**********
* PROGRAMMING LANGUAGE GRAMMAR ANTHOLOGY *
**********

Name of Project: Programming Language Grammar Anthology

Organization: Computer Systems Research Group
University of Toronto

Principal Investigator: Prof. D.B. Wortman

Address: Computer Systems Research Group
University of Toronto
Toronto, Ontario, CANADA M5S 1A4

Description of Project:

Objectives: To build a machine and human readable data base of formal grammars describing programming languages. The data base will provide a standard set of test grammars in a common form for evaluating the capabilities of new and existing parsing algorithms.

Starting date: February, 1972

Completion date: Open

Current status: A grammar for describing grammars has been developed (Figure 1). The data base presently contains grammars for about 30 languages including FORTRAN, COBOL, PL/I, PASCAL, ALGOL-W, ALGOL 60, ALGOL 68.

Methodology: Grammars are being accumulated via literature search and by CONTRIBUTIONS FROM INTERESTED PARTIES.

Potential Documentation: A technical report containing a listing of all the grammars in the data-base is planned. A copy will be distributed automatically to all contributors, further distribution is uncertain at this time.

Our hope is to be able to make the data base available in machine-readable form for a nominal fee.

Name of Contact for Future Reference: Principal Investigator

Source of Funding: Computer Systems Research Group
National Research Council of Canada



```
"a grammar for describing grammars"


grammar = production_list '.' ;

production_list = production_set ,
                  production_list ';' production_set ;

production_set = filler non_terminal '=' alternative_list ;

filler =        ,
         filler ' ' ,
         filler "'" comment_text "'" ;

comment_text =
               comment_text comment_character ;

comment_character = id_character ,
                    meta_character ,
                    quote ;

id-character = letter ,
               digit ,
               special_character ;

letter = 'a' , 'b' , 'c' , 'd' , 'e' , 'f' , 'g' ,
         'h' , 'i' , 'j' , 'k' , 'l' , 'm' , 'n' ,
         'o' , 'p' , 'q' , 'r' , 's' , 't' , 'u' ,
         'v' , 'w' , 'x' , 'y' , 'z' ;

digit = '0' , '1' , '2' , '3' , '4' , '5' , '6' , '7' , '8' , '9' ;

special_character = '#' , '|' , '$' , '-' , '@' , '%' ,
                    '*' , '<' , '/' , '+' , '_' , ')' ,
                    '!' , '&' , '>' , ':' , '¬' , '?' ,
                    '¢' , '(' ;

meta_character = '=' , ',' , '.' , ';' , ' ' ;

quote = "'" ;

non-terminal = identifier ' ' filler ;

identifier = id_character ,
             identifier id_character ;

alternative_list = token_string ,
                   alternative_list ',' token_string ;

token_string = filler ,
               token_string  token ;

token = non-terminal ,
        quote terminal quote ' ' filler ;

terminal = terminal_character ,
           terminal terminal_character ;

terminal_character = id_character ,
                     meta_character ,
                     quote quote ,
                     "'"
```

Figure 1. A Grammar-Grammar