PARSER GENERATOR - BNF INPUT

```
||TABLE1
  1  id
  2  num
  3  lnum
  4  string
  5  lstring
  6  char
  7  ,
  8  ;
  9  :
 10  ..
 11  =>
 12  ←
 13  =
 14  #
 15  <
 16  >
 17  <=
 18  >=
 19  ~
 20  +
 21  -
 22  *
 23  /
 24  ↑
 25  .
 26  @
 27  !
 28  INTEGER
 29  CARDINAL
 30  CHARACTER
 31  BOOLEAN
 32  STRING
 33  RECORD
 34  POINTER
 35  ARRAY
 36  DESCRIPTOR
 37  PROCEDURE
 38  PORT
 39  SIGNAL
 40  ERROR
 41  PROCESS
 42  PROGRAM
 43  MONITOR
 44  RELATIVE
 45  LONG
 46  TYPE
 47  FRAME
 48  TO
 49  ORDERED
 50  BASE
 51  OF
 52  PACKED
 53  RETURNS
 54  MONITORED
 55  OVERLAID
 56  COMPUTED
 57  MACHINE
 58  DEPENDENT
 59  DIRECTORY
 60  DEFINITIONS
 61  IMPORTS
 62  EXPORTS
 63  SHARES
 64  LOCKS
 65  USING
 66  PUBLIC
 67  PRIVATE
 68  ENTRY
 69  INTERNAL
 70  CODE
 71  ABS
 72  AND
 73  MAX
 74  MIN
 75  MOD
 76  NOT
 77  OR
 78  LENGTH
 79  NEW
 80  START
 81  FORK
 82  JOIN
 83  LOOPHOLE
 84  SIZE
 85  FIRST
 86  LAST
 87  MEMORY
 88  REGISTER
 89  NULL
 90  IF
 91  THEN
 92  ELSE
 93  WITH
 94  FROM
 95  FOR
 96  INCREASING
 97  DECREASING
 98  IN
 99  THROUGH
100  UNTIL
101  WHILE
102  REPEAT
103  FINISHED
104  RETURN
105  EXIT
106  LOOP
107  GOTO
108  GO
109  WAIT
110  RESTART
111  NOTIFY
112  BROADCAST
113  STOP
114  RESUME
115  CONTINUE
116  RETRY
117  TRANSFER
118  STATE
119  OPEN
120  ENABLE
121  ANY
122  EXITS
123  )
124  ]
125  }
126  END
127  ENDLOOP
128  ENDCASE
129  (
130  [
131  {
132  BEGIN
133  DO
134  SELECT
135  EOF

||TABLE2
137  goal
138  unit
139  directory
140  includelist
141  includeitem
142  definitions
143  module
144  classhead
145  defhead
146  defbody
147  locks
```

```
148  interface
149  imports
150  exports
151  modulelist
152  moduleitem
153  shares
154  declist
155  declaration
156  attributes
157  entry
158  idlist
159  idlist'
160  identlist
161  identlist'
162  typeexp
163  typeid
164  typecons
165  monitored
166  dependent
167  reclist
168  pairlist
169  pairitem
170  typelist
171  variantpair
172  variantpart
173  vcasehead
174  tagtype
175  variantlist
176  variantitem
177  subreclist
178  ordered
179  base
180  pointertype
181  pointerprefix
182  array
183  indextype
184  transfermode
185  arguments
186  arglist
187  returnlist
188  fieldlist
189  initialization
190  initvalue
191  codelist
192  procaccess
193  statement
194  block
195  blockhead
196  begin
197  bindlist
198  binditem
199  exits
200  elsepart
201  casehead
202  casestmtlist
203  casestmtitem
204  caselabel
205  casetest
206  otherpart
207  forclause
208  direction
209  dotest
210  do
211  doexit
212  exitlist
213  exititem
214  enables
215  catchhead
216  catchlist
217  catchitem
218  catchcase
219  lhslist
220  statementlist
221  statementlist'
222  transfer
223  optargs
```

```
224  explist
225  orderlist
226  keylist
227  keyitem
228  optexp
229  exp
230  transferop
231  caseexplist
232  caseexpitem
233  disjunct
234  conjunct
235  negation
236  not
237  relation
238  optrelation
239  relop
240  relationtail
241  range
242  interval
243  bounds
244  sum
245  addop
246  product
247  multop
248  factor
249  primary
250  desclist
251  prefixop
252  typeop
253  lhs
254  qualifier
255  memory

||TABLE3

  0   0 * * *          ::= goal EOF

  1   0 goal           ::= . unit .
  2   0                  | . unit ..

  3  10 unit           ::= directory definitions module

  4   4 directory      ::=
  5   9                  | DIRECTORY includelist ;

  6   6 includelist    ::= includeitem
  7   7                  | includelist , includeitem

  8  11 includeitem    ::= id : FROM string
  9 222                  | id : FROM string USING [ idlist ]

 10   4 definitions    ::=
 11   0                  | DEFINITIONS FROM idlist ;

 12  12 module         ::= id : classhead = attributes block
 13  12                  | id : defhead = attributes defbody

 14  13 classhead      ::= PROGRAM arguments interface
 15 201                  | MONITOR arguments locks interface

 16  14 defhead        ::= DEFINITIONS shares

 17  21 defbody        ::= begin declist END

 18 202 locks          ::=
 19 203                  | LOCKS primary
 20 204                  | LOCKS primary USING id : typeexp

 21   0 interface      ::= imports exports shares

 22   4 imports        ::=
 23   9                  | IMPORTS modulelist

 24   4 exports        ::=
 25   0                  | EXPORTS idlist
```

```
26   6 modulelist      ::= moduleitem
27   7                   | modulelist , moduleitem

28  15 moduleitem      ::= id
29  16                   | id : id

30   4 shares          ::=
31   0                   | SHARES idlist

32   5 declist         ::=
33   7                   | declist declaration ;

34  22 declaration     ::= identlist attributes entry typeexp initialization
35  23                   | identlist attributes TYPE = attributes typeexp

36  24 attributes      ::=
37  25                   | PUBLIC
38  26                   | PRIVATE

39 223 entry           ::=
40 224                   | ENTRY
41 225                   | INTERNAL

42   8 idlist          ::= idlist'

43  27 idlist'         ::= id
44  28                   | id , idlist'

45   8 identlist       ::= identlist'

46  27 identlist'      ::= id :
47  28                   | id , identlist'

48   1 typeexp         ::= id
49   0                   | typeid
50   0                   | typecons

51  29 typeid          ::= INTEGER
52  30                   | CARDINAL
53  31                   | CHARACTER
54  32                   | BOOLEAN
55  33                   | STRING
56  34                   | id . id
57  35                   | id id
58  36                   | id typeid

59  37 typecons        ::= interval
60  38                   | id interval
61  39                   | typeid interval
62  40                   | { idlist }
63  41                   | monitored dependent RECORD reclist
64  42                   | ordered base pointertype
65  43                   | array indextype OF typeexp
66  44                   | DESCRIPTOR FOR typeexp
67  45                   | transfermode arguments
68 212                   | id RELATIVE typeexp
69 213                   | typeid RELATIVE typeexp
70  46                   | LONG typeexp
71  47                   | FRAME [ id ]

72  85 monitored       ::=
73 205                   | MONITORED

74  48 dependent       ::=
75  49                   | MACHINE DEPENDENT

76  50 reclist         ::= [ pairlist ]
77  50                   | [ typelist ]
78  51                   | [ pairlist , variantpair ]
79  52                   | [ variantpair ]
80  53                   | [ variantpart ]

81   6 pairlist        ::= pairitem
82   7                   | pairlist , pairitem

83  54 pairitem        ::= identlist attributes typeexp
```

```
 84  55 typelist        ::= typecons
 85  55                   | typeid
 86  56                   | id
 87  57                   | typecons , typelist
 88  57                   | typeid , typelist
 89  58                   | id , typelist

 90  54 variantpair     ::= identlist attributes variantpart

 91  59 variantpart     ::= SELECT vcasehead FROM variantlist ENDCASE

 92  60 vcasehead       ::= id : attributes tagtype
 93  61                   | COMPUTED tagtype
 94  62                   | OVERLAID tagtype

 95  63 tagtype         ::= *
 96   0                   | typeexp

 97   6 variantlist     ::= variantitem ,
 98   7                   | variantlist variantitem ,

 99  64 variantitem     ::= idlist => subreclist

100   0 subreclist      ::= reclist
101  65                   | NULL

102  85 ordered         ::=
103  67                   | ORDERED

104  85 base            ::=
105  67                   | BASE

106  68 pointertype     ::= pointerprefix
107   0                   | pointerprefix TO typeexp

108   3 pointerprefix   ::= POINTER
109   0                   | POINTER interval

110  66 array           ::= ARRAY
111  67                   | PACKED ARRAY

112   4 indextype       ::=
113   0                   | typeexp

114  69 transfermode    ::= PROCEDURE
115  70                   | PORT
116  71                   | SIGNAL
117  72                   | ERROR
118  73                   | PROCESS
119  74                   | PROGRAM

120   0 arguments       ::= arglist returnlist

121   4 arglist         ::=
122   0                   | fieldlist

123   4 returnlist      ::=
124   0                   | RETURNS fieldlist

125   9 fieldlist       ::= [ pairlist ]
126   9                   | [ typelist ]

127  75 initialization  ::=
128  66                   | ← initvalue
129  67                   | = initvalue

130   0 initvalue       ::= exp
131  76                   | procaccess block
132  77                   | CODE
133  78                   | MACHINE CODE BEGIN codelist END

134 214 codelist        ::= orderlist
135 215                   | codelist ; orderlist

136  79 procaccess      ::=
137  80 statement      ::= lhs
138  81                  | lhs ← exp
139  82                  | [ explist ] ← exp
140  83                  | block
141  84                  | IF exp THEN statement elsepart
142  86                  | casehead casestmtlist ENDCASE otherpart
143  87                  | forclause dotest do enables statementlist doexit ENDLOOP
144  90                  | EXIT
145 216                  | LOOP
146  91                  | GOTO id
147  92                  | GO TO id
148  93                  | RETURN optargs
149  94                  | transfer lhs
150 207                  | WAIT lhs
151  95                  | ERROR
152  96                  | STOP
153  97                  | STOP [ ! catchlist ]
154  98                  | NULL
155  99                  | RESUME optargs
156 100                  | CONTINUE
157 101                  | RETRY
158 102                  | lhs ← STATE

159   0 block          ::= blockhead END
160  89                  | blockhead exits END

161  17 blockhead      ::= begin enables declist statementlist

162   3 begin          ::= BEGIN
163  18                  | BEGIN OPEN bindlist ;

164   6 bindlist       ::= binditem
165   7                  | bindlist , binditem

166  19 binditem       ::= exp
167  20                  | id : exp

168   9 exits          ::= EXITS exitlist
169   9                  | EXITS exitlist ;

170   4 elsepart       ::=
171   0                  | ELSE statement

172  66 casehead       ::= SELECT exp FROM
173  67                  | WITH binditem SELECT optexp FROM

174   6 casestmtlist   ::= casestmtitem ;
175   7                  | casestmtlist casestmtitem ;

176 105 casestmtitem   ::= caselabel => statement

177   6 caselabel      ::= casetest
178   7                  | caselabel , casetest

179 106 casetest       ::= optrelation
180 107                  | exp

181   4 otherpart      ::=
182   0                  | => statement

183   4 forclause      ::=
184 108                  | FOR id ← exp , exp
185 109                  | FOR id direction IN range
186 110                  | THROUGH range

187 111 direction      ::=
188 111                  | INCREASING
189 112                  | DECREASING

190   4 dotest         ::=
191   0                  | WHILE exp
192 113                  | UNTIL exp

193   3 do             ::= DO
194  18                  | DO OPEN bindlist ;
```

```
195 114 doexit         ::=
196 115                  | REPEAT exitlist
197 115                  | REPEAT exitlist ;
198 116                  | REPEAT exitlist ; FINISHED => statement
199 116                  | REPEAT exitlist ; FINISHED => statement ;
200 117                  | REPEAT FINISHED => statement
201 117                  | REPEAT FINISHED => statement ;

202   6 exitlist       ::= exititem
203   7                  | exitlist ; exititem

204 118 exititem       ::= idlist => statement

205  85 enables        ::=
206 119                  | ENABLE catchitem ;
207 120                  | ENABLE BEGIN catchlist END ;
208 121                  | ENABLE BEGIN catchhead END ;

209   6 catchhead      ::= catchcase ;
210   7                  | catchhead catchcase ;

211   0 catchlist      ::= catchitem
212 122                  | catchhead catchitem

213 123 catchitem      ::= catchcase
214   5                  | ANY => statement

215 105 catchcase      ::= lhslist => statement

216   6 lhslist        ::= lhs
217   7                  | lhslist , lhs

218   4 statementlist  ::=
219   0                  | statement
220   8                  | statementlist'
221 124                  | statementlist' statement

222   6 statementlist' ::= statement ;
223   7                  | statementlist' statement ;

224 125 transfer'      ::= SIGNAL
225 126                  | ERROR
226 218                  | RETURN WITH ERROR
227 127                  | START
228 128                  | RESTART
229 208                  | JOIN
230 209                  | NOTIFY
231 210                  | BROADCAST
232 129                  | TRANSFER WITH
233 130                  | RETURN WITH

234   4 optargs        ::=
235   0                  | [ explist ]

236   8 explist        ::= orderlist
237   8                  | keylist

238   6 orderlist      ::= optexp
239   7                  | orderlist , optexp

240   6 keylist        ::= keyitem
241   7                  | keylist , keyitem

242 140 keyitem        ::= id : optexp

243   4 optexp         ::=
244   0                  | exp

245 141 exp            ::= transferop lhs
246 143                  | IF exp THEN exp ELSE exp
247 144                  | casehead caseexplist ENDCASE => exp
248 145                  | lhs ← exp
249   0                  | disjunct

250 125 transferop     ::= SIGNAL
```

```
251 126                    | ERROR
252 146                    | NEW
253 127                    | START
254 211                    | FORK
255 208                    | JOIN

256   6 caseexplist    ::= caseexpitem ,
257   7                    | caseexplist caseexpitem ,

258 105 caseexpitem    ::= caselabel => exp

259C  0 disjunct       ::= conjunct
260 147                    | disjunct OR conjunct

261C  0 conjunct       ::= negation
262 148                    | conjunct AND negation

263C  0 negation       ::= relation
264 149                    | not relation

265   0 not            ::= ~
266   0                    | NOT

267C  0 relation       ::= sum
268 150                    | sum optrelation

269   0 optrelation    ::= relationtail
270 151                    | not relationtail

271 152 relop          ::= =
272 153                    | #
273 154                    | <
274 155                    | <=
275 156                    | >
276 157                    | >=

277   0 relationtail   ::= relop sum
278 158                    | IN range

279   0 range          ::= interval
280   1                    | id
281   0                    | typeid
282  38                    | id interval
283  39                    | typeid interval

284 159 interval       ::= [ bounds ]
285 160                    | [ bounds )
286 161                    | ( bounds ]
287 162                    | ( bounds )

288   0 bounds         ::= exp .. exp

289C  0 sum            ::= product
290 142                    | sum addop product

291 163 addop          ::= +
292 164                    | -

293C  0 product        ::= factor
294 142                    | product multop factor

295 165 multop         ::= *
296 166                    | /
297 167                    | MOD

298C  0 factor         ::= primary
299 168                    | - primary

300C  0 primary        ::= lhs
301   2                    | num
302 226                    | lnum
303 169                    | char
304 170                    | string
305 219                    | lstring
306 171                    | [ explist ]
307 172                    | prefixop [ orderlist ]
```

```
308 220                    | INTEGER [ explist ]
309 221                    | CARDINAL [ explist ]
310 141                    | typeop [ typeexp ]
311 173                    | @ lhs
312 174                    | DESCRIPTOR [ desclist ]

313   0 desclist         ::= exp
314 175                    | exp , exp
315 176                    | exp , exp , typeexp

316 177 prefixop         ::= LONG
317 178                    | ABS
318 179                    | MIN
319 180                    | MAX
320 181                    | BASE
321 182                    | LENGTH

322 183 typeop           ::= SIZE
323 184                    | FIRST
324 185                    | LAST

325   1 lhs              ::= id
326   0                    | ( exp )
327   0                    | lhs qualifier
328 186                    | LOOPHOLE [ exp ]
329 187                    | LOOPHOLE [ exp , typeexp ]
330 188                    | memory [ exp ]

331 189 qualifier        ::= [ explist ]
332 190                    | [ explist ! catchlist ]
333 191                    | . id
334 192                    | ↑

335 193 memory           ::= MEMORY
336 194                    | REGISTER

==== LALR(1) TABLE STATISTICS ====

NUMBER OF STATES =  347
COUNTS: TSCAN1280, TSCANREDUCE2189, TREDUCE1402, NSCAN1204, NSCANREDUCE 287
STATES WITH NONTERMINAL ENTRIES = 210
OTHER STATES ACCESSED VIA TERMINAL SYMBOLS =  51
TERMINAL ENTRIES =  1420,   NONTERMINAL ENTRIES =   255
```