND, OR, BC, SENZON
                           SOBO <index> SOBC/C
    <which ID>
                  FINE > 2 SOBO, ES
                   FOLLOW > FOllow (anithmetic_Val)
                               <assignment Stret > / < module Reuse Stret >
    < simple stat>
                 FINE > ZID, SOBO, USE 3
                 FOLLOW + FOLLOW (Statement)
   < assignment Start>
                              10 < which start>
                 FINEY LIDG
                 FOLLOW (statement)
                                             / ( Value Aur Stut >
                            < 1 Value 1D Strut >
    < rehich Stmt>
                  FINES ZASSEGNOP, SOBOS
                  FOLLOW (assignment start)
```

2

```
FINE > ZASSZUNOPS
                     howard FOLLOW (which struct)
        < 1 Value Acus Strut > -->
                              SOBO <index > SOBC
                                                      ASSIGNOP
                                                                 CEXPULSSION SEMICOL
                     FINE + 250 BOS
                     FOLLOW ( which stut)
                  - NOW ID
(28)
        <index>
                 FINE + ZNUM, 205
                  FOLLOW > ERANGEOF, SOBC 5
       Emoduly Ruse Start > -> <optional> USE MODULE ID WITH PARAMETERS Changilist> SEMICOL
(29)
                   Fine of Esobor 5 USES
                                                         - assuming
                                                            ACKI can't be passed
                    FOLLOW (Simple Struct)
                                                                        as well as taken
                                                                                 as function
                                                                                     output)
                            < modt4ist> <M3>
      < param list>
                    \longrightarrow
(39)
                    FINEY ENUN, ID, RNUN, TRUE, FALSES
                     FOLLOW + LSEMIEUL 9
                            <mathst> <N3> (C
                    COMMA
(31)
                   FINE + ZUNNA, ES
                  FULLWWY & SENZULS
                 NUM RNUM | 10 TRUE | FALSE
     < multiset >
                FINE - ZNUMIRNUM , ZOITRUE, FALSE S
                Follow + 2 COMMA SENECULS
                     SOBO cidlist> SOBC ASCION OP/E
   coptional)
                  FIXE ZSOBO, ES
                  FOLLOWY ZUSES
        ∠idlist> → 10 < N4>
  (39)
             FINES ZIDS
              willow + ¿ sarc g
                 - COMMA 10 <N4> 16
 (33)
             FINE -> ZEMMA, ES
       <expression) -> <arithmeatic or boolex hu> | < unauy>
              RULIOW -> 250BCS
 (36)
              FIXLD & TIFIBOLIO, NUM, ROUM, PLUS, NEWUS S
              FOLLOW > 25 ENZULS
       < unauy - op > <N5>
(37)
              Fixen & Plusa, MINUSS
               WILLOW + ZSEMZWIS
                          BO carithemetic Expu) BC | carithemetic -var)
       < N5>
(38)
                 CINE , EBOINUMINIAMUM S
                 SCHOOLS SENTINGS WINDS
      < vnaly-op>
(39)
                FINES ZPLUS, NINUSS
                FULLOW & ZBO, NUN, ID, KNUNG
      < arithmetic or bealexpr> -> < boolexpr> (NG) < boolexpr> <NG>
(40)
               FIRST STRUE, FALSE, BO, ID, NUM, KNUMS
                FOLLOW -> ESENZION, BC &
```

(4) (NE) _ Clogical op> < boolexpr> <NE> (E)

Fix b > 2 ANO, OR, E S

FOLLO W > 2 SENECOL, BCS

(92) < boolexpr> -> < bool-var) / authematic Expr> < N7>
First > { TRUE, FASSE, BO, ID, NUM, RNUM S
FOLLOWS ZANDIOR, SENECOL, BCS

(93) <N7) -> < rel-of> < arithmatic - expr) | e

FIRST + 2 LT, LE, GT, GE, ED, NE, E3

FOLLOW + 2 AND, OR, BC, SEMICOL 3

(49) caritumetic _expr> --> <term> <N8>
FIRST - & BO, ID, NUM, RNUM?

FOLLOW - & LT, LE, GT, GE, EQ, NE, AND, OR, BC, SEMICOL 3

(45) <N8) -> <OPD < HUM > <N8 > | E FIRST 1 { PLUS, MINUS, E3 FOLLOW + } LT, LE, GT, GE, EQ, NE, AND, OR, BC, SEMI COI]

(G) < term) -> < factor>< N9)

FIRST - 2 BO, ID, NUM, RNUM3

FOLLOW + 3 PLUS, MINUS, LT, LE, GT, GE, EB, NE, AND, OR, BC, SEMICOL3

(47) < N9> -> < Ob2> < factor> < N9> | E FIRST + { MUL, DIV, E} FOLLOW + (FOLLOW (term)

(98) < factor > ___ Bo(bal Expr) Bc / < auithematic _ vari)

FIRST + ¿ Bo, IO, NUM, RNUM3

FOLLOW + ¿ MUL, DZV, PLUS, MINUS, LT, LE, GT, GE, Eg, NE, AND, OK, BC, SEMICOL3

(49) <OPI> - PLUS | MINUS
FIRST 1 & PLUS, MINUS }
FOLLOW 1 & BO, ZO, NUM, RNUM 3

(50) <0P2> -> MUL/DIV FIRST + { MUL, DIVS FOLLOW + { BO, ID, NUM, RNUM}

(51) < logical Op> -- ANDLOR FIRST- & ANDLOR

FOLLOW 1 { TRUE, FALSE, BO, IO, NUM, RNUM3

FIRST + {LT, LE, GT, GE, ED, NE3
FOLLOWT { BO, ID, NUM, RNUM?

Assuming

Q + (b < c)

Q * (b < = c)

is syntachically

correct

& error

will

re shown

during

symanhic

aralysi's

dichere Stat> DECLARE <id4+> COLON <dota Type > SEMICOL FIRST + EDECLARES FOLLOW + FOLLOW (Statement) 2 conditional Start> START < cese stants > < default > END -> SWITCH BO ID BC FIRST+ ESWITCHS FOLLOW + FOLLOW (Stolement) < N/0> SEMICOL CASE <value> COLON <sbut> BREAK < ase Stants > FIRST - ¿ CASES FOLLOW & DEFAULT, ENDS SEMICOL (NIO) / <stnt> BREAK CASE <value> COLON FIRST + {CASE, E} FOLLOWY & DEFAULT, ENOS NUM TRUE FALSE (value) FIRST + & NUM, TRUE, FALSE 3 FOLLOWS & OFFAUT, EN COLON } DEFAULT COLON & STATEMENTS > BREAK SEMICOL E FIRST + ¿ DEFAULT, ES FOLLOW + & END3 -> FOR BO ID IN CrayES BC START estatements > END <i turative Strut> WHILE BO Carithenetic or boolean EXPUS BC START < Statements > END FIRST + { FOR, WHILE } FOLLOW (Statement) RANGEOP NUM (60) FIRST + {NUMS FOLLOW 1 & BC} 1

1

#

10

: