电子科技大学 信息与软件工程 学院

标 准 实 验 报 告

（实验）课程名称 编译技术

电子科技大学教务处制表

电 子 科 技 大 学

实　　验　　报　　告

学生姓名：　　邓萌达　　学 号：　2018091620008

指导教师：　周尔强

实验地点：　　信软楼 400　　实验时间：　2019/12/05

一、实验室名称：信软 400

二、实验项目名称：LR 语法分析

三、实验学时：4 学时

四、实验内容及步骤：

- ast.h

  定义语法树的头文件，构造语法树类型 past

```
#endif //HANDIN_AST_H

# include "define.h"
# include <stdlib.h>
# include <stdio.h>

extern FILE* yyout;

typedef struct _ast ast;
typedef struct _ast* past;
struct _ast{
    char* token_type;
    char* value;
    past left;
    past right;
};
```

- ast.c

  提供构造语法树和输出语法树的方法，new_node 和 show_tree。

```c
past new_node(char* token_type, char* value, past l, past r){
    past node = (past) malloc(sizeof(ast));
    if (node == NULL) {
        puts("ERROR! out of memory");
        exit(0);
    }
    node->token_type = token_type;
    node->value = value;
    node->left = l;
    node->right = r;
    return node;
}

void show_tree(past node, int high){
    if (node == NULL) return;
    if (node->token_type == token_Compound_list) {
        show_tree(node->left, high);
        show_tree(node->right, high);
        return;
    }

    for (int i = 0; i < high; i++) fprintf(yyout,"\t");
    fprintf(yyout, "%s: %s\n", node->token_type, node->value);
    show_tree(node->left, high + 1);
    show_tree(node->right, high + 1);
}
```

- define.h

  定义语法输出的 token:

```c
# define token_Compound_list "Compound_list"
# define token_external_decl "Decl_Stmt"
# define token_Decl_List "Decl_List"
# define token_number "number_lvalue"
# define token_string "string_lvalue"
# define token_v "Value_varible"
# define token_v_decl "Value_Decl"
# define token_func_decl "Func_Decl"
# define token_instr_list "Instr_List"
# define token_v_list "value_list"
# define token_para "parameter"
# define token_type_id "type"
# define token_stat_list "Stmt_List"
# define token_rt_stat "Ret_Stmt"
# define token_IF_stat "If_Stmt"
# define token_While_stat "While_Stmt"
# define token_Pint_stat "Print_Stmt"
# define token_Scan_stat "Scan_Stmt"
```

- lrlex.l

  对于相应的词法生成相应的 token，并记录一些在 lrparser.y 中需要的值。

```
"}"                 { return yytext[0];                              }
"=="                { cmp_text = strdup(yytext); return CMP;         }
"!="                { cmp_text = strdup(yytext); return CMP;         }
">="                { cmp_text = strdup(yytext); return CMP;         }
"<="                { cmp_text = strdup(yytext); return CMP;         }
{INTEGER}           { return number;                                 }
{FLOAT}             { return number;                                 }
{STRING}            { return string;                                 }
"+="                { assign_text = strdup(yytext); return ASSIGN;   }
"-="                { assign_text = strdup(yytext); return ASSIGN;   }
"*="                { assign_text = strdup(yytext); return ASSIGN;   }
"/="                { assign_text = strdup(yytext); return ASSIGN;   }
"%="                { assign_text = strdup(yytext); return ASSIGN;   }
">"                 { cmp_text = strdup(yytext); return CMP;         }
"<"                 { cmp_text = strdup(yytext); return CMP;         }
"int"               { return INT;                                    }
"string"            { return STR;                                    }
"str"               { return STR;                                    }
"void"              { return VOID;                                   }
"if"                { return IF;                                     }
"else"              { return ELSE;                                   }
"while"             { return WHILE;                                  }
"print"             { return PRINT;                                  }
"scan"              { return SCAN;                                   }
"return"            { return RETURN;                                 }
{IDENTIFIER}        { ID_text = strdup(yytext); return ID;           }
```

- lrparser.y

  运用 bison，重新定义 yyval 为 past

```
char* yytext;
#define YYSTYPE past
```

  利用 bison 分析词法时候的栈，进行建树：

```
S: program { show_tree($1, 0); }
    ;

program: external_declaration { $$ = $1; }
       | program external_declaration { $$ = new_node(token_Compound_list, ""
       ;

external_declaration: function_definition { $$ = $1; }
                    | declaration { $$ = $1; }
                    ;

function_definition: type declarator compound_statement { $$ = new_node(token_
       ;

declaration: type init_declarator_list ';' { $$ = new_node(token_decl, "", $1,
       ;

init_declarator_list: init_declarator { $$ = $1; }
                    | init_declarator_list ',' init_declarator { $$ = new_node
                    ;

init_declarator: declarator { $$ = $1; }
               | declarator '=' add_expr { $$ = new_node(token_Compound_list
```

- main.c

  进行文件输入输出：

```c
extern FILE* yyin;
extern FILE* yyout;

extern int yyparse();

int main(int argv, char* argc[]) {
    if (argv >= 2){
        yyin = fopen(argc[1], "r");
        if (yyin == NULL) {
            printf("Can not open the file: %s\n", argc[1]);
            return 404;
        }
    }
    if (argv >= 3){
        yyout = fopen(argc[2], "w");
    }

    yyparse();
    return 0;

}
```

- Makefile

  生成各个中间文件，编译 main 可执行文件。

```
main: ast.c main.c lrparser.tab.c lex.yy.c
    clang ast.c main.c lrparser.tab.c lex.yy.c -o main
lex.yy.c: lrparser.tab.h lrlex.l
    flex lrlex.l
lrparser.tab.c: lrparser.y
    bison -d lrparser.y
lrparser.tab.h: lrparser.y
    bison -d lrparser.y
```

- README.md

  介绍本实验运行的环境和使用的方法。

```
## Eviroment
```
System: MAC OS 10.15.1
C Compiler: Apple clang version 11.0.0 (clang-1100.0.33.12)
Flex: flex 2.5.35 Apple(flex-32)
Bison: bison (GNU Bison) 2.3
```

## Usage

```
make main
./main [inputfile] [outputfile]
```
```

五、实验运行结果：

测试 test.c 文件：

六、实验结论与总结：

1. 这次的实验可以较好的满足了实验的所有的要求，并且可以正常的运行。
2. 这次实验运用了 bison，flex 两个语法分析/词法分析的工具，让我学习并掌握了这两个工具的运用，使得这次实验相比于上次自己完全写语法分析代码量少了很多，过程也更加清晰。
3. 通过这次实验，让我对 LR 分析法更加熟悉，了解了归约和消除的分析方法，让我对语法分析有了更加深刻的理解。
4. Makefile 的初步学习和使用简化了编译时的复杂度。
5. 每一次写编译技术实验都能了解到一些关于 c 语言以及其编译器的新知识，编译技术实验不同于其他“学生管理系统”之类的实验，具有一定的实用性和挑战性：各种工具的运用、较大较复杂的代码量。每一次的实验都能让我学习到不同的知识，让我了解到并改进我的不足。
6. 感谢周尔强老师授课以及对我这次实验中的耐心指导

报告评分：

指导教师签字：