

**TINY+语法分析器**

**课 程： 编译原理**

**姓 名： 侯佳耀**

**学 号： 20182131055**

**班 级： 计科4班**

**设计时间：2021年12月6日**

|  |
| --- |
| 一．实验内容及目的  1、实验内容：  1.1.扩充的语法规则有：实现 do while循环，for循环，扩充算术表达式的运算符号：-= 减法赋值运算符号（类似于C语言的-=）、求余%、乘方^  1.2.扩充比较运算符号：==（等于），>(大于)、<=(小于等于)、>=(大于等于)、<>(不等于)等运算符号  1.3.新增支持正则表达式以及用于repeat循环、do while循环、if条件语句作条件判断的逻辑表达式：运算符号有 and（与）、 or（或）、 not（非） 具体文法规则自行构造，可参考云盘中参考书P97及P136的文法规则。  (1) Dowhile-stmt-->do stmt-sequence while(exp);  (2) for-stmt-->for identifier:=simple-exp to simple-exp do stmt-sequence enddo 步长递增1  (3) for-stmt-->for identifier:=simple-exp downto simple-exp do stmt-sequence enddo 步长递减1  (4) -= 减法赋值运算符号、求余%、乘方^、>=(大于等于)、<=(小于等于)、>(大于)、<>(不等于)运算符号的文法规则请自行组织。  (5)把tiny原来的赋值运算符号(:=)改为(=),而等于的比较符号符号（=）则改为（==）  (6)为tiny语言增加一种新的表达式——正则表达式，其支持的运算符号有 或(|) 、连接(&)、闭包(#)、括号( ) 以及基本正则表达式 。  (7)为tiny语言增加一种新的语句，ID:=正则表达式  (8)为tiny语言增加一种新的表达式——逻辑表达式，其支持的运算符号有 and(与) 、or (或)、非(not)。  (9)为了实现以上的扩充或改写功能，还需要对原tiny语言的文法规则做好相应的改造处理。  2、实验要求：  （1）要提供一个源程序编辑界面，以让用户输入源程序（可保存、打开源程序）  （2）可由用户选择是否生成语法树，并可查看所生成的语法树。  （3）应该书写完善的软件文档  （4）要求应用程序应为Windows界面。 |
| **二．实验过程**  2.1实验设计  2.1.1 TINY+语言语法  TINY+语言的语法为：   1. 注释：放在一对大括号内，不能嵌套； 2. 关键字：read write if end repeat until else； 3. 类型：只支持整型和布尔型； 4. 运算符：+ - \* / ( ) < = :=，其中:=为赋值运算，=为判断。没有>和<=和>=。   TINY的词法单元为：   |  |  |  | | --- | --- | --- | | 词法单元类型 | 词法单元 | 词素（例） | | 关键字  （预定义符） | IF | if | | THEN | then | | ELSE | else | | END | end | | REPEAT | repeat | | UNTIL | until | | READ | read | | WRITE | write | | 自定义符 | ID | myName | | NUM | 123 | | 运算符 | ASSIGN | := | | EQ | = | | LT | < | | PLUS | + | | MINUS | - | | TIMES | \* | | OVER | / | | LPAREN | ( | | RPAREN | ) | | SEMI | ; | | 错误 | ERROR | >= |   2.1.2 TINY+语言语法  TINY+语言的语法为：   1. 注释：放在一对大括号内，不能嵌套； 2. 关键字：read write if end repeat until else while do endwhile enddo for to downto； 3. 类型：只支持整型和布尔型； 4. 运算符：+ - \* / % ^ ( ) < == = > >= <= <>，其中=为赋值运算，==为判断,新增>= <= > <>（不等于）   TINY+的词法单元为：   |  |  |  | | --- | --- | --- | | 词法单元类型 | 词法单元 | 词素（例） | | 关键字  （预定义符） | IF | if | | THEN | then | | ELSE | else | | END | end | | REPEAT | repeat | | UNTIL | until | | WHILE | while | | ENDWHILE | endwhile | | DO | do | | ENDDO | enddo | | FOR | for | | TO | to | | DOWNTO | downto | | READ | read | | WRITE | write | | ID | myName | | 自定义符 | NUM | 123 | | ASSIGN | = | | 运算符 | EQ | == | | NEQ | <> | | LT | < | | GT | > | | NLT | >= | | NGT | <= | | PLUS | + | | MINUS | - | | TIMES | \* | | OVER | / | | REMAINDER | % | | POWER | ^ | | LPAREN | ( | | RPAREN | ) | | SEMI | ; | | 错误 | ERROR |  |   TINY+的文法为：   |  |  |  | | --- | --- | --- | | 非终结符 | 含义 | 展开 | | program | 程序 | stmt\_seq | | stmt\_seq | 若干条语句 | stmt{SEMI stmt} | | stmt | 单条语句 | if\_stmt | repeat\_stmt | assign\_stmt | read\_stmt | write\_stmt | dowhile\_stmt | while\_stmt |  for\_stmt | error | | if\_stmt | 判断语句 | IF exp THEN stmt\_seq [ELSE stmt\_seq] END | | repeat\_stmt | 循环语句 | REPEAT stmt\_seq UNTIL exp | | assign\_stmt | 赋值语句 | ID ASSIGN exp | | read\_stmt | 输入语句 | READ ID | | write\_stmt | 输出语句 | WRITE exp | | exp | 判断表达式 | simple\_exp [LT simple\_exp | GT simple\_exp | NLT simple\_exp | NGT simple\_exp | EQ simple\_exp| NEQ simple\_exp] | | simple\_exp | 加减表达式 | simple\_exp {PLUS term | MINUS term}| term | | term | 乘除表达式 | term {TIMES power | OVER power | REMAINDER power} | power | | power | 乘方表达式 | Power POWER factor | factor | | factor | 括号表达式 | LPAREN exp RPAREN | NUM | ID | error |   即：  program->stmt\_seq  stmt\_seq->stmt{;stmt}  stmt->if\_stmt  stmt->repeat\_stmt  stmt->assign\_stmt  stmt->read\_stmt  stmt->write\_stmt  stmt->error  if\_stmt->IF exp THEN stmt\_seq [ELSE stmt\_seq] END  repeat\_stmt->REPEAT stmt\_seq UNTIL exp  assign\_stmt->ID=exp  read\_stmt->READ ID  write\_stmt->WRITE exp  exp-> simple\_exp [LT simple\_exp | GT simple\_exp | NLT simple\_exp | NGT simple\_exp | EQ simple\_exp| NEQ simple\_exp]exp->simple\_exp=simple\_exp  simple\_exp->simple\_exp+term  simple\_exp->simple\_exp-term  term->term\*power  term->term/power  term->term%power  power->power^factor  factor->(exp)  factor->NUM  factor->ID  factor->error  2.2实验过程  2.2.1实验环境  硬件：ASUS笔记本（WIN11 professional）  软件：Visual studio 2022 community，MFC  2.2.2 GUI界面介绍  根据上述文法及运算符的变更，实现了相关的文法扩充功能。另外采用了C++的MFC模块进行可视化界面的编写。  可视化界面如下图：    **Button的按键功能如下：**  打开源程序：  即供主程序打开TNY格式的TINY语言源程序的按钮，可以实现读入TNY源程序并显示在源程序下方的Text框内。  保存源程序：  可保存输入的源程序（TNY格式）至指定目录  生成语法树： 可生成TINY语言的语法树  查看语法树：  可将语法树打印在语法树下方的语法树Text框内  2.2.3 样例测试  样例1.samples.TNY  结果如下：    样例2.test.TNY  结果如下：    **四、实验总结**  通过本次实验，更加深入的学习TINY语言编译器的实现，并了解了如何自己扩充TINY语言的文法规则，用理论设计来实现更简洁的语言翻译，这次的实验也凸显了文法设计的重要性。  由于时间紧迫，正则表达式和减法赋值需要调整较多程序架构，故暂未未完善，For循环发现了Bug。 |
| 五、参考文献：  （1）CSDN blog  （2）编译原理及实践，机械工业出版社，Kenneth C.Louden著，冯博琴 冯岚等译  （3）数据结构（用C++面向对象语言描述） 清华大学第二版 |