```
1                       DEFINE( *NEXTITEM(NULL)*,*NEXTITEM* )
2                       DEFINE( *NEWSTATE(OLD,TKN)*,*NEWSTATE* )
3                       DEFINE( *NEXTTKN(NULL)*,*NEXTTKN* )
4                       DEFINE( *NEXTFNC(NULL)*,*NEXTFNC* )
5                       DEFINE( *EXTEND(KEY)*,*EXTEND* )
6                       DEFINE( *EXTENDSS(NULL)FLAG,STATE,TKN,NSTATE,XSTATE* )
7                       DEFINE( *ASSEM(LOC,OP,RAND,COMM)A,B* )
8                       DEFINE( *SAVE(NULL)* )
9                       DEFINE( *SPUT(ITEM)TEMP* )
10                      DEFINE( *PUT(ITEM)* )
                *
11                      DEFINE( *DUMPFNC(NULL)*,*DUMPFNC* )
12                      DEFINE( *DUMPTKN(NULL)*,*DUMPTKN* )
13                      DEFINE( *DUMPSTATET(NULL)*,*DUMSTATET* )
14                      DEFINE( *DUMPRULES(NULL)*,*DUMPRULES* )
15                      DEFINE( *DUMPNONTERM(NULL)*,*DMPNONTERM* )
16                      DEFINE( *PRINTSS(SS,FLAG)NM,S* )
17                      DEFINE( *PRINTRULES(NULL)LINE* )
18                      DEFINE( *PRINTS(S)NM,LN,T* )
19                      DEFINE( *PRINTTKNS(NULL)* )
                *
                #
20                      DATA( *LISTEL(VALUE,NEXT)* )
                *
21                      F1 = *F1,*
22                      F2 = *F2,*
23                      F3 = *F3,*
24                      T1 = *T1,*
25                      T2 = *T2,*
26                      T3 = *T3,*
27                      T4 = *T4,*
28                      T5 = *T5,*
29                      T6 = *T6,*
30                      T7 = *T7,*
31                      T8 = *T8,*
32                      T9 = *T9,*
33                      T10 = *T10,*
34                      T11 = *T11,*
35                      S2 = *S2,*
36                      S4 = *S4,*
37                      S5 = *S5,*
38                      R4 = *R4,*
39                      R5 = *R5,*
40                      R6 = *R6,*
41                      R7 = *R7,*
42                      R8 = *R8,*
43                      SS1 = *SS1,*
44                      SS2 = *SS2,*
45                      SS4 = *SS4,*
46                      SS5 = *SS5,*
                *
47                      DETACH( *INPUT* )
48                      INPUT( *INPUT*, *GRAMMAR*, 80 )
49                      OUTPUT( *ASSEMLINE*, *SCMMOD*, ** )
```

```
50                      OUTPUT( *SAVELINE*, *SAVEFF*, ** )
                *
                *
51                      SYNERR = ****** SYNTAX ERROR, *
                *
                *
                *
52                      :(GO)
                *
53              NEXTTKN ITEM = NEXTITEM()
54                      IDENT( ITEM, *.* )  :S(FRETURN)
55                      NEXTTKN = $( T1 ITEM )
56                      IDENT( NEXTTKN, ** )  :F(RETURN)S(FAIL1)
                *
57              NEXTFNC ITEM = NEXTITEM()
58                      IDENT( ITEM, *.* )  :S(FRETURN)
59                      NEXTFNC = $( F1 ITEM )
60                      IDENT( NEXTFNC, ** )  :F(RETURN)S(FAIL2)
                *
61              NEXTITEM LINE  SPAN(* *) BREAK(* *) . NEXTITEM = **  :S(RETURN)
62              NEXTITEM1 LINE = INPUT
63                      OUTPUT = LINE
64                      LINE  POS(0) ***     :F(NEXTITEM)S(NEXTITEM1)
                *
65              NEWSTATE NEWSTATE = $( S4 OLD *.* TKN )
66                      IDENT( $(S4 OLD *.* TKN), ** )  :F(RETURN)
67                      STATECNT = STATECNT + 1
68                      $(S2 STATECNT ) = $( S2 OLD) *.* TKN
69                      $( S4 OLD *.* TKN ) = STATECNT
70                      NEWSTATE = STATECNT  :(RETURN)
                *
                *
                *
                *                       EXTEND A LIST BY TRANSITIVITY THROUGH NON TERMINALS
                *
71              EXTEND  TKN = TERMTKNCNT + 1
72              EXTEND1 GT( TKN, TKNCNT )  :S(EXTEND2)
73                      $(T6 TKN) = *X*
74                      TKN = TKN + 1  :(EXTEND1)
                *
75              EXTEND2 FLAG = *OFF*
76                      TKNHD = TERMTKNCNT + 1
                *
77              EXTEND3 GT(TKNHD, TKNCNT )  :S(EXTEND9)
78                      CHKTKN = $(T6 TKNHD )
79                      $(T6 TKNHD) = $(KEY TKNHD)
80                      TKN = $(KEY TKNHD)
                *
81              EXTEND4 IDENT( TKN, CHKTKN)  :S(EXTEND8)
82                      LE( TKN, TERMTKNCNT)  :S(EXTEND7)
83                      SUBTKN = $(KEY TKN)
                *
84              EXTEND5 IDENT(SUBTKN, *X* )  :S(EXTEND7)
85                      IDENT( $(KEY TKNHD *.* SUBTKN), ** )  :F(EXTEND6)
```

```
86                  $(KEY TKNHD ≠,≠ SUBTKN) = $(KEY TKNHD)
87                  $(KEY TKNHD) = SUBTKN
88                  FLAG = ≠ON≠
                *
89          EXTEND6 SUBTKN = $(KEY TKN ≠,≠ SUBTKN)          :(EXTEND5)
                *
90          EXTEND7 TKN = $(KEY TKNHD ≠,≠ TKN )   :(EXTEND4)
                *
91          EXTEND8 TKNHD = TKNHD + 1   :(EXTEND3)
                *
92          EXTEND9 IDENT( FLAG, ≠OFF≠ )          :S(RETURN)F(EXTEND2)
                *
                *
                *                   EXTEND STATE SET REPRESENTED IN ARRAY SET
                *                   BY LAMBDA MOVES
                *
                *                   SETS SET AND SETX EMPTY AT END
                *                         (ASSUMES SO AT START FOR SETX )
                *
                *                   RETURNS NUMBER OF EXTENDED SET
                *
93          EXTENDSS FLAG = ≠OFF≠
94                  STATE = 1
                *
95          EXTENDSS1 GT( STATE, STATECNT) :S(EXTENDSS5)
96                  IDENT( SET[STATE], ≠≠ )   :S(EXTENDSS4)
97                  IDENT( SETX[STATE], ≠1≠ ) :S(EXTENDSS4)
98                  SETX[STATE] = 1
99                  TKN = TERMTKNCNT + 1
                *
100         EXTENDSS2 GT(TKN, TKNCNT)     :S(EXTENDSS4)
101                 NSTATE = $(S4 STATE ≠,≠ TKN )
102                 IDENT( NSTATE, ≠≠ )     :S(EXTENDSS3)
103                 XSTATE = $(S4 0 ≠,≠ TKN )
104                 IDENT( SET[XSTATE], 1 )   :S(EXTENDSS3)
105                 SET[XSTATE] = 1
106                 FLAG = ≠ON≠
                *
107         EXTENDSS3 TKN = TKN + 1       :(EXTENDSS2)
                *
108         EXTENDSS4 STATE = STATE + 1   :(EXTENDSS1)
                *
109         EXTENDSS5 IDENT( FLAG, ≠OFF≠ )          :F(EXTENDSS6)
110                 STATE = 1
111                 SSTATE = ≠≠
                *
112         EXTENDSS6 GT( STATE , STATECNT)  :S(EXTENDSS8)
113                 IDENT( SET[STATE], ≠≠ )        :S(EXTENDSS7)
114                 SSTATE = SSTATE ≠,≠ STATE
                *
115         EXTENDSS7 SET[STATE] = ≠≠
116                 SETX[STATE] = ≠≠
117                 STATE = STATE + 1  :(EXTENDSS6)
                *
```

```
118     EXTENDSSR EXTENDSS = $( SS1  SSTATE)
119               IDENT( EXTENDSS, ## )    :F(RETURN)
120               SSCNT = SSCNT + 1
121               EXTENDSS = SSCNT
122               $( SS1 SSTATE) = SSCNT
123               $(SS2 SSCNT ) = SSTATE     :(RETURN)
          *
          *
          *                    OUTPUT AN ASSEMBLY LINE
          *
          *
124     ASSEM     A = LOC #                    #
125               A TAB(10) . B  = ##
126               A = R OD #                    #
127               A  TAB(20) . B  = ##
128               IDENT( COMM, ## )    :S(ASSEM1)
129               A = B RAND #                                    #
130               A  TAB(50) . B  = ##
131               ASSEMLINE = B COMM   :(RETURN)
          *
132     ASSEM1    ASSEMLINE = B RAND   :(RETURN)
          *
          *
          *                  SUBROUTINE TO SAVE INFO FOR SECOND PASS
          *
135     SAVE      REWIND( #SAVEF# )
134               PUT( TERMTKNCNT )
135               PUT( TKNCNT )
136               PUT(FILECNT )
137               PUT( FNCCNT )
138               PUT( STATECNT )
          *
139               I = 1
          *
140     SAVE1     GT( I, FNCCNT )     :S(SAVE2)
141               SPUT( F2 I )
142               SPUT( F3 I )
143                I = I + 1            :(SAVE1)
          *
144     SAVE2     I = 1
          *
145     SAVE3     GT( I, TERMTKNCNT)  :S(SAVE4)
146               SPUT( T2 I )
147               SPUT( T3 I )
148               SPUT( T4 I )
149               SPUT( T11 I )
150               PUT( T10 I )
151               PUT( #X# )
152               I = I + 1      :(SAVE3)
          *
153     SAVE4     GT( I, TKNCNT )     :S(SAVE7)
154               SPUT( T2 I )
155               SPUT( T3 I )
156               SPUT( T4 I )
```

```
157                 K = TIO I
158                 SPUT( K )
159                 J = $(K)
                *
160         SAVE5   IDENT( J, #X# )         :S(SAVE6)
161                 L = K #,# J
162                 SPUT(L)
163                 J = $(L)    :(SAVE5)
                *
164         SAVE6   I = I + 1   :(SAVE4)
                *
165         SAVE7   I = 1
                *
166         SAVE8   GT( I, RULECNT)         :S(SAVE9)
167                 SPUT( R4 I )
168                 SPUT( R5 I )
169                 SPUT( R6 I )
170                 I = I + 1               :(SAVE8)
                *
171         SAVE9   I = 0
                *
172         SAVE10  GT( I, STATECNT )       :S(SAVE13)
173                 SPUT( S2 I )
174                 SPUT( S4 I )
175                 SPUT( S5 I )
                *
176                 J = 1
                *
177         SAVE11  GT(J, TKNCNT)           :S(SAVE12)
178                 SPUT( S4 I #,# J )
179                 J = J + 1               :(SAVE11)
                *
180         SAVE12  I = I + 1               :(SAVE10)
                *
181         SAVE13  PUT( #,,,,,,# )
182                 OUT( *ITEMS OUT = #  ITEMCNT )          :(RETURN)
                *
                *
                *                   ROUTINES USED BY SAVE
                *
                *
183         PUT     SAVELINE = ITEM
184                 ITEMCNT = ITEMCNT + 1           :(RETURN)
                *
                *
185         SPUT    TEMP = $(ITEM)
186                 IDENT( TEMP, ## )   :S(RETURN)
187                 PUT( ITEM)
188                 PUT( TEMP )         :(RETURN)
                *
                *
                *
                *
                *
```

```
*
*
*
*                       DUMP FUNCTION DATA
*
189     DUMPFNC   I = 1
190               OUTPUT = ##
191               OUTPUT = #      FUNCTION TABLE#
192               OUTPUT = ##
193     DUMPFNC1  GT( I, FNCCNT)          :S(RETURN)
194               OUTPUT = I # # $(F2 I ) # # $(F3 I) # # $(F1 $(F2 I))
195               I = I + 1 :(DUMPFNC1)
*
*
*                       DUMP TOKEN TABLE
*
196     DUMPTKN   OUTPUT = ##
197               OUTPUT = ##
198               OUTPUT = #      TOKENS TERMINAL #
199               OUTPUT = ##
200               I=1
201     DUMPTKN1  GT( I, TERMTKNCNT)  :S(DUMPTKN2)
202               PART = I # # $(T2 I) # # $(T3 I) # # $(T4 I) #  # $(T11 I )
203               OUTPUT = PART # # $(T1 $(T2 I ))
204               I= I + 1  :(DUMPTKN1)
*
205     DUMPTKN2  OUTPUT = ##
206               OUTPUT = #       TOKENS   NON TERMINAL#
207               OUTPUT = ##
*
208     DUMPTKN3  GT( I, TKNCNT )  :S(RETURN)
209               OUTPUT = I # # $(T2 I) # # $(T3 I) # # $(T1 $(T2 I))
210               I = I + 1 :(DUMPTKN3)
*
*
211     DUMPSTATET          OUTPUT = ##
212                         OUTPUT = ##
213               OUTPUT = #      STATE TABLE#
214               I=0
*
215     DUMPSTATET1         GT( I, STATECNT) :S(RETURN)
216               OUTPUT = I # # $(S2 I) # # $(S5 I)
217               J = 1
218     DUMPSTATET2         GT(J, TKNCNT) :S(DUMPSTATET4)
219               IDENT($(S4 I #,# J), ## )  :S(DUMPSTATET3)
220               OUTPUT = #                 # J # # $(S4 I #,# J )
221     DUMPSTATET3         J = J + 1  :(DUMPSTATET2)
222     DUMPSTATET4         I = I + 1   :(DUMPSTATET1)
*
*
223     DUMPRULES OUTPUT = ##
224               OUTPUT = ##
225               OUTPUT = #      RULE TABLE#
226               OUTPUT = ##
```

```
227                 I = 1
228     DUMPRULES1          GT( I, RULECNT)    :S(RETURN)
229                 OUTPUT = I ≠ ≠ $(R4 I) ≠ ≠ $(R5 I) ≠ ≠ $(R6 I)
230                     OUTPUT = ≠            ≠ $(R7 I)
231                   E = $(R8 I)
232     DUMPRULES2 IDENT(E, ≠≠)    :S(DUMPRULES3)
233                 OUTPUT = ≠              ≠ VALUE(E)
234                 E = NEXT(E)   :(DUMPRULES2)
235     DUMPRULES3          I = I + 1    :(DUMPRULES1)
        *
        *
        *
        **                  DUMP NON TERMINAL EXTRA INFO
        *
236     DMPNONTERM          OUTPUT =≠≠
237                 OUTPUT = ≠≠
238                 OUTPUT = ≠       NON TERMINAL EXTRA INFO ≠
239                 OUTPUT = ≠≠
240                 I = TERMTKNCNT + 1
        *
241     DMPNONTERM1         GT( I, TKNCNT )      :S(RETURN)
242                 OUTPUT = I ≠ ≠ $( T2 I )
243                 OUTPUT = ≠         INITIALS≠
244                 J = $( T5 I )
245     DMPNONTERM1A        IDENT( J , ≠X≠ )     :S(DMPNONTERM2)
246                         OUTPUT = ≠          ≠ J
247                         J = $( T5 I ≠,≠ J ) :(DMPNONTERM1A)
        *
248     DMPNONTERM2         OUTPUT = ≠≠
249                 OUTPUT = ≠          INITIAL TERMINALS≠
250                 J = $(T7 I )
251     DMPNONTERM3                      IDENT( J, ≠X≠ )      :S(DMPNONTERM4)
252                         OUTPUT = ≠      ≠ J
253                         J = $(T7 I ≠,≠ J )   :(DMPNONTERM3)
254     DMPNONTERM4         OUTPUT = ≠≠
255                 OUTPUT = ≠      FINALS≠
256                 J = $(T9  I)
257     DMPNONTERM5         IDENT( J, ≠X≠)  :S(DMPNONTERM6)
258                         OUTPUT = ≠          ≠ J
259                         J = $(T8 I ≠,≠ J )  :(DMPNONTERM5)
        *
260     DMPNONTERM6         OUTPUT = ≠≠
261                 OUTPUT = ≠      FINAL NON TERMINALS≠
262                 J = $(T9 I)
263     DMPNONTERM7  IDENT(J , ≠X≠)   :S(DMPNONTERM8)
264                 OUTPUT = ≠      ≠ J
265                 J = $(T9 I ≠,≠ J)  :(DMPNONTERM7)
        *
266     DMPNONTERM8         OUTPUT = ≠≠
267                 OUTPUT = ≠≠
268                 OUTPUT = ≠       FOLLOWING TERMINALS≠
269                 OUTPUT = ≠≠
270                 J = $(T10 I)
271     DMPNONTERM9         IDENT(J, ≠X≠)   :S(DMPNONTERM10)
```

```
272                 OUTPUT = ≠          ≠ J
273                 J = $(T)0 I ≠,≠ J )    :(DMPNONTERM9)
                *
274         DMPNONTERM1  I = I + 1   :(DMPNONTERM1)
                *
                *
                *
                *                   PRINT OUT A STATE SET
                *
                *
275         PRINTSS     NM = $(SS2 SS) ≠,≠
276         PRINTSS1    NM ≠,≠ BREAK(≠,≠) , S = ≠≠   :F(RETURN)
277                     PRINTS(S)
278                     IDENT( FLAG, ≠≠ )    :S(PRINTSS1)
279                     OUT(OUTPUT )         :(PRINTSS1)
                *
                *
                *
                *                   PRINT OUT A STATE
                *
280         PRINTS      NM = $(S2 S) ≠,≠
281                     LN = ≠          ≠
282         PRINTS1     NM ≠,≠ BREAK(≠,≠) , T    = ≠≠    :F(PRINTS2)
283                     LN = LN ≠ ≠ $(T2 T)      :(PRINTS1)
284         PRINTS2     OUTPUT = LN   :(RETURN)
                *
                *
                *                   PRINT OUT RULE TABLE
                *
285         PRINTRULES              OUTPUT = ≠≠
286                     OUTPUT = ≠≠
287                     OUTPUT = ≠       RULE INFO≠
288                     OUTPUT = ≠≠
289                     I = 1
290         PRINTRULES1   GT(I, RULECNT)    :S(RETURN)
291                     LINE = ≠  ≠ I ≠     ≠ $(T2 $(R4 I)) ≠    ≠ $(F2 $(R5 I))
292                     LINE = LINE ≠     ≠ $(R6 I)
293                     OUTPUT = LINE
294                     I = I + 1     :(PRINTRULES1)
                *
                *
                *
                *                   PRINT OUT TOKEN INFO
                *                   AND PLACE IN ASSEMBLY ALSO
                *
295         PRINTTKNS OUTPUT = ≠≠
296                     OUTPUT = ≠≠
297                     OUTPUT = ≠     TOKEN DEFINITIONS≠
298                     OUTPUT = ≠≠
299                     I = 1
300         PRINTTKNS1  GT( I, TKNCNT )    :S(RETURN)
301                     TEMP = $(T2 I) ≠                              ≠
302                     TEMP   TAB(24)  .  TEMP1  = ≠≠
303                     OUTPUT = TEMP1  $(T3 I)
```

```
304                ASSEM( ≠XXX≠, ≠SET≠, $(T3 I), $(T2 I) )
305                I = I + 1       :(PRINTTKNS1)
           *
           *
           *
           *
           *
           *
           *
           *                    ACTUAL CODE STARTS HERE
           *
           *                    FIRST READ IN TOKEN DEFINITIONS, TERMINAL FIRST
           *
           *
           *
306        GO      ASSEM( ≠≠, ≠IDENT≠, NEXTITEM() )
307                ASSEM( ≠≠, ≠EXT≠, ≠SUBGRAM≠ )
308                ASSEM( ≠≠, ≠EXT≠, ≠GETTKN≠ )
309                ASSEM( ≠≠, ≠EXT≠, ≠GETKYWD≠ )
310                ITEM= NEXTITEM()
311                IDENT( ITEM, ≠FL≠ )  :F(G01)
           *
312                ASSEM( ≠≠, ≠ENTRY≠, ≠FL≠ )
           *
313        G01     KEYBASE = NEXTITEM()
           *
314                TKNCNT = 0
315                ASSEM( ≠KEYWDL≠, ≠BSS≠, 0 )
           *
316        TKNS1   ITEM1 = NEXTITEM()
317                IDENT( ITEM1, ≠,≠ )  :S(TKNS2)
318                ITEM2 = NEXTITEM()
319                IDENT( ITEM2, ≠,≠ )              :S(FAILE7)
320                ITEM3 = NEXTITEM()
321                IDENT( ITEM3, ≠,≠ )              :S(FAIL7)
322                TKNCNT = TKNCNT + 1
323                IDENT( ITEM3, ≠GETKYWD≠ )        :F(TKNS1A)
           *
324                KEYCNT = KEYCNT + 1
325                ASSEM( ≠≠, ≠DATA≠, ≠DL≠ ITEM2 )
326                ASSEM( ≠≠, ≠VFD≠, ≠60/≠ KEYBASE ≠+≠ KEYCNT )
327                ITEM2 = KEYBASE ≠+≠ KEYCNT
328                $(T1 TKNCNT ) = ≠KEYWDL≠    :(TKNS1C)
           *
329        TKNS1A  IDENT( ITEM3, ≠SUBGRAM≠ )        :F(TKNS1B)
           *
330                ITEM4 = NEXTITEM()
331                IDENT( ITEM4, ≠,≠ )              :S(FAIL7)
332                ASSEM( ≠≠, ≠EXT≠, ITEM4 )
333                $(T1 TKNCNT ) = ITEM4            :(TKNS1C)
           *
334        TKNS1B  $(T1 TKNCNT) = 0    :(TKNS1C)
           *
335        TKNS1C  IDENT( NEXTITEM() , ≠.≠ )        :F(FAIL4)
```

```
336                     $(T1 ITEM1) = TKNCNT
337                     $(T2 TKNCNT) = ITEM1
338                     $(T3 TKNCNT) = ITEM2
339                     $(T4 TKNCNT) = ITEM3
340                     $(T7 TKNCNT) = TKNCNT
341                     $(T7 TKNCNT '.' TKNCNT ) = 'X'    :(TKNS1)
                *
                *
                *                   NOW WE READ IN NON TERMINAL TOKEN DEFINITIONS
                *
342     TKNS2   TERMTKNCNT = TKNCNT
343                     ASSEM( '*', 'DATA', 0 )
                *
344     TKNS3   ITEM1 = NEXTITEM()
345                     IDENT( ITEM1, '.' )   :S(FNCS0)
346                     ITEM2 = '2000B+' ( TKNCNT + 1 = TERMTKNCNT )
347                     IDENT( NEXTITEM() , '.' )       :F(FAIL4)
348                     TKNCNT = TKNCNT + 1
349                     $(T1 ITEM1) = TKNCNT
350                     $(T2 TKNCNT) = ITEM1
351                     $(T3 TKNCNT) = ITEM2
352                     $(T5 TKNCNT) = 'X'
353                     $(T7 TKNCNT) = 'X'
354                     $(T8 TKNCNT) = TKNCNT
355                     $(T8 TKNCNT '.' TKNCNT) = 'X'
356                     $(T9 TKNCNT) = 'X'
357                     $(T10 TKNCNT) = 'X'   :(TKNS3)
                *
                *
                *
                *                   NOW READ IN FUNCTION DEFINITIONS
                *
358     FNCS0   FNCCNT = 0
                *
359     FNCS1   ITEM1 = NEXTITEM()
360                     IDENT(ITEM1, '.')    :S(FNCS2)
361                     ITEM2 = NEXTITEM()
362                     IDENT( NEXTITEM(), '.' )        :F(FAIL4)
363                     FNCCNT = FNCCNT + 1
364                     ASSEM( '*', 'EXT', ITEM2 )
365                     $(F1 ITEM1) = FNCCNT
366                     $(F2 FNCCNT) = ITEM1
367                     $(F3 FNCCNT) = ITEM2            :(FNCS1)
                *
368     FNCS2
                *
                *
                *
                *                   THIS CODE READS IN THE RULES
                *                   AND GENERATES THE SYMBLE STATE TABLE
                *
                *
369                     STATECNT = 0
370                     RULECNT = 0
```

```
371                 DMYEL = LISTEL( ##, ## )
372         RULE1   RULEFNC = NEXTFNC()  :F(RULE4)
373                 IDENT( NEXTITEM(), #:# )  :F(FAIL5)
374                 RULETKN = NEXTTKN()     :F(FAIL)
375                 IDENT( NEXTITEM(), #:=# )  :F(FAIL6)
376                 RULECNT = RULECNT + 1
377                 $(R4 RULECNT) = RULETKN
378                 $(R5 RULECNT) = RULEFNC
379                 STATE = 0
380                 STATE = NEWSTATE(STATE,RULETKN)
381                 RULESIZE = 0
382                 EL = DMYEL
383                 NEXT(EL) = ##
384         RULE2   TKN = NEXTTKN()  :F(FAIL7)
385                 IDENT( $(T5 RULETKN #.# TKN), # )  :F(RULE3)
386                 $(T5 RULETKN #.# TKN) = $(T5 RULETKN)
387                 $(T5 RULETKN) = TKN
388         RULE3   STATE = NEWSTATE(STATE,TKN)
389                 NEXT(EL) = LISTEL(TKN,##)
390                 EL = NEXT(EL)
391                 RULESIZE = RULESIZE + 1
392                 LASTTKN = TKN
393                 TKN = NEXTTKN()      :S(RULE3)
394                 IDENT( $(T8 RULETKN #.# LASTTKN), # )  :F(RULE3A)
395                 $(T8 RULETKN #.# LASTTKN) = $(T8 RULETKN)
396                 $(T8 RULETKN) = LASTTKN
397         RULE3A  $(S5 STATE) = RULECNT
398                 $(R6 RULECNT) = RULESIZE
399                 $(R7 RULECNT) = STATE
400                 $(R8 RULECNT) = NEXT(DMYEL)
401                 RAND = #12/0,18/# $(F3 RULEFNC ) #,12/# RULESIZE
402                 RAND = RAND #,18/# $(T3 RULETKN )
403                 ASSEM( #R,# RULECNT , #VFD#, RAND )      :(RULE1)
404         RULE4
405                 PRINTTKNS()
406                 PRINTRULES()
407                 OUTPUT = ##
408                 OUTPUT = ##
409                 OUTPUT = ##
410                 EXTEND(T5)
411                 EXTEND(T8)
```

```
*
*
*
*                    CONSTRUCTION OF LIST OF TERMINALS THAT
*                    CAN BEGIN A NON TERM
*
412                  HDTKN = TERMTKNCNT + 1
413       FSTTERMS1  GT(HDTKN, TKNCNT)     :S(FSTTERMS5)
414                  TKN = $(T5 HDTKN)
415       FSTTERMS2  IDENT( TKN, '(' )     :S(FSTTERMS4)
416                  GT(TKN, TERMTKNCNT)   :S(FSTTERMS3)
417                  $(T7 HDTKN '.' TKN ) = $(T7 HDTKN)
418                  $(T7 HDTKN) = TKN
419       FSTTERMS3  TKN = $(T5 HDTKN '.' TKN)  :(FSTTERMS2)
420       FSTTERMS4  HDTKN = HDTKN + 1     :(FSTTERMS1)
421       FSTTERMS5
*
*
*                    CONSTRUCTION OF LIST ON ONON TERMS THAT CAN
*                    END A NON TERMINAL
*
422                  HDTKN = TERMTKNCNT + 1
423       LASTNONT1  GT(HDTKN, TKNCNT )    :S(LASTNONT5)
424                  TKN = $(T8 HDTKN)
425       LASTNONT2  IDENT( TKN, '(' )     :S(LASTNONT4)
426                  LE( TKN, TERMTKNCNT)          :S(LASTNONT3)
427                  $(T9 HDTKN '.' TKN ) = $(T9 HDTKN)
428                  $(T9 HDTKN) = TKN
429       LASTNONT3  TKN = $(T8 HDTKN '.' TKN)  :(LASTNONT2)
430       LASTNONT4  HDTKN = HDTKN + 1     :(LASTNONT1)
431       LASTNONT5
*
*
*
*
*                    CONSTRUCT LIST OF TERMINALS THAT
*                    CAN FOLLOW EACH NON TERM
*
432                  RULE = 1
433       NXTTERMS1  GT(RULE, RULECNT)     :S(NXTTERMS9)
434                  L = $(R& RULE)
*
435       NXTTERMS2  NONTERM = VALUE(L)
436                  L = NEXT(L)
437                  LE( NONTERM, TERMTKNCNT)      :S(NXTTERMS8)
*
438                  IDENT( L, '.')        :S(NXTTERMS7)
439                  FOLLOWER = VALUE(L)
*
440                  TKN1 = $(T9 NONTERM)
441       NXTTERMS3  IDENT( TKN1, '(' )    :S(NXTTERMS2)
*
442                  TKN2 = $(T7 FOLLOWER)
443       NXTTERMS4  IDENT( TKN2, '(' )    :S(NXTTERMS6)
```

```
444                IDENT( $(T10 TKN1 ≠.≠ TKN2 ), ≠≠ )  :F(NXTTERMS5)
445                $(T10 TKN1 ≠.≠ TKN2 ) = $(T10 TKN1)
446                $(T10 TKN1 ) = TKN2
                   *
447        NXTTERMS5 TKN2 = $( T7 FOLLOWER ≠.≠ TKN2 )     :(NXTTERMS4)
                   *
448        NXTTERMS6 TKN1 = $( T9 NONTERM ≠.≠ TKN1 )    :(NXTTERMS3)
                   *
449        NXTTERMS7 RULE = RULE + 1        :(NXTTERMS1)
                   *
450        NXTTERMS8 IDENT(L.≠≠)   :S(NXTTERMS7) F(NXTTERMS2)
                   *
451        NXTTERMS9
                   *
                   *
452                SAVE()
                   *
                   *
                   *
                   *                READ IN LIST OF TOKENS TO BE LOOKED UP
                   *                     IN THIS GRAMMAR
                   *
                   *
453        SPTKNS  TOKEN = NEXTTKN()    :F(SPTKNS1)
454                EXTNAME = NEXTITEM()
455                IDENT( EXTNAME, ≠.≠ )          :S(FAIL7)
456                IDENT( NEXTITEM(), ≠.≠ )       :F(FAIL4)
457                PUT( TOKEN )
458                PUT( EXTNAME )        :(SPTKNS)
                   *
459        SPTKNS1 PUT( ≠....≠ )
                   *
460                :(END)
                   *
                   *
461        FAIL1   OUT( SYNERR  ITEM ≠ NOT A TOKEN≠ ) :(FAIL)
462        FAIL2   OUT( SYNERR  ITEM ≠ NOT A FUNCTION≠ )  :(FAIL)
463        FAIL4   OUT( SYNERR  ≠ PERIOD EXPECTED≠ ) :(FAIL)
464        FAIL5   OUT( SYNERR  ≠≠ ≠:≠ EXPECTED ≠≠ )  :(FAIL)
465        FAIL6   OUT( SYNERR  ≠≠ ≠::=≠ EXPECTED ≠≠ )  :(FAIL)
466        FAIL7   OUT( SYNERR  ≠ PERIOD NOT EXPECTED≠ ) :(FAIL)
                   *
467        FAIL    OUTPUT = SYNERR
468                ASSEM( ≠≠, ≠ERR≠, ≠≠, SYNERR )
                   *
                   *
                   *
                   *
                   *
469        END
```

SUCCESSFUL COMPILATION

CMMDE FL 4000R

```
*
BLANK              00B       GETTKN    .
IDENT              1002B     GETTKN    .
CR                 155B      SUBGRAM   S.PARAM   .
PARAM              5001B     SUBGRAM   S.PARAM   .
*
```

```
 NEWV      NEWV      GETKYWD  .
 KILLV     KILLV     GETKYWD  .
 MDATA     MDATA     GETKYWD  .
 MCAP      MCAP      GETKYWD  .
 VIEW      VIEW      GETKYWD  .
 NEWDF     NEWDF     GETKYWD  .
 NEWDR     NEWDR     GETKYWD  .
 KILLDF    KILLDF    GETKYWD  .
 KILLDR    KILLDR    GETKYWD  .
 NEWBLK    NEWBLK    GETKYWD  .
 KILLBLK   KILLBLK   GETKYWD  .
 COPYBDFILE  COPYBDFILE  GETKYWD
 KILLOBJ   KILLOBJ   GETKYWD  .
 DELOWN    DELOWN    GETKYWD  .
 DELLINK   DELLINK   GETKYWD  .
 NEWKEY    NEWKEY    GETKYWD  .
 ADDKEY    ADDKEY    GETKYWD  .
 DELKEY    DELKEY    GETKYWD  .
*
 .
*
*
 SENTENCE            .
 LINE                .
*
 .
*
 IDENTITY            F.NOACT  .
 DSPCAP              F.DCAP   .
 PRNTWRD             F.PWRD   .
 PRNTWRDS            F.PWRDS  .
 NEWVAR              F.NEWV   .
 KILLVAR             F.KILLV  .
 MOVEDTM             F.MVD    .
 MOVECAP             F.MVC    .
 VIEW.STACK          F.VIEW   .
 NEWDSKFILE          F.NEWDF  .
 NEWDIR              F.NEWDR  .
 KILLDSKFILE         F.KLLDF  .
 KILLDIR             F.KLLDR  .
 NEWBLK              F.NEWB   .
 KILLBLK             F.KILLB  .
 COPYBDFILE          F.CPYBF  .
 KILLOBJ             F.KLOBJ  .
 DELOWN              F.DLOWN  .
 DELLINK             F.DLNK   .
 NEWKEY              F.NEWKY  .
 ADDKEY              F.ADDKY  .
 DELKEY              F.DELKY  .
*
 .
*
*
 IDENTITY : LINE ::= SENTENCE CR .
*
*
 DSPCAP     : SENTENCE ::= PCAP  BLANK PARAM              .
 PRNTWRD    : SENTENCE ::= PDATA BLANK PARAM              .
 PRNTWRDS   : SENTENCE ::= PDATA BLANK PARAM PARAM        .
 NEWVAR     : SENTENCE ::= NEWV  BLANK PARAM              .
 KILLVAR    : SENTENCE ::= KILLV BLANK PARAM              .
 MOVEDTM    : SENTENCE ::= MDATA BLANK PARAM PARAM        .
 MOVECAP    : SENTENCE ::= MCAP  BLANK PARAM PARAM        .
 VIEW.STACK : SENTENCE ::= VIEW  BLANK PARAM              .
```

```
NEWDIR      : SENTENCE  ::=  NEWDR  BLANK  PARAM  PARAM  PARAM      .
KILLDSKFILE : SENTENCE  ::=  KILLDF  BLANK  PARAM                  .
KILLDIR     : SENTENCE  ::=  KILLDR  BLANK  PARAM                  .
NEWBLK      : SENTENCE  ::=  NEWBLK  BLANK  PARAM                  .
KILLBLK     : SENTENCE  ::=  KILLBLK  BLANK  PARAM                 .
COPYBDFILE  : SENTENCE  ::=  COPYBDFILE  BLANK  IDENT  BLANK  IDENT
                                          BLANK  PARAM .
KILLOBJ     : SENTENCE  ::=  KILLOBJ  BLANK  PARAM                     .
DELOWN      : SENTENCE  ::=  DELOWN  BLANK  PARAM                      .
DELLINK     : SENTENCE  ::=  DELLINK  BLANK  PARAM                     .
NEWKEY      : SENTENCE  ::=  NEWKEY  BLANK PARAM                       .
ADDKEY      : SENTENCE  ::=  ADDKEY  BLANK PARAM  PARAM  PARAM         .
DELKEY      : SENTENCE  ::=  DELKEY  BLANK PARAM  PARAM                .
*
.
```

TOKEN DEFINITIONS

```
BLANK                   002
IDENT                   1002B
CR                      155B
PARAM                   5001B
PCAP                    4000B+1
PDATA                   4000B+2
NEWV                    4000B+3
KILLV                   4000B+4
MDATA                   4000B+5
MCAP                    4000B+6
VIEW                    4000B+7
NEWDF                   4000B+8
NEWDR                   4000B+9
KILLDF                  4000B+10
KILLDR                  4000B+11
NEWBLK                  4000B+12
KILLBLK                 4000B+13
COPYBDFILE              4000B+14
KILLOBJ                 4000B+15
DELOWN                  4000B+16
DELLINK                 4000B+17
NEWKEY                  4000B+18
ADDKEY                  4000B+19
DELKEY                  4000B+20
SENTENCE                2000B+1
LINE                    2000B+2
```

RULE INFO

```
1     LINE   IDENTITY    2
2     SENTENCE   DSPCAP     3
3     SENTENCE   PRNTWRD     3
4     SENTENCE   PRNTWRDS     4
5     SENTENCE   NEWVAR     3
6     SENTENCE   KILLVAR     3
7     SENTENCE   MOVEDTN     4
8     SENTENCE   MOVECAP     4
9     SENTENCE   VIEW.STACK     3
10      SENTENCE    NEWDSKFILE     3
11      SENTENCE    NEWDIR     5
12      SENTENCE    KILLDSKFILE     3
13      SENTENCE    KILLDIR     3
14      SENTENCE    NEWBLK     3
15      SENTENCE    KILLBLK     3
16      SENTENCE    COPYBDFILE     7
```

```
18      SENTENCE    DELDOWN     3
19      SENTENCE    DELLINK     3
20      SENTENCE    NEWKEY      3
21      SENTENCE    ADDKEY      5
22      SENTENCE    DELKEY      4
```

```
*
 LINE      S.LINE    .
*
  .
```