实验 9 Yacc 分析程序生成器

实验难度：★★☆☆☆
建议学时：2 学时

一、 实验目的

- 掌握 Yacc 输入文件的格式。
- 掌握使用 Yacc 自动生成分析程序的方法。

二、 预备知识

- 要求已经学习了 BNF（上下文无关文法），能够正确编写简单的 BNF。
- 熟练掌握了各种自底向上的分析方法，特别是 Yacc 所使用的 LALR(1)算法。
- 本实验使用 Yacc 的一个实现版本——GNU Bison 作为分析程序生成器。

三、 实验内容

3.1 阅读实验源代码

sample.txt 文件

此文件是 Yacc 的输入文件。根据 Yacc 输入文件的格式，此文件分为三个部分（由%%分隔），各个部分的说明可以参见下面的表格。

名称	说明
第一部分	在%{和%}之间的直接插入 C 源代码文件的内容。包括了要包含的头文件，以及用于指示分析程序输出调试信息的 YYDEBUG 预定义类型。
第二部分	定义了一个简单的文法规则“A -> (A) | a”，但是，对规则进行匹配时，没有执行任何操作。
第三部分	这部分中的内容会直接插入 C 源代码文件。此部分内容的说明可以参见下面的表格。

内容	说明
main 函数	在 main 函数中首先声明了 Yacc 的全局变量 yydebug，然后将其赋值为 0（0 表示不启动调试），最后调用 yyparse 函数对从标准输入读取到的内容进行分析。
yylex 函数	此函数从标准输入读取字符并进行相关的处理。首先会忽略包括空格在内的空白，然后，如果输入了回车，就返回 0，让分析程序停止，否则就返回从标准输入读取到的字符，用于和文法进行匹配。
yyerror 函数	当从标准输入读取到的字符与文法匹配失败时，会调用此函数输出错误信息。

ytab.c 文件

此文件是 Yacc 输出的 C 源代码文件。当使用 Yacc 处理 sample.txt 文件时，就会生成此文件。新建项目中，此文件的内容是空的。

ytab.h 文件

此文件是 Yacc 输出的头文件。为 Yacc 使用选项“--defines=ytab.h”时，就会生成此文件。此文件可以被包括在需要使用 Yacc 所生成的定义的任何文件中。新建项目中，此文件的内容是空的。

y.output.txt 文件

此文件是 Yacc 输出的文件。为 Yacc 使用选项“--report-file=y.output.txt”时，就会生成此文件。此文件包含了被分析程序使用的 LALR(1)分析表的文本描述。新建项目中，此文件的内容是空的。

y.output.html 文件

此文件是 y.output.txt 文件内容的 HTML 语言表示,可以使用更加直观的方式显示分析表的信息。新建项目中，此文件的内容是空的。

y.dot.txt 文件

此文件是 y.output.txt 文件内容的 DOT 语言表示，可以使用图形化的方式显示 DFA 自动机。新建项目中，此文件的内容是空的。

3.2 生成项目

按照下面的步骤生成项目：

1. 按 Ctrl+Shift+B,在弹出的下拉列表中选择“生成项目”。

1) 在生成的过程中，首先使用 Bison 程序根据输入文件 sample.txt 来生成各个输出文件，然后，将生成的 ytab.c 文件重新编译、链接为可以运行的可执行文件。
2) 如果成功生成了 ytab.c 文件，读者可以在“TERMINAL”运行以下命令并按回车，使用 DOTTY 程序来打开 y.dot.txt 文件，可以使用图形化的方式查看 DFA 自动机。

```
dotty y.dot.txt
```

3) 读者可以在“EXPLORER”窗口中右击 y.output.html 文件，在弹出的菜单中选择“Open Preview”,打开此文件,其内容与 y.output.txt 文件类似，但是查看更加方便直观。(注意，如果浏览器中显示乱码，需要将浏览器的编码改为“UTF-8”，简单的修改方法是在乱码页面中点击右键，选择“编码”中的“UTF-8”)。
4) 在生成的 ytab.c 文件中,尝试找到 sample.txt 文件中第一部分和第三部分 C 源代码插入的位置，并尝试查找 yylex 函数和 yyerror 函数是在哪里被调用的。

提示：如果需要使用 DOTTY 程序手动打开 y.dot.txt 文件，只需要在“TERMINAL”中输入“dotty.exe”并按回车，然后在 DOTTY 程序中点击右键，选择菜单中的“load graph”，打开项目目录中的 y.dot.txt 文件。

3.3 运行项目

在没有对项目的源代码进行任何修改的情况下，按照下面的步骤运行项目：

1、 选择“Run”菜单中的“Run Without Debugging”(快捷键 Ctrl+F5)。

在“TERMINAL”中输入“(a)”字符串后按回车，扫描程序不会输出任何错误信息，说明文法匹配成功。而如果在“TERMINAL”窗口中输入类似“()”或“b”字符串后按回车，就会输出默认的错误信息。

3.4 编写一个简单的计算器程序

按照下面的步骤完成此练习：

1. 修改 sample.txt 文件中的内容，实现一个简单的计算器程序，其文法如下（粗体表示终结符）：

   exp → exp addop term | term

addop → + | -

term → term mulop factor | factor

mulop → *

factor → (exp) | number

2. 重新生成项目。如果生成失败，根据“TERMINAL”窗口中的提示信息修改源代码中的语法错误。
3. 按 Ctrl+F5 启动执行项目。

当在“TERMINAL”窗口中输入表达式“2+3”后按回车，可以正确计算出表达式的结果。

注意：

1、 如果执行的结果不正确，可以通过添加断点和单步调试的方法来查找错误的原因。

2、 断点应该添加在 sample.txt 文件中需要中断的 C 源代码行，不要添加在 ytab.c 文件中，否则无法命中断点。

3.5 通过验证

按照下面的步骤继续实验：

1. 按 Ctrl+Shift+B，然后在下拉列表中选择“测试”，启动验证。在“TERMINAL”窗口中会显示验证的结果。如果验证失败，可以在“EXPLORER”窗口中，右击“result_comparation.html”文件，在弹出的菜单中选择“Open Preview”，可以查看用于答案结果文件与读者编写程序产生的结果文件的不同之处，从而准确定位导致验证失败的原因。

四、思考与练习

1. 尝试为计算器文法绘制 LALR(1)的分析表，并绘制表达式“2+3”的分析动作表。提示：将 main 函数中的 yydebug 赋值为 1 后，就可以在 Windows 控制台窗口中获得分析程序的分析动作。
2. 尝试为计算器程序添加整除运算符“/”，并可以为包含整除运算符的表达式计算出正确的值。
3. 修改计算器的 YACC 输入文件，使之能够输出以下有用的错误信息：
   - 为表达式“(2 +3”生成错误信息“丢失右括号”
   - 为表达式“2+3)”生成错误信息“丢失左括号”
   - 为表达式“2 3”生成错误信息“丢失运算符”
   - 为表达式“(2+)”生成错误信息“丢失操作数”