实验报告

-- 编译原理 实验一

姓名：梁宇方 学号：171860695

邮箱：leungjyufong2019@outlook.com

a）我的程序实现了所有的必做功能和选做功能

I）必做内容

1）检查 C 词法中没有定义过的字符：在 lexical.l 文件中使用正则表达式和响应函数来检查出”定义过”的 TOKEN；剩余的字符就是没有定义过的字符，使用通配符来匹配，相应的响应函数调用 yyerror 报错.

2）检查语法错误：在 syntax.y 文件中使用 Bison 的语法来重写 C 语法；最后能规约到起始符号 Program 的情况为没有语法错误，反之则有语法错误，程序会自动调用 yyerror 进行报错.

3）打印语法树：对于没有语法错误的程序，自底而上构建一颗抽象语法树，然后从根结点开始深度优先遍历．在词法分析器中构造叶子结点，在语法分析器中构造中间结点.

II）选做内容

1）识别八进制和十六进制数：修改 lexical.l 文件中对 INT 的正则定义，使之可以包含八进制数和十六进制数的情况，即

```
dec [1-9][0-9]*|0
oct 0[0-7]*
hex 0[xX][0-9a-fA-F]*
INT {dec}|{oct}|{hex}
```

2）浮点数的科学计数法表示：同上，前导可以是浮点数或整数

```
flt [0-9]+\.[0-9]*|\.[0-9]+
FLOAT {flt}|{flt}[eE][-\+]?[0-9]+|{dec}[eE][-\+]?[0-9]+
```

3）行注释和块注释：在词法分析器处使用正则表达式识别这些注释，将其响应函数设置为空函数，这部分内容不会反馈给语法分析器，因此不会影响原有结构

```
LINECOMMENT \/\/.*
BLOCKCOMMENT \/\*[^*]*\*+([^\/*][^*]*\*+)*\/
```

b）直接使用默认的 makefile 进行编译，即键入 make，即可完成编译

c）亮点

1） 加入了部分错误恢复功能，遇到部分语法错误时可以继续进行语法分析

```
Error type A at Line 2: Invalid ID '0main'.
Error type B at Line 3: syntax error.
Error type B at Line 6: syntax error.
Error type B at Line 8: syntax error.
Error type B at Line 12: syntax error.
Error type B at Line 17: syntax error.
Error type B at Line 23: syntax error.
Error type B at Line 24: syntax error.
Error type B at Line 26: syntax error.
Error type B at Line 29: syntax error.
Error type B at Line 31: syntax error.
Error type B at Line 34: syntax error.
Error type B at Line 36: syntax error.
Error type B at Line 37: syntax error.
Error type B at Line 41: syntax error.
Error type A at Line 41: Mysterious character '"'.
Error type A at Line 41: Mysterious character '"'.
Error type B at Line 43: syntax error.
Error type A at Line 44: Mysterious character '&'.
Error type B at Line 45: syntax error.
Error type A at Line 45: Invalid ID '9a6c'.
Error type A at Line 47: Mysterious character '?'.
Error type A at Line 47: Mysterious character '?'.
Error type B at Line 47: syntax error.
Error type B at Line 48: syntax error.
lanceloia@lanceloia-UX310UQK:~/Documents/cp2020/Lab$
```

2）区分了减号和负号的优先级和结合性，并能构建出正确的语法树

对于表达式 8--3*5，给出了语法树(片段)为

```
Exp (2)
  INT: 8
MINUS
Exp (2)
  Exp (2)
    MINUS
    Exp (2)
      INT: 3
  STAR
  Exp (2)
    INT: 5
```

对于表达式----2-3，给出了语法树(片段)为

```
Exp (2)
  MINUS
  Exp (2)
    MINUS
    Exp (2)
      MINUS
      Exp (2)
        MINUS
        Exp (2)
          MINUS
          Exp (2)
            INT: 2
MINUS
Exp (2)
  INT: 3
```