
编译原理实验1：词法分析与语法分析

实验报告

吴澄杰

151220122

wuchengjie@smail.nju.edu.cn

2018年5月5日

一、实验进度

此次试验，我使用词法分析工具 GNU Flex 和语法分析工具 GNU Bison 实现了一个对 C 语言源代码进行词法分析和语法分析的程序。

我完成了实验的必做要求：程序可以检测出源代码中的词法错误和语法错误并报告相应错误类别，可以在源代码没有词法和语法错误的情况下输出语法分析树。此外，我完成了所有的选做要求：我的程序可以识别并处理八进制数、十六进制数、指数形式的浮点数、单行注释和多行注释。我的程序通过了实验要求中的所有 4 个必做样例和 6 个选做样例。

二、编译与实验

在工程目录下，输入 make 命令，即可完成编译，可生成程序 parser。用 C 语言源代码文件名作为参数运行程序 parser 就可以得到在该文件上的词法分析和语法分析结果。

工程目录下的 example 文件夹中有 4 个必做样例的源代码和正确结果，example_ext 文件夹中有 6 个选做样例的源代码和正确结果。若要检测程序在样例上运行结果的正误，只需将样例文件名作为参数给到 parser 运行即可。比如，`./parser ./example/example1.txt` 就可以得到程序在第一个测试样例上的运行结果。

三、具体功能描述

1. 词法分析

使用词法分析器生成工具 Flex，只需要定义好各词法单元的正则表达式，就可以方便地完成词法分析。为了实现选做要求，我还实现对八进制数、十六进制数、指数形式浮点数和注释的正则表达式表述。具体实现在文件 lexical.l 中。

若词法分析器匹配到一种词法单元，就会生成一个终结符节点，并且将当前词素的属性保存在该节点中。若任何一种词法单元都没有匹配到，那么词法分析器就会报错，输出 A 类型的词法错误。

2. 语法分析

使用语法分析器生成工具 Bison，定义好非终结符、终结符、属性类型和产生式，就可以生成语法分析器进行语法分析。终结符就是所有种类的词法单元，这与词法分析器中的

定义是一致的。非终结符及其产生式则按照题目附录中所要求的定义进行实现。具体实现在文件 syntax.y 中。

为了构造语法树，需要保存每个词素的属性值。在 grammarTree.h 中，我定义了属性值的结构体：

```
struct Node {
    union {
        int type_int;
        float type_float;
        double type_double;
        char type_string[32];
    };
    int line;
    char type[32];
    int num_children;
    struct Node* children[MAX_CHILDREN];
    struct Node* parent;
    bool terminal;
};
```

其中，line 记录了该词素出现的行号，type 记录了该词素的类型，num_children 记录在语法分析树中该词素的子节点个数，children 数组记录了指向子节点的指针，parent 记录了指向语法分析树中父节点的指针，terminal 记录了该语法符号是否是终结符。type_int, type_float, type_double, type_string 则具体记录了一些语义值。

在词法分析过程中，每个被检测出的词素都会被赋予一个节点。在语法分析的过程中，每个非终结符也会有结构体，并且在自底向上分析过程中，在产生式被归约的时候，建立产生式左端符号到产生式右边符号在语法分析树中的边，并设置产生式左端符号的属性。申请语法树节点、设置语法树属性值、建立语法树边的相关实现在 grammarTree.c 中。

如此一来，若输入正确的 C 语言源代码，那么语法分析结束后，语法树就已经建成了。只要按照实验要求打印语法树即可。

3. 消除冲突

完成上述内容后编译，会发现 Bison 输出了存在冲突的警告。因此需要设置运算符的结合性和优先级来消除冲突。我按照 C 语言的标准设置了结合性和优先级后，就成功地消除了冲突。

4. 报告错误

报告词法错误比较简单。只要在词法分析过程中发现了未定义的词法单元，输出错误，并标明为 A 类型词法错误即可。

为了报告不同的语法错误，并且在发生语法错误后能够继续语法分析过程，就需要使用到 Bison 错误恢复的功能，要设置特殊的产生式放置特殊符号 error 的位置。在发生错误时，Bison 会根据 error 符号进行再同步，继续语法分析过程。

例如，在本实验中，为了检测并报告缺少分号、右方括号的错误，就可以用 `error` 代替原产生式中的分号、右方括号，并且在这条特殊产生式的动作中打印出错误信息，并标明是 B 类型语法错误。

四、实验总结

此次实验，接触并熟悉到了 Flex 和 Bison 工具的使用，在实践中熟悉了词法分析和语法分析的过程，并且通过实践，掌握了优先级设置消除冲突、设置 `error` 符号恰当地报告语法错误等各项技巧，最终成功地实现了实验要求的各项功能。