PARSER GENERATOR - BNF INPUT

```
||TABLE1
  1  id
  2  num
  3  sr
  4  sr1
  5  ,
  6  ;
  7  ..
  8  ←
  9  +
 10  -
 11  *
 12  /
 13  ↑
 14  .
 15  @
 16  !
 17  $
 18  %
 19  INTEGER
 20  CARDINAL
 21  CHARACTER
 22  BOOLEAN
 23  STRING
 24  UNSPECIFIED
 25  MOD
 26  DESCRIPTOR
 27  BASE
 28  LENGTH
 29  SIZE
 30  MEMORY
 31  )
 32  ]
 33  (
 34  [
 35  EOF

||TABLE2
 37  goal
 38  typespec
 39  typeid
 40  typeconstruct
 41  stmtlist
 42  stmt
 43  exp
 44  sum
 45  addop
 46  product
 47  multop
 48  factor
 49  primary
 50  builtincall
 51  typeop
 52  lhs
 53  qualifier
 54  interval
 55  explist

||TABLE3

  0   0 * * *          ::= goal EOF

  1   0 goal           ::= stmtlist

  2   1 stmtlist       ::= stmt
  3   2                  | stmtlist ; stmt

  4   3 stmt           ::= exp
  5   4                  | lhs ← exp

  6   5 exp            ::= sum
```

```
 7  20 sum             ::= product
 8  21                   | sum addop product

 9  22 addop           ::= +
10  23                   | -

11  24 product         ::= factor
12  25                   | product multop factor

13  26 multop          ::= *
14  27                   | /
15  28                   | MOD

16  30 factor          ::= primary
17  31                   | - primary

18  40 primary         ::= lhs
19  41                   | ( exp )
20  43                   | builtincall
21  44                   | @ lhs

22  50 builtincall     ::= LENGTH [ lhs ]
23  51                   | BASE [ lhs ]
24  52                   | DESCRIPTOR [ exp ]
25  53                   | DESCRIPTOR [ exp , exp ]
26  54                   | typeop [ typespec ]

27  65 typeop          ::= SIZE

28  72 lhs             ::= id
29  73                   | num
30  74                   | sr1
31  75                   | sr
32  76                   | ( exp ) qualifier
33  77                   | lhs qualifier
34  80                   | MEMORY [ interval ]
35  81                   | MEMORY [ exp ]
36  83                   | id $ id
37  84                   | num $ id

38  90 qualifier       ::= . id
39  91                   | ↑
40  92                   | %
41  93                   | % typespec
42  94                   | [ explist ]
43  95                   | [ interval ]

44 105 typespec        ::= typeid
45 107                   | typeconstruct

46 110 typeid          ::= INTEGER
47 111                   | CARDINAL
48 112                   | CHARACTER
49 113                   | BOOLEAN
50 114                   | STRING
51 115                   | UNSPECIFIED
52 116                   | id $ id
53 119                   | id
54 120                   | id typeid

55 125 typeconstruct   ::= @ typespec

56 140 explist         ::= exp
57 141                   | explist , exp
58 142                   |

59 143 interval        ::= exp .. exp
60 144                   | exp ! exp
```

- UNUSED(U) OR UNDEFINED(D) SYMBOLS (REFER TO TABLES 1 & 2)
35U 37U

- CHAIN(C) OR EMPTY(E) PRODUCTIONS (REFER TO TABLE 3)

```
  58E


==== SLR(1) TABLE STATISTICS ====

NUMBER OF STATES =   54
COUNTS: TSCAN 192, TSCANREDUCE 103, TREDUCE 146, NSCAN  73, NSCANREDUCE  68
STATES WITH NONTERMINAL ENTRIES =  32
OTHER STATES ACCESSED VIA TERMINAL SYMBOLS =  10



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

NUMBER OF STATES =   54
COUNTS: TSCAN 192, TSCANREDUCE 103, TREDUCE 141, NSCAN  73, NSCANREDUCE  68
STATES WITH NONTERMINAL ENTRIES =  32
OTHER STATES ACCESSED VIA TERMINAL SYMBOLS =  10


PARSE TABLE DATA


SCANNER HASH TABLE PROBES, TERMINAL SYMBOL->PROBE COUNT, HASH CODE
  1-> 1  27    2-> 1  35    3-> 1  39    4-> 2  27    7-> 1   6   19-> 1  26   20-> 2  53
 21-> 2   6   22-> 1  34   23-> 1  13   24-> 2  52   25-> 1  43   26-> 5  27   27-> 1  25
 28-> 2  26   29-> 1  11   30-> 2  11

TERMINAL ENTRIES =   145,   NONTERMINAL ENTRIES =    24

-- parse tables
  hashval: CARDINAL =  53;

  endmarker: Symbol = 35;
  lastntstate: State =  23;

  LALRTable: PUBLIC TYPE = PRIVATE MACHINE DEPENDENT RECORD [
    scantable: RECORD[
      hashtab: ARRAY [0.. 53] OF VocabHashEntry,
      scantab: ARRAY CHARACTER [40C..177C] OF Symbol,
      vocabbody: RECORD [  -- a string body
        length, maxlength: CARDINAL,
        text: ARRAY [0.. 57) OF UNSPECIFIED],
      vocabindex: ARRAY [0.. 35] OF CARDINAL],
    parsetable: RECORD[
      proddata: ARRAY [0.. 60] OF ProductionInfo,
      nstate: ARRAY [0.. 23] OF CARDINAL,
      nlen: ARRAY [0.. 23] OF CARDINAL,
      nsym: ARRAY [0.. 23] OF Symbol,
      nact: ARRAY [0.. 23] OF ActionEntry,
      ntdefaults: ARRAY [0.. 20] OF ActionEntry,
      asstl: ARRAY [0.. 54] OF AsstlEntry,
      tstate: ARRAY [0.. 54] OF CARDINAL,
      tsym: ARRAY [0..144] OF Symbol,
      tact: ARRAY [0..144] OF ActionEntry]];
```