

H1 L2 语法分析 实验报告

181250015 陈彦泽

H2 1 实现功能

- 在L1的基础上，实现语法树、语法错误的输出

H2 2 实现方法

- 修改L1的词法分析器，使其返回词法单元，并设置 `yylval`
- 添加语法树节点的数据结构，并定义添加子节点、输出语法树的方法
- 在 `syntax.y` 中书写文法规则，生成语法树
- 判断是否存在语法、词法错误，若存在则输出错误信息，若不存在则输出语法树

H2 3 踩坑录

H3 redefinition

添加文件 `Node.h` 和 `Node.c` 后报错

```
Node.h:1:16: error: redefinition of 'struct TreeNode'
typedef struct TreeNode{
           ^~~~~~
In file included from syntax.y:3:0:
Node.h:1:16: note: originally defined here
typedef struct TreeNode{
           ^~~~~~
```

网上查了资料，发现可能是因为头文件中的循环引用导致错误

解决方案：在 `Node.h` 中加上

```
#ifndef NODE_H_
#define NODE_H_
...
#endif
```

H3 MINUS歧义

根据优先级表，有两个MINUS，`%left PLUS MINUS`和`%right NOT MINUS`，直接写果然编译报错了

```
./src/syntax.y:58.1-6: error: %right redeclaration for MINUS
%right MINUS NOT
^^^^^^
./src/syntax.y:56.1-5:      previous declaration
%left PLUS MINUS
^^^^^^
```

删掉第二个MINUS，大部分情况下没有问题，但还是在用例hard-9上出现了问题。

用例大致如下

```
struct { int c = a + 1 - -b().X/2;};
```

```

Exp (2)
  MINUS
  Exp (2)
    Exp (2)
      ID: b
      LP
      RP
      DOT
      ID: X
    DIV
    Exp (2)
      INT: 2

```

这里的优先级发生了错误，应该先和MINUS Exp规约，再做Exp DIV Exp规约

尝试了很久没有找到解决方案，想到关于悬空else问题里用的`%prec`，网上搜索一下，果然可以解决这个问题。

将优先级表改为`%right NOT UMINUS`，文法改为`MINUS Exp %prec UMINUS`，问题就解决了。

H3 一些冲突

对于这个用例

```
int Y ( int pdBwaV ) {{ { }}
```

会报错

```

entering state 3
reading a token: Next token is token RC (3.3: )
error type B at Line 3: syntax error,near '}'
shifting token error (3.3: )
entering state 11

```

查了一下状态，是因为

```

State 3
14 StructSpecifier: LC . error RC
error shift, and go to state 11

```

解决方案：

```
StructSpecifier : ... | LC error RC{errorSyntaxFlag = 1;}
```

改为

```
StructSpecifier : ... |STRUCT OptTag LC error
RC{errorSyntaxFlag = 1;}
```

就正常了