

## PL0 语言的 EBNF 范式

### EBNF 范式的符号说明

<>: 表示语法构造成分，为非终结符

::= : 该符号的左部由右部定义，读作“定义为”

| : 表示或

{ } : 表示括号内的语法成分可重复

[ ] : 括号内成分为任选项

( ) : 圆括号内成分优先

与原来版本比做出改变的标为红色

<程序> ::= <分程序>.

<分程序> ::= [<常量说明部分>][<变量说明部分>][<过程说明部分>]<语句>

<常量说明部分> ::= CONST<常量定义>{,<常量定义>;}

<常量定义> ::= <标识符>=<无符号整数>|<浮点数>|<字符>|<字符串>

<无符号整数> ::= <数字>{<数字>}

<变量说明部分> ::= VAR<标识符>[ (:<类别> [ ARRAY \[<无符号整数>..<无符号整数>{,<无符号整数>..<无符号整数>}] OF <类别> ] ) | ((,<标识符>):<类别> ) { (:<类别> [ ARRAY \[<无符号整数>..<无符号整数>{,<无符号整数>..<无符号整数>}] of <类别> ] ) | ((,<标识符>):<类别> ) };

<类别>::=INT

<标识符> ::= <字母>{<字母>|<数字>}

<过程说明部分> ::= <过程首部><分程序>{,<过程说明部分>;}

<过程首部> ::= PROCEDURE<标识符>;

<语句> ::= <赋值语句>|<复合语句>|<条件语句>|<当型循环语句>|<过程调用语句>|<读语句>|<写语句>|<空>|<FOR 循环语句>|<REPEAT 循环语句>

<赋值语句> ::= <标识符>:=<表达式>

<复合语句> ::= BEGIN<语句>{,<语句>}END

<条件> ::= <表达式><关系运算符><表达式>|ODD<表达式>

<条件语句> ::= IF<条件>THEN<语句>

<表达式> ::= [+|-]<项>{<加法运算符><项>}

<项> ::= <因子>{<乘法运算符><因子>}

<因子> ::= <标识符>|<无符号整数>|('(<表达式>'))

<加法运算符> ::= +|-

<乘法运算符> ::= \*/

<关系运算符> ::= =|<|<=|>|>=

<当型循环语句> ::= WHILE<条件>DO<语句>

<过程调用语句> ::= CALL<标识符>

<读语句> ::= READ('(<标识符>{,<标识符>}')

<写语句> ::= WRITE('(<表达式>{,<表达式>}')

<字母> ::= a|b|...|X|Y|Z

<数字> ::= 0|1|...|8|9

## 扩展语法的 EBNF 范式

<FOR 循环语句>::= FOR<>:=<表达式>TO<表达式>DO<语句>

<REPEAT 循环语句>::= REPEAT<语句>UNTIL<条件>

<浮点数>::=[-]{数字}(.{数字})[ [Ee] [-+]{数字} ]

<字符>::='.' (想表达 “中任何字符都可以，借用正则表达式里的通配符来表示)

<字符串>::="">{<字符>}"