

## 1 拓展的PL/0文法

鉴于中文的文法在编程时的对应关系容易混乱，我就先将所给的文法翻译成了英文。具体文法如下：

```
program    ::=    block .

block      ::=    [ constdec ][ vardec ]{ [ procdec ][ fundec ] } compstmt

constdec   ::=    const constdef {, constdef };

constdef   ::=    ident = const

const      ::=    [+|-] unsign | character

character  ::=    ' letter ' | ' digit '

string     ::=    “{ASCII characters with decimal code number varys from 32 to 126 exclude 34}”

unsign     ::=    digit { digit }

ident      ::=    letter { letter | digit }

vardec     ::=    var vardef ; { vardef ; }

vardef     ::=    ident {, ident } : type

type       ::=    basictype | array ' [ unsign ] ' of basictype

basictype  ::=    integer | char

procdec    ::=    prothead block {; prothead block };

fundec     ::=    funhead block {; funhead block };

prothead   ::=    procedure ident ' ( [ paralist ] ) ' ;

funhead    ::=    function ident ' ( [ paralist ] ) ' : basictype ;

paralist   ::=    [ var ] ident {, ident } : basictype {; paralist }

statement  ::=    assignstmt | ifstmt | repeatstmt | pcallstmt
                | compstmt | readstmt | writestmt | forstmt | nullstmt

assignstmt ::=    ident := expression | funident := expression
                | ident ' [ expression ] ' := expression

funident   ::=    ident

expression ::=    [+|-] term { addop term }

term       ::=    factor { multop factor }

factor     ::=    ident | ident ' [ expression ] ' | unsign ' ( ' expression ' ) ' | fcallstmt

fcallstmt  ::=    ident ' ( [ arglist ] ) '

arglist    ::=    argument {, argument }

argument   ::=    expression

addop      ::=    + | -

multop     ::=    * | /
```

$condition ::= expression \text{ relop } expression$   
 $relop ::= < \mid <= \mid > \mid >= \mid = \mid <>$   
 $ifstmt ::= \text{if } condition \text{ then } statement$   
 $\quad \mid \text{if } condition \text{ then } statement \text{ else } statement$   
 $repeatstmt ::= \text{repeat } statement \text{ until } condition$   
 $forstmt ::= \text{for } ident := expression \text{ (to|downto) } expression \text{ do } statement$   
 $pcallstmt ::= ident '([ arglist ] )'$   
 $compstmt ::= \text{begin } statement \{ ; statement \} \text{end}$   
 $readstmt ::= \text{read}(' ident \{, ident \}')$   
 $writestmt ::= \text{write}(' string , expression ') \mid \text{write}(' string ') \mid \text{write}(' expression ')'$   
 $letter ::= a \mid b \mid c \mid \dots \mid z \mid A \mid B \mid C \mid \dots \mid Z$   
 $digit ::= 0 \mid 1 \mid 2 \mid 3 \mid \dots \mid 9$