

实验一

Chenning Yu 151242062 匡亚明学院 151242062@smail.nju.edu.cn

程序应该如何被编译？

- `make all` 会依次按照Syntax/syntax.y生成Syntax/syntax.tab.c, Syntax/syntax.tab.h, 按照Lex/lexical.l生成Lex/lex.yy.c, 最后生成与Makefile处于同一文件夹下的parser。该parser能够用来测试实验指南中出现的样例。
- `make debug` 是为了协助我debug语法生成过程的命令。它与 `make all` 生成parser的过程一致，不过这里的parser会开启诊断模式。
- `make test1` , `make test2` , `make test3` , `make test4` 分别对应于实验指南中的必做样例1到样例4。
- `make test5` , `make test6` , `.....` , `make test10` 分别对应于实验指南中的选做样例1到样例6。

对于助教测试，建议先敲入 `make all` 重新生成一遍parser。

如需测试已有样例，则使用上述的 `make testN` 命令； `N` 为对应的样例编号。

如需测试新的样例，请使用 `./parser path/to/testfile` 命令； `path/to/testfile` 对应新的样例的相对路径。

实验实现了哪些功能？

完成了实验一的必做与选做部分的所有内容。粗体字为本代码亮点。

- 词法分析
 - 在Lex/lexical.l中创建叶节点。
 - 检测错误的八进制数和十六进制数。
 - 支持识别 `//` 以及非嵌套 `/**/` 的注释。

- 语法分析
 - 在Syntax/syntax.y中创建非叶子节点，并给父节点添加子节点。
 - 语法树以先序插入实现。
 - 有针对不同情况的错误恢复。
 - 有针对性的错误恢复的不同的提示。如针对test2，parser会输出：

```
Error type B at line 5: syntax error. Unexpected near ',', Wrong Exp between [ and ].
Error type B at line 6: syntax error. Unexpected near 'else', Wrong Stmt between ) and ELSE.
```

- 恢复之后的语法树可以进行输出。此功能需要使用 `./parser path/to/testfile -debug` 命令，可以看见程序恢复错误语法之后生成的语法树。

实验总结

这次实验比较轻松。我熟悉了Lex与Bison，并用他们分析了C--，进一步了解了课本知识。在实验中我由于测试样例的输出与预期不对，debug了很久，最后发现是在Lex/lexical.l中的正则匹配的 `*` 没有加转义符号。这表明我在编写此类规则时仍需要注意空格、引号，以及给字符转义。