**编译原理课程实验报告**

**实验2：语法分析**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 王晨懿 | | 院系 | 计算机科学与技术学院 | | | | 学号 | | | 1162100102 | | |
| 任课教师 | | 辛明影 | | | | 指导教师 | 辛明影 | | | | | | |
| 实验地点 | | 格物208 | | | | 实验时间 | 2019.4.21 | | | | | | |
| 实验课表现 | | 出勤、表现得分 | | |  | 实验报告  得分 |  | | 实验总分 | | |  | |
| 操作结果得分 | | |  |
| 一、需求分析 | | | | | | | | | | 得分 | | |  |
| 要求：采用至少一种句法分析技术（SLR(1)、LR(1)）对类高级语言中的基本语句进行句法分析。阐述句法分析系统所要完成的功能。 | | | | | | | | | | | | | |
| 2文法设计 | | | | | | | | | | 得分 | | |  |
| 要求：给出如下语言成分的文法描述。   * 函数 * 声明语句（变量声明**）** * 表达式及赋值语句 * 分支语句：if\_then\_else * 循环语句：do\_while * 数组  2.1函数文法  |  |  | | --- | --- | | P' -> P | 拓广文法 起始符号P’ | | P -> func funcs | 主体由一个个函数组成 | | funcs -> func funcs | 函数 | | funcs -> ε | 函数 | | func -> type id ( args ) func\_body | 函数主体 | | type -> int | 变量类型 int | | type -> short | 变量类型 short | | type -> long | 变量类型 long | | type -> char | 变量类型 char | | type -> float | 变量类型 float | | type -> double | 变量类型 double | | type -> boolean | 变量类型 Boolean | | type -> record | 变量类型 record | | type -> void | 类型 void | | type -> unsigned type | 无符号类型 | | args -> type id arg | 首个参数 | | args -> ε | 首个参数 | | arg -> , type id arg | 参数 | | arg -> ε | 参数 | | func\_body -> ; | 函数声明 | | func\_body -> block | 函数块 | | block -> { define\_stmts stmts } | 声明语句+语句 | | factor -> id call\_func | 函数调用 | | call\_func -> ( es ) | 输入参数 | | es -> expression eps | 输入参数 | | eps -> , expression eps | 输入参数 | | eps -> ε | 输入参数 |  2.2声明语句  |  |  | | --- | --- | | define\_stmts -> define\_stmt define\_stmts | 声明语句 | | define\_stmts -> ε | 声明语句 | | define\_stmt -> type id init vars ; | 声明语句 | | init -> = expression | 声明语句 变量 | | init -> [ const ] | 声明语句 数组 | | init -> ε | 不初始化 | | vars -> , id init vars | 声明语句 | | vars -> ε | 声明语句 | | stmts -> stmt stmts | 语句 | | stmts -> ε | 语句 | | stmt -> assign\_stmt | 赋值语句 | | stmt -> jump\_stmt | 跳转语句 | | stmt -> iteration\_stmt | 迭代语句 | | stmt -> branch\_stmt | 分支语句 |  2.3表达式及赋值语句  |  |  | | --- | --- | | assign\_stmt -> expression ; | 赋值语句 | | logical\_expression -> ! expression bool\_expression | 逻辑运算 | | logical\_expression -> expression bool\_expression | 逻辑运算 | | bool\_expression -> lop expression bool\_expression | 逻辑运算 | | bool\_expression -> ε | 逻辑运算 | | lop -> && | 关系运算 && | | lop -> || | 关系运算 || | | block\_stmt -> { stmts } | 语句块 | | isnull\_expr -> expression | 语句可空 | | isnull\_expr -> ε | 语句可空 | | expression -> value operation | 一般表达式 | | operation -> compare\_op value | 比较语句 | | operation -> equal\_op value | 赋值语句 | | operation -> ε | 操作符 | | compare\_op -> > | 比较符号 | | compare\_op -> >= | 比较符号 | | compare\_op -> < | 比较符号 | | compare\_op -> <= | 比较符号 | | compare\_op -> == | 比较符号 | | compare\_op -> != | 比较符号 | | equal\_op -> = | 赋值符号 | | value -> item value' | 算数运算 | | value' -> + item value' | 算数运算 + | | value' -> - item value' | 算数运算 - | | value' -> ε | 算数运算 | | item -> factor item' | 算数运算 | | item' -> \* factor item' | 算数运算 \* | | item' -> / factor item' | 算数运算 / | | item' -> % factor item' | 算数运算 % | | item' -> ε | 算数运算 | | factor -> ( value ) | 高优先级 括号 | | factor -> const | 常数 | | factor -> id | 变量 | | factor -> id [ const ] | 数组 | | factor -> id call\_func | 函数调用 |  2.4分支语句  |  |  | | --- | --- | | branch\_stmt -> switch ( id ) { case\_stmt case\_stmts default\_stmt } | 分支语句 switch | | case\_stmts -> case\_stmt case\_stmts | switch-case | | case\_stmts -> case const : stmts | switch-case | | case\_stmt -> ε | switch-case | | default\_stmt -> default : stmts | switch-default | | default\_stmt -> ε | switch-default | | branch\_stmt -> if ( logical\_expression ) block\_stmt result | 分支语句 if-else | | result -> else block\_stmt | if-else | | result -> ε | if-else |  2.5循环语句  |  |  | | --- | --- | | jump\_stmt -> continue ; | 跳转语句 continue | | jump\_stmt -> break ; | 跳转语句 break | | jump\_stmt -> return isnull\_expr ; | 跳转语句 return | | iteration\_stmt -> while ( logical\_expression ) block\_stmt | while循环 | | iteration\_stmt -> for ( isnull\_expr ; isnull\_expr ; isnull\_expr ) block\_stmt | for循环 | | iteration\_stmt -> do block\_stmt while ( logical\_expression ) ; | do-while循环 |  2.6数组  |  |  | | --- | --- | | init -> [ const ] | 声明语句 数组 | | factor -> id [ const ] | 数组使用 | | | | | | | | | | | | | | |
| 3系统设计 | | | | | | | | | | 得分 | | |  |
| 要求：分为系统概要设计和系统详细设计。  （1）系统概要设计：给出必要的系统宏观层面设计图，如系统框架图、数据流图、功能模块结构图等以及相应的文字说明。  （2）系统详细设计：对如下工作进行展开描述   * 核心数据结构的设计 * 主要功能函数说明 * 程序核心部分的程序流程图  3.1系统概要设计3.1.1系统框架图3.1.2数据流图  3.1.3功能模块结构图3.2系统详细设计3.2.1核心数据结构的设计3.2.2主要功能函数说明3.2.3程序核心部分的程序流程图 | | | | | | | | | | | | | |
| **4系统实现及结果分析** | | | | | | | | | | 得分 | | |  |
| 要求：对如下内容展开描述。   1. 系统实现过程中遇到的问题； 2. 输出该句法分析器的分析表； 3. 针对测试程序输出其句法分析结果； 4. 输出针对此测试程序对应的语法错误报告； 5. 对实验结果进行分析。   注：其中的测试样例需先用已编写的词法分析程序进行处理。 | | | | | | | | | | | | | |
| 指导教师评语：  日期： | | | | | | | | | | | | | |