```
/SYNTAX VERIFIER VARIABLES
SUA=BTEM+5
SUD=BTEM+7
SJE=BTEM+10
SJF=BTEM+11
SUG=BTEM+12
SUK=BTEM+13
SUP=BTEM+14
SUQ=BTEM+15
SUR=BTEM+16
SUS=BTEM+17
SUT=BTEM+20
```

```
/STV IOT AND SUBR
STV,       LAW IR1
,STV,      LIO I (USERAC)
,STVJ,     DIO D
           DAP I (BTEM)
           LAC I (USERPC)
           DAC I (BTEM+1)
           LAC I (FSA)
           DAC G               /TEXT PTR
           DAC I (SUD)
           LAC B0
           DAC I (SUT)         /MULTIPLIER
           LAC I (SUQ)
           DAC I (SUK)
           DAC C               /OUTPUT PTR
           LAC I (SUP)
           DAC I (SUF)         /PUSHDOWN PTR

/MAIN LOOP FOR INTERPRETATION
STVML,     LIO I D             /GET PSEUDO INSTR
           ISP I (SUT)
           IDX D               /IDX PC CONTINGENT ON MULTIPLIER
           DIO E               /SAVE CURRENT PSEUDO INSTR
           CLA
           RCL 3S
           ADD (LAC STVDTB)
           DAC A
B2,        XCT A               /CONSTANT 100000, DISPATCH LOOKUP
           SZL I
           IDA                 /ADD 1 IF IN WIN MODE
           DAC A
           CLA↑USWP↑UCLF 7                 /CLEAR ALL FLAGS EXCEPT LINK
           RCL 3S
           IFI                 /SET FLAGS 4,5,6 FROM BITS 3,4,5
           LIO E
           JMP A               /DISPATCH JMP

/IDX OPCODE
STV44,     DZM I (BTEM+5) /GROUP COUNTER
           JSP STVTRA          /GET TEXT PTR INTO E
           LAC G
           DAC I (BTEM+4) /SPACE FOR RESTORING
           DAC I (BTEM+2) /FARTHEST SUCCESSFUL SO FAR
STV4A,     JSP STVLE           /LCH I E; DAC A; SKIP ON NOT EOM
           JMP STV4C           /EOM; SUCCEEDED
STV4A1,    LAC G
           SAD I (BTEM+2)
           STF 1               /THIS WILL BE A NEW RECORD IF IT WINS
           JSP ATM             /LCH I G; DAC F
           LAC A
           SAD F
           JMP STV4A           /CHAR'S AGREE, LOOP
STV4B,     JSP STVLE           /FAILURE, WAIT FOR EOM
           JMP STV4D           /FOUND, TRY NEXT CASE
           JMP STV4B
```

```
STV4C,      SZF I 1             /SUCCESS; IS THIS A NEW RECORS?
            JMP STV4D           /NO, IGNORE
            STF 2               /SET WIN FLAG
            LAC G
            DAC I (BTEM+2) /REMEMBER HIS TEXT PTR
            LAC I (BTEM+5)
            DAC I CBTM3         /REMEMBER HIS GROUP
STV4D,      LAC I (BTEM+4) /ROUTINE TO TRY NEXT CASE
            DAC G
            CLF 1
            JSP STVLE
            JMP STV4D1          /2 EOM'S IN A ROW; NEW GROUP
            JMP STV4A1          /GO INTO CHAR, LOOP
STV4D1,     IDX I (BTEM+5)
            JSP STVLE
            JMP STV4E           /3 EOM'S IN A ROW; FINISHED
            JMP STV4A1

STV4E,      SZF I 2             /WAS THERE A WINNER?
            JMP STVLOS
            LAC I (BTEM+2) /YES, REMEMBER HIS PTR
            DAC G
            LAC I CBTM3         /SAVE HIS GROUP #
STV14,      DAC I C
            IDX C
STV4,       JMP STVML           /ALL DONE; ALSO LOSE MODE
            SZF 4
            JMP STV44           /IDX OPCODE
            LAI                 /?
            SCM!UIDA            /ADD OPCODE, CALCULATE MULTIPLIER
            DAC I (SUT)
STVØ,       JMP STVML
            JSP ATM
            JSP REATM           /CHECK CHAR, AGAINST ATOM TABLE
            JMP STVML           /OK
STVLOS,     CLL!UCML            /NG, ENTER LOSE MODE
            LAC PØ
            DAC I (SUT)         /CLOBBER MULTIPLIER
STV5,       JMP STVML
STV5L,      JSP ATM
            JSP REATM           /CHECK CHAR, AGAINST ATOM TABLE
            JMP STV5L           /OK, KEEP CYCLING
STV5UL,     LAC I (SUD)
            IDC
            DAC E
STV5UM,     LAW I 1
            ADD G
            IDC
            IDC
            DAC G
            IDC
            SAD E
            JMP STVML
            LCH G
            SAD ML6
            JMP STV5UM
```

```
STV1,       JMP STVML
            JSP STVTRA      /GET TEXT PTR
            LAC G
            DAC I (FSA)     /UPDATE FSA SO IT CAN BE REFERRED TO
            LAC I E         /GET ARG
            SZF 4
            JMP STV14       /DCH OPCODE; SAVE ARG
STV1L,      JSP STVLE
             JMP STVML      /EOM, XCT OPCODE SUCCEEDED
            JSP ATM
            LAC A
            SAD F
            JMP STV1L       /KEEP INSISTING ON EXACT MATCH
            JMP STVLOS      /NO MATCH, FAIL

STV2,       JMP STV2L       /LAC IN LOSE MODE
            LAW 2           /LAC IN WIN MODE
            AND E
            SZA
            JMP STV2SP      /SUPER POP
STVPOP,     LAC I (SUF)     /NORMAL POP
            SAD I (SUP)
            JMP STVNPP      /TOP LEVEL, CAN'T POP
            SUB (3)
            DAC I (SUF)     /UPDATE PUSHDOWN PTR
            DAC A           /TEMPORARY PUSHDOWN PTR
            LAC I A
            DAC I (SUD)     /POP SUD
            IDX A
            LAC I A
            CLI             /PREPARE TO UNPACK POP SUT;SUK; PSEUDO PC
            SCR 6S
            SZL
            CLA             /NO MULTIPLIER IN LOSE MODE
            XOR BØ
            DAC I (SUT)
            RIL 6S
            IDX A
            LAC I A
            RCL 6S
            DIO I (SUK)
            RAR 6S          /PSEUDO PC TO AC
            JMP STV7I       /PSEUDO GO
```

```
STVNPP,  LAC I (BTEM)
         SZL
         JMP STVRUR     /GIVE R1 ON CAN'T LOSE POP
         JSP ATM
         LAC F
         SAS ML4
         JMP STVWL      /NOT EOM, LOSE THEN LAC
STV2SP,  IDX I (BTEM)   /GIVE R2
STVRUR,  DAP H          /SUBR TO RESET USER'S REGISTERS
         LAC C
         DAC I (SUG)    /OUTPUT PTR
         LAC D
         DAC I (SUA)    /PSEUDO PC
         LAC G
         DAC I (FSA)    /TEXT PTR
         LAC I (BTEM+1)
         DAC I (USERPC) /REAL PC (SOMETIMES REDUNDANT)
         JMP HEXIT

/DON'T LAW IN LOSE MODE
STV7L,   SZF I 4
         JMP STV7II
         DZM E          /LOOK LIKE LAC Ø
/WIN ENTRY TO LAC IN LOSE MODE
STVWL,   CLL↑UCML       /ENTER LOSE MODE
STV2L,   LAC I (SUK)    /RESTORE PTRS TO BEG OF LEVEL
         DAC C
         LAC I (SUD)
         DAC G
         LAC E
         SAD CLACH      /(LAC H)
         JMP STV2SP
         LAW 5
         AND E
         SAS B17        /STAY IN LOSE MODE ON LAC 1
         CLL            /ENTER WIN MODE ON LAC Ø; LAC 5
         LIO B17
         DIO E          /LOOK LIKE LAC 1 ON RECYCLE (LAC 5)
         SZA
         JMP STVPOP     /POP ON LAC 1; LAC 5
```

```
STV3,       JMP STVML
            LAC I (SUF)     /PUSH OPCODE (DIP)
            SAD I (SUE)
            JMP STVLOS      /CAN'T PUSH, LOSE
            DAC A           /TEMPORARY PUSHDOWN PTR
            LAC I (SUD)     /PUSH SUD
            DAC I A
            IDX A
            LAC I (SUT)     /PACK PUSH SUT, SUK, PSEUDO PC
            LIO I (SUK)
            RIL 6S
            RCL 6S
            DAC I A
            IDX A
            LAC D
            RCL 6S
            RAR 6S
            DAC I A
            IDX A
            DAC I (SUF)     /UPDATE PUSHDOWN PTR
            LAC C
            DAC I (SUK)     /UPDATE "BEGINNING OF LEVEL" PTRS
            LAC G
            DAC I (SUD)
            LAC B0
            DAC I (SUT)     /CLOBBER MULTIPLIER
            JMP STV7II      /NOW DO A "LAW" OPCODE

STV7,       JMP STV7L
STV7II,     JSP STVTRA
            LAC E
STV7I,      DZM D
            DAP D
STV6,       JMP STVML
            JSP STVTRA
            JSP STVRUR
            LAC E
            JMP GO

/STVL AND STVW IOTS
STVL,       CML
            LAC B0
            DAC I (SUT)
STVW,       LAC I (FSA)     /UNRESTORE USER REGISTERS
            DAC G
            LAC I (SUG)
            DAC C
            LAC I (SUA)
            JMP STV7I
```

```
/ATOM TABLE SUBROUTINES
/ENTRY TO GET A CHAR, AND SKIP ALL IRRELEVANT (SUS) CHAR,S
ATM,      SUB B17
          DAP H
          LIO I (SUS)     /PUT MASK IN IO
          LCH I G
          SAD ML6
          JMP ATMW        /WARNING
ATMR0,    DAC F           /SAVE CHAR,
          RAL 6S
ATMI,     ADD (LAC ATOM)
          DAC B
          XCT B           /LOOK UP ATOM TABLE ENTRY
          SZA I
          JMP XHEXIT      /R2, EOM ALWAYS FAILS
ATMWC,    DIO B           /SAVE MASK
          RIL 3S          /PUT CLASS BIT IN SIGN OF IO
          XAI
          SPA
          JMP ATMD        /DIFFERENT CLASSES
          XAI             /RESTORE AC
          AND B
          AND (37777)
          SZA I
XHEXIT,   IDX H           /NO BITS THE SAME, FAIL (R2)
          JMP HEXIT       /OK, R1

ATMD,     RIL 1S          /MOVE "OTHER CLASS" BIT INTO SIGN OF IO
          SPI I
          IDX H
          JMP HEXIT

ATMW,     LCH I G
          SAD ML4
          JMP ATMR0       /TREAT RUBOUT LIKE EOM
          RAR 6S
          IOR ML6
          DAC F           /12BIT CHAR, IN F
ATMWI,    LAC (404000)    /ATOM TABLE ENTRY FOR 12 BIT CHAR,S
          JMP ATMWC

/ENTRY TO TEST CHAR, IN F AGAINST MASK IN E
REATM,    DAP H
          CLA
          LIO F
          RCL 6S
          LIO E
          SNI
          SAS (74)
          JMP REATM1
          JMP STV5UL
REATM1,   SAS (77)
          JMP ATMI
          JMP ATMWI
```

```
/ROUTINE TO LCH I E, DAC A AND SKIP ON NOT EOM
STVLE,    DAP H
          LCH I E
          SAD ML6
          JMP STVLEW      /WARNING
STVLEI,   DAC A
          SAS ML4
          IDX H           /NOT EIM
          JMP HEXIT
STVLEW,   RCL 6S
          LCH I E
          RCR 6S
          JMP STVLEI      /12 BIT CHAR, IN A

/RELOCATABLE TRACING SUBROUTINE FOR STV
STVTRA,   DAP H
          LAW 7777        /TOP 6 BITS Ø FIRST TIME THROUGH
          AND E
          LIO E
STVTRL,   RIL 4S
          SPI
          ADD I (SUR)
          DAC E
          RIL 1S
          SPI I
          JMP HEXIT
          LAC I E
          AND (6Ø7777
          LIO I E
          JMP STVTRL

/OPCODE DISPATCH TABLE
STVDTB,   JMP STVØ
          JMP STV1
          JMP STV2
          JMP STV3
          JMP STV4
          JMP STV5
          JMP STV6
          JMP STV7
```

```
/ATOM TABLE, SIGN BIT HAS CLASS = Ø MEANS EOM
ATOM,B13, 2Ø              /SPACE
          4ØØØ1Ø          /!
B5,       1ØØØØ           /"
          41ØØØØ          /#
          41ØØØØ          /$
          41ØØØØ          /%
          41ØØØØ          /&
B8,       1ØØØ            /'
          4ØØØ2Ø          /(
          4ØØØ4Ø          /)
          4ØØØØ1          /*
B11,      1ØØ             /+
B7,       2ØØØ            /,
B12,      4Ø              /-
CJORG,
B1Ø,      2ØØ             /.
          4ØØØØ2          //
B16,      REPEAT 1Ø., 2   /DIGITS
          4ØØØØ4          /:
B6,       4ØØØ            /;
          4ØØ4ØØ          /<
          4Ø1ØØØ          /=
          4Ø2ØØØ          />
B14,      1Ø              /?
ATOM+4Ø,  41ØØØØ          /@
B9,       4ØØ             /A
B17,      REPEAT 3, 1     /BCD
4ØØ       REPEAT 3, 1     /E FGH
4ØØ       REPEAT 5, 1     /I JKLMN
4ØØ       REPEAT 5, 1     /O PQRST
4ØØ       REPEAT 3, 1     /U VWX
4ØØ       1               /Y Z
          4ØØ1ØØ          /[
          Ø               /EOM
          4ØØ2ØØ          /]
B15,      4               /CRLF
ATOM+77,  /NO ENTRY FOR 77

START
```