```
[1]
     <PROGRAM>
                             \rightarrow <IDENT><ENV><DATA><PROC>
[2]
     <IDENT>
                                identification division <END INST> program-id.
                                 ID<END INST> author. WORDS<END INST>
                                 date-written. WORDS <END INST>
     <WORDS>
[3]
                                 ID <WORDS LR>
[4]
     <WORDS LR>
                                ID <WORDS LR>
[5]
[6]
     <END INST>
                                \cdot n
                                 environment division
<br/>
\langle {\rm END\_INST} \rangle configuration
[7]
     <ENV>
                                 section<END_INST> source-computer.
                                 WORDS<END INST> object-computer.
                                 WORDS<END INST>
                                 data division<END INST> working-storage
[8]
     <DATA>
                                 section<END INST> <VAR LIST>
[9]
     <VAR LIST>
                                 <VAR DECL> <VAR LIST>
[10]
     <VAR DECL>
                             \rightarrow <LEVEL> ID pic IMAGE <VAR DECL TAIL>
[11]
[12]
    <VAR_DECL_TAIL>
                                value INTEGER<END INST>
[13]
                                 <END INST>
[14]
     <LEVEL>
                                INTEGER
     <PROC>
                                 procedure division<END INST> ID
[15]
                                 section<END INST> <LABELS> end program ID.
                                 <LABEL><END_INST> <INSTRUCTION_LIST>
[16]
    <LABELS>
                                 <LABELS LR>
[17]
                                 <LABEL><END INST> <INSTRUCTION LIST>
     <LABELS LR>
                                 <LABELS LR>
[18]
                                \varepsilon
[19]
     <LABEL>
                                ID
[20]
     <INSTRUCTION_LIST>
                                <INSTRUCTION> <INSTRUCTION LIST>
[21]
[22]
     <INSTRUCTION>
                                <ASSIGNATION>
[23]
                             \rightarrow <IF>
[24]
                             \rightarrow <CALL>
[25]
                             \rightarrow <READ>
[26]
                                <WRITE>
[27]
                                stop run<END INST>
[28]
    <ASSIGNATION>
                                 move <EXPRESSION> to ID<END INST>
[29]
                                 compute ID = <EXPRESSION><END INST>
[30]
                                 add <EXPRESSION> to ID<END INST>
                                 subtract < EXPRESSION > from \ ID < END \quad INST >
[31]
[32]
                                multiply <ASSIGN END><END INST>
[33]
                                 divide <ASSIGN_END><END_INST>
[34]
    <ASSIGN END>
                                 <EXPRESSION>,<EXPRESSION> giving ID
[35]
     <EXPRESSION>

ightarrow <EXP AND> <EXPRESSION LR>
[36]
    <EXPRESSION LR>
                             \rightarrow or <EXP AND> <EXPRESSION LR>
[37]
                             \rightarrow
    \langle \text{EXP\_AND} \rangle

ightarrow <EXP EQUAL> <EXP AND LR>
[38]
[39]
    <EXP AND LR>
                                 and <EXP EQUAL> <EXP AND LR>
```

```
[40]
[41]
    <EXP EQUAL>
                            \rightarrow <EXP ADD> <EXP EQUAL LR>
[42]
    <EXP EQUAL LR>
                            \rightarrow = <EXP ADD> <EXP EQUAL LR>
[43]

ightarrow < <EXP ADD> <EXP EQUAL LR>
                            \rightarrow > <EXP ADD> <EXP EQUAL LR>
[44]
[45]
                               <= <EXP_ADD> <EXP_EQUAL_LR>
                               >= <EXP ADD> <EXP EQUAL LR>
[46]
[47]
    <EXP ADD>
                               <EXP MULT> <EXP ADD LR>
[48]
    <EXP\_ADD\_LR>
                            \rightarrow + <EXP MULT> <EXP ADD LR>
[49]
[50]
                            \rightarrow - <EXP MULT> <EXP ADD LR>
[51]
[52]
    <EXP_NOT> <EXP_MULT_LR>
                            [53]
    <EXP_MULT_LR>
                               / < EXP NOT > < EXP MULT LR >
[54]
                            \rightarrow
[55]
[56]
    <EXP_NOT>
                            \rightarrow -<EXP NOT>
[57]
                               not <EXP_NOT>
[58]
                               <EXP PARENTHESIS>
[59]
    <EXP PARENTHESIS>
                            \rightarrow (\langle EXPRESSION \rangle)
[60]

ightarrow <EXP TERM>
    [61]
                            \rightarrow ID
[62]
                               INTEGER
                            \rightarrow
[63]
                            \rightarrow true
[64]
                               false
[65]
    \langle IF \rangle
                               if <EXPRESSION> then <INSTRUCTION_LIST>
                                <IF END>
[66]
    <IF END>
                                else <INSTRUCTION LIST> end-if
[67]
                                end-if
[68]
    <CALL>
                                perform ID < CALL_TAIL>
                                until <EXPRESSION><END INST>
[69]
    <CALL_TAIL>
                               <END INST>
[70]
[71]
    <READ>
                                accept ID<END INST>
[72]
                                display <WRITE TAIL>
    <WRITE>
[73]
    <WRITE_TAIL>
                               <EXPRESSION><END_INST>
[74]
                                STRING<END INST>
```

| Variable | First^1 | $Follow^1$ |
|---------------------|--------------------|-------------|
| <program></program> | identification | |
| <IDENT $>$ | identification | environment |
| <WORDS $>$ | ID | |
| $<$ WORDS_LR $>$ | ID, ε | |

| <end_inst></end_inst> | | program-id, date-written, environment, configuration, source-computer, object-computer, data, working-storage, INTEGER, ε , ID, move, compute, add, substract, multiply, divide, if, perform, accept, display, stop |
|---|--|---|
| <env></env> | environment | data |
| <data></data> | data | procedure |
| <var list=""></var> | INTEGER, ε | - |
| <var_list></var_list> | INTEGER, E INTEGER | procedure |
| _ | | INTEGER, ε |
| <var_decl_tail></var_decl_tail> | value, . | INTEGER, ε |
| <level></level> | INTEGER | ID |
| <proc></proc> | procedure | 1 |
| <labels></labels> | ID | end |
| <labels_lr></labels_lr> | ID, ε | end |
| <label></label> | ID | · |
| <instruction_list></instruction_list> | move, compute, add, substract, multiply, divide, if, perform, accept, display, stop, ε | ID, ε |
| <instruction></instruction> | move, compute, add, substract, | move, compute, add, substract, mul- |
| | multiply, divide, if, perform, | tiply, divide, if, perform, accept, dis- |
| | accept, display, stop | play, stop, ε |
| <assignation></assignation> | move, compute, add, substract, | move, compute, add, substract, mul- |
| (118818111111111) | multiply, divide | tiply, divide, if, perform, accept, dis- |
| | maroipi, arvide | play, stop, ε |
| <assign end=""></assign> | -, not, (, ID, INTEGER, true, false | piaj, stop, c |
| <expression></expression> | -, not, (, ID, INTEGER, true, false | to, ., from, , , giving,), then |
| <expression lr=""></expression> | or, ε | to, ., from, , , giving,), then |
| <exp and=""></exp> | -, not, (, ID, INTEGER, true, false | or, ε |
| <exp and="" lr=""></exp> | and, ε | or, ε |
| <exp equal=""></exp> | -, not, (, ID, INTEGER, true, false | and, ε |
| <u> </u> | | _^ |
| <exp_equal_lr> <exp_add></exp_add></exp_equal_lr> | $=, <, >, <=, >=, \varepsilon$ -, not, (, ID, INTEGER, true, false | and, ε |
| <exp_add_lr></exp_add_lr> | $+$, $-$, ε | |
| <exp mult=""></exp> | | $=,<,>,<=,>=,\varepsilon$ |
| <u> </u> | -, not, (, ID, INTEGER, true, false | $+$, -, ε |
| <exp_mult_lr></exp_mult_lr> | *, /, ε | +, -, ε * / - |
| <exp_not></exp_not> | -, not, (, ID, INTEGER, true, false | *, /, ε |
| <exp_parenthesis></exp_parenthesis> | (, ID, INTEGER, true, false | *, /, ε |
| <exp_term></exp_term> | ID, INTEGER, true, false | *,/,arepsilon |
| <if></if> | if | move, compute, add, substract, mul- |
| | | tiply, divide, if, perform, accept, dis- |
| ID DND | 1 1.0 | play, stop, ε |
| <if_end></if_end> | else, end-if | move, compute, add, substract, mul- |
| | | tiply, divide, if, perform, accept, dis- |
| | | play, stop, ε |
| | | |

<CALL>perform move, compute, add, substract, multiply, divide, if, perform, accept, display, stop, ε <CALL TAIL>until, . move, compute, add, substract, multiply, divide, if, perform, accept, display, stop, ε <READ> move, compute, add, substract, mulaccept tiply, divide, if, perform, accept, display, stop, ε <WRITE>display move, compute, add, substract, multiply, divide, if, perform, accept, display, stop, ε <WRITE_TAIL>STRING, -, not, (, ID, INTEGER, move, compute, add, substract, multrue, false tiply, divide, if, perform, accept, display, stop, ε