PL/0 语言的文法

BNF 是描述语言文法的基本形式，除此之外，还可以使用其他的一些表示方法，本节使用扩充的 BNF（简称 EBNF）和语法图来描述 PL/0 语言的文法。

1．PL/0 语言文法的 EBNF 表示

（1）EBNF 简介

EBNF 是对 BNF 的扩充，在 BNF 中的元符号有：“::=”（或“→”）、“|”、“〈 〉”，EBNF 在此基础上又增加了三种元符号，约定如下。

① 花括号{ }

表示其内语法成分可以重复，其中，

{α}：表示符号串 α 可重复 0 到任意次数，即 α^*。

$\{\alpha\}_m^n$：表示符号串 α 可重复 m 到 n 次。

例如：A→Aα|β，可改写为：A→{α}β。

② 方括号[ ]

表示方括号内的成分为任选项。

[α]：表示符号串 α 为任选项。

例如：A→αβ|α，可改写为：A→α[β]。

③ 圆括号( )

(α)：圆括号内的符号串 α 优先，利用圆括号可在产生式右部中提取公因子。

例如：A→$\alpha\beta_1|\alpha\beta_2|\cdots|\alpha\beta_n$，可改写为：A →$\alpha(\beta_1|\beta_2|\cdots|\beta_n)$

例如，用 EBNF 描述〈整数〉文法的定义：

〈整数〉∷=[+|-]〈数字〉{〈数字〉}

〈数字〉∷=0|1|2|3|4|5|6|7|8|9

可以看出，EBNF 表示法要比 BNF 表示法清晰、简单得多。

（2）PL/0 语言文法的 EBNF 表示

〈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〉

〈无符号整数〉∷=〈数字〉{〈数字〉}

〈变量说明部分〉∷=var〈标识符〉{，〈标识符〉}；

〈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〉|〈条件语句〉|〈当型循环语句〉|

〈过程调用语句〉|〈读语句〉|〈写语句〉|〈复合语句〉|〈空〉

〈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〉

〈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〉∷=#|=|<|<=|>|>=

〈条件语句〉∷=if〈条件〉then〈语句〉

〈过程调用语句〉∷=call〈标识符〉

〈当型循环语句〉∷=while〈条件〉do〈语句〉

〈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|2|…|8|9

对上述表示法作以下几点说明：

① 由第一条产生式可以看出，PL/0 的程序是由非终结符“分程序”和终结符“.”这个串定义的。

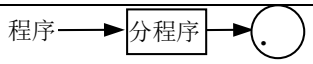
② 〈条件〉中的“odd”用来判断一个整数表达式的奇偶性，是奇数返回 1，否则返回 0。

③ 〈关系运算符〉中的“#”是不等于符号。

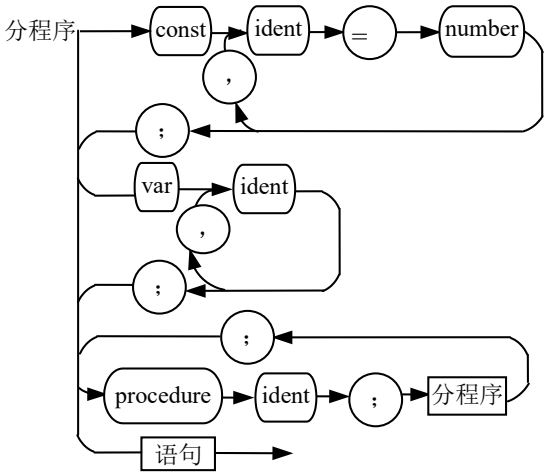
④ 〈读语句〉和〈写语句〉中的'('和')'表示这两个符号不是 EBNF 的元符号，而是读写语句中必有的一对括号，如 read(b)，write(2*c)。

2．PL/0 语言的语法图描述

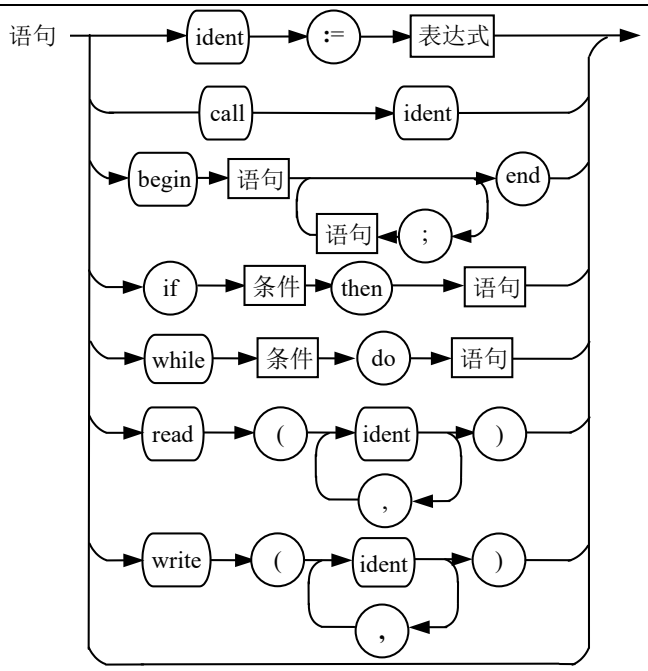
用语法图描述语法规则的优点是直观、易读。在图 2.5 所示的语法图中，用椭圆和圆圈中的英文表示终结符，用长方形内的中文表示非终结符。



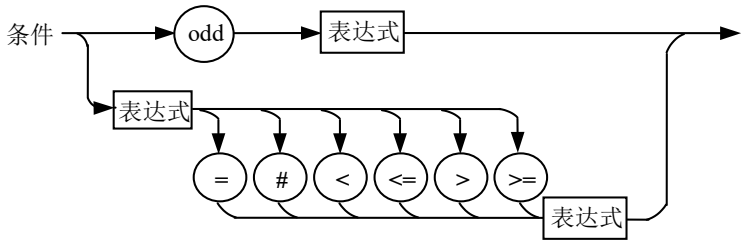
(a) 程序语法描述图



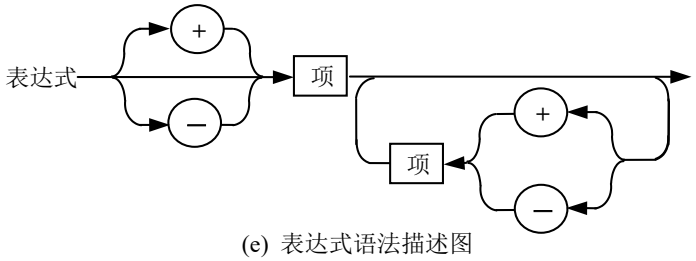
(b) 分程序语法描述图



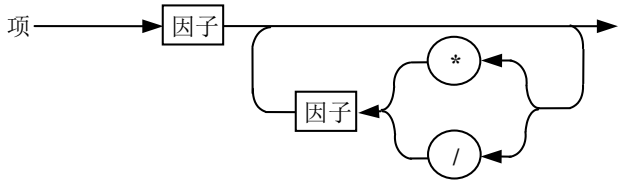
(c) 语句语法描述图



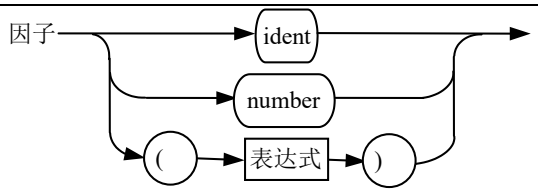
(d) 条件语法描述图



(e) 表达式语法描述图



(f) 项语法描述图



(g) 因子语法描述图

图 2.5　PL/0 语言的语法图

习　　题

1．通过阅读 PL/0 语言的文法描述，指出下列 PL/0 程序中的错误。

```
var a,b,c;
begin
   read(a,b);
   c=100
   if(a>0) then{b=b+1;write(b);}    else write(c);
   write(a,b,c);
end.
```