Task2-语法分析器

学号	姓名
20319045	刘冠麟

实验结果

```
[build] functional-3/058_scope.sysu.c ..................................... 100.00/100.00
[build] functional-3/059_sort_test1.sysu.c ................................ 100.00/100.00
[build] functional-3/060_sort_test7.sysu.c ................................ 100.00/100.00
[build] functional-3/061_empty_stmt.sysu.c ................................ 100.00/100.00
[build] functional-3/062_side_effect.sysu.c ............................... 100.00/100.00
[build] functional-3/063_nested_calls2.sysu.c ............................. 100.00/100.00
[build] functional-3/064_while_if.sysu.c .................................. 100.00/100.00
[build] mini-performance/00_bitset1.sysu.c ................................ 100.00/100.00
[build] mini-performance/01_mm1.sysu.c .................................... 100.00/100.00
[build] mini-performance/crypto-1.sysu.c .................................. 100.00/100.00
[build] mini-performance/dead-code-elimination-1.sysu.c ................... 100.00/100.00
[build] mini-performance/fft0.sysu.c ...................................... 100.00/100.00
[build] mini-performance/hoist-1.sysu.c ................................... 100.00/100.00
[build] mini-performance/if-combine1.sysu.c ............................... 100.00/100.00
[build] mini-performance/instruction-combining-1.sysu.c ................... 100.00/100.00
[build] mini-performance/integer-divide-optimization-1.sysu.c ............. 100.00/100.00
[build]
[build] task2
[build] 总分（加权）：100.00/100.00
[build] ================================================================================
[build] 成绩单已保存：  /Users/liuguanlin/SYsU-lang2/build/test/task2/score.txt
[build] JSON 格式：  /Users/liuguanlin/SYsU-lang2/build/test/task2/score.json
[driver] 生成完毕: 00:00:09.464
[build] 生成已完成，退出代码为 0
```

改进建议

某些测试样例安排得不太妥当

测例中出现的情况个人觉得应该循环递进，将新的结构或者类型一起出现容易让人难以定位问题，比如测例35:

```c
#include <sysy/sylib.h>
// test if-else-if
int ifElseIf() {
  int a;
  a = 5;
  int b;
  b = 10;
  if(a == 6 || b == 0xb) {
    return a;
  }
  else {
    if (b == 10 && a == 1)
      a = 25;
    else if (b == 10 && a == -5)
      a = a + 15;
    else
      a = -+a;
  }

  return a;
}

int main(){
  return 0;
}
```

在这个测例中**if-elseif-else、逻辑与或和函数的定义**同时第一次出现。

我认为应该将**考察函数定义单独划分为一种类型的测例**，而不是混杂在其他类型的测例中，否则会让人以为是当前测例考察的文法出了错误，从而花费大量的时间重复推导、编写原本已经正确的文法和cpp代码。

我在实验时其他的if测例都能通过，唯独这一条不能通过，我一开始以为是if-else的文法定义或者`||` `&&`的定义有问题，修改和尝试很久以后才逐渐定位到原来是函数的声明没有能够处理导致出错，导致浪费了大量的时间在错误的方向上。

应该提示一下`#include`语句考察的实际上是函数的声明（或者说提示一下会转换成函数声明）

在测例根目录中`test\cases`中未编译的源码如下：

```c
#include <sysy/sylib.h>
```

然而实际上真正进行测试输入的源码是在`build`文件夹下的`test\cases`，在这里面源码中的include会被转化成一系列函数声明：

```
# 1 "/Users/liuguanlin/SYsU-lang2/test/rtlib/include/sysy/
# 10 "/Users/liuguanlin/SYsU-lang2/test/rtlib/include/sysy
void _sysy_starttime(int lineno);
void _sysy_stoptime(int lineno);


int _sysy_getch();
void _sysy_putch(int a);

int _sysy_getint();
void _sysy_putint(int a);

int _sysy_getarray(int a[]);
void _sysy_putarray(int n, int a[]);
# 2 "/Users/liuguanlin/SYsU-lang2/test/cases/functional-3/
```

我一开始发现凡是有`#include`的测例都出现了断错误，本来以为是没能处理include，反复查看lexer、文法和测例后才发现是没能处理函数声明

应该提供较为简便的测试自己编写测例的途径

现在的版本想要测试自己修改或者编写的测例来定位问题，只能修改根目录`test\cases`下的测例并且删除`build`文件夹后再重新构建才能测试，这个过程比较漫长，很浪费时间，而且有忘记备份修改测例的风险。

还有群中提到的**修改task1\answer.txt**的方法治标不治本，这种方法**只是修改了输入的token流，用来对比的答案并不会因此改变**，只能用来定位导致ABORT或者断错误等导致文件输出错误的问题，对于其他情况需要更改源代码进行定位的情况并不适用。比如对于原本能通过测例09：

```
[build] functional-0/009_hex_defn.sysu.c ............................. 100.00/100.00
```

我删除answer.txt中选中的行：

```
int 'int'    [StartOfLine]  Loc=</Users/liuguanlin/SYsU-lang
identifier 'main'    [LeadingSpace] Loc=</Users/liuguanlin/S
l_paren '('     Loc=</Users/liuguanlin/SYsU-lang2/test/cases
r_paren ')'     Loc=</Users/liuguanlin/SYsU-lang2/test/cases
l_brace '{'     Loc=</Users/liuguanlin/SYsU-lang2/test/cases
int 'int'    [StartOfLine] [LeadingSpace]   Loc=</Users/liug
identifier 'a'   [LeadingSpace] Loc=</Users/liuguanlin/SYsU-
semi ';'        Loc=</Users/liuguanlin/SYsU-lang2/test/cases
identifier 'a'    [StartOfLine] [LeadingSpace]    Loc=</Users/
equal '='    [LeadingSpace]Loc=</Users/liuguanlin/SYsU-lang
numeric_constant '0xf'    [LeadingSpace]Loc=</Users/liuguanl
semi ';'        Loc=</Users/liuguanlin/SYsU-lang2/test/cases
return 'return'  [StartOfLine] [LeadingSpace]   Loc=</Users/
identifier 'a'   [LeadingSpace] Loc=</Users/liuguanlin/SYsU-
semi ';'        Loc=</Users/liuguanlin/SYsU-lang2/test/cases
r_brace '}'  [StartOfLine]  Loc=</Users/liuguanlin/SYsU-lang
eof ''      Loc=</Users/liuguanlin/SYsU-lang2/test/cases/fun

```

测试的结果分数反而下降了，推测是因为答案已经生成（根据根目录下的源代码），修改answer.txt后反而导致没有识别被删除的token从而导致分数下降：

```
[build] functional-0/009_hex_defn.sysu.c ............................. 27.27/100.00
```

所以我觉得应该提供一种快速测试自定义测例的途径。

助教团队都非常非常认真细心负责!!教学上已经非常好了，这点不需要改进^ ^