



编译原理实验报告

程序语言设计实验



2019-3-14

北京理工大学计算机学院

谭超 1120161874

语言概述:

我对 C 语言的一个语言子集的词法规则和语法规则用文法进行了描述,其中主要包括了部分表达式、赋值语句=、分支语句 if-else 和循环语句 while。

字母表:

{a..z, A..Z, 0..9, {, }, (,), [,], +, -, *, /, %, <, >, =, !, \, _ ;}

词法规则:

DIGITS->' 0' .. ' 9' ;
LETTER->' a' .. ' z' | ' A' .. ' Z' ;
LBRACE->' {' ;
RBRACE->' }' ;
LPAREN->' (' ;
RPAREN->')' ;
LBRACKET->' [' ;
BRRACKET->']' ;
ASSIGNMENT_OPERATOR->' =' ;
ALGORITHM_OPERATOR->' +' | '-' | '*' | '/' | '%' ;
RELATIONAL_OPERATOR->' <' | '>' | '<=' | '>=' | '==' | '!=' ;
WHILE->' while' ;
IF->' if' ;
ELSE->' else' ;
ENDLINE->' \n' ;
UNDERLINE->' _' ;
SEMICOLON->' ;' ;

语法规则:

开始的部分

start->statementList;

整个程序由许多个段组成

statementList->statement*;

每个段可以是一个一般表达式(以 ';' 结尾)或者 while 语句或者 if 语句

statement->expression SEMICOLON

 | whileStatement
 | ifStatement
 ;

一般表达式又可分为声明表达式、赋值表达式和逻辑表达式

Expression->declareExpression

 | assignmentExpression
 | logicalExpression
 ;

声明表达式用于声明变量

declareExpression->datatypeDeclare variableName [ASSIGNMENT_OPERATOR

expression];

赋值表达式用于给变量赋值

assignmentExpression→

variableName ASSIGNMENT_OPERATOR expression;

逻辑表达式可以是一个最基本表达式或者组合逻辑表达式

logicalExpression→primaryExpression

| logicalExpression operator logicalExpression

;

最基本表达式是不可再分的

primaryExpression→variableName

| datatype

;

while 语句

whileStatement→WHILE LPAREN expression RPAREN [LBRACE] statementList [RBRACE]

;

if 语句

ifStatement→IF LPAREN expression RPAREN [LBRACE] statementList [RBRACE] [elseStatement]

;

else 语句

elseStatement→ELSE [LBRACE] statementList [RBRACE] ;

datatypeDeclare→'int' | 'string' ;

datatype→int|string;

int→DIGITS+;

string→(LETTER|UNDERLINE) {LETTER|DIGITS|UNDERLINE}

变量名

variableName→LETTER{LETTER|DIGITS} {LBRACKET int RBRACKET};

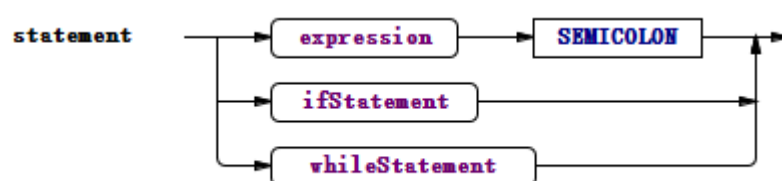
运算符

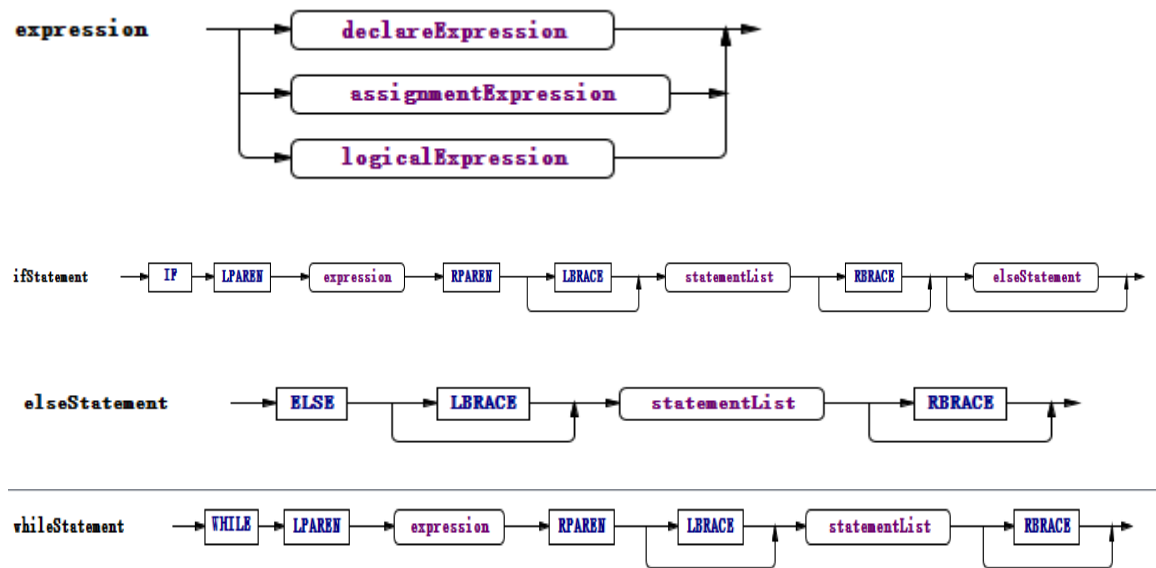
operator→ALGORITHM_OPERATOR|RELATIONAL_OPERATOR;

其他:

在这个C语言的子集中,定义了 int 和 string 两种类型的变量,算术运算符有 +、-、*、/、%, 关系运算符有<、>、<=、>=、==、!=, 另外关键字有 while、if、else。这个文法描述还存在的问题有: 运算符的优先等级没有区分。

部分语法图:





实验心得体会：

在使用 ANTLRworks 编写语法规则与词法规则的时候，能即时生成语法图使得语法规则和词法规则的编写更加简洁明了。在编写语法规则和词法规则的过程中，我对整个文法的树状结构有了一个更加直观的认识。