report

解决方法记录

1. 进制

- 将`s\t`从`auto s = tk.substr(b, e - b).str(), t = tk.substr(0, tk.find(" ")).str();`变为`auto s = tk.substr(b, e - b), t = tk.substr(0, tk.find(" "));`
- 增加

```
    long num;
    s.getAsInteger(0, num);
    yylval = new Tree("IntegerLiteral", "", std::to_string(num));
```

- 修改`yylval = new Tree("id", s);`为`yylval = new Tree("id", s.str());`

2. if-else

```
%right IF_THEN T_ELSE
Stmt: T_IF T_L_PAREN Cond T_R_PAREN Stmt  %prec IF_THEN
| T_IF T_L_PAREN Cond T_R_PAREN Stmt T_ELSE Stmt
```

3. block中含有block
   改写block item

```
BlockItem: Stmt {

}
| BlockItem Stmt {
    $1->addBrother($2);
    $$ = $1;
}
| Block {

}
| BlockItem Block {
    $1->addBrother($2);
    $$ = $1;
}
;
```

不能改写stmt增加block, 会导致许多rs、rr冲突, 可以利用以下命令查看

```
bison /workspace/SYsU-lang/parser/parser.y -Wcounterexamples
```

4. 多维数组
   初始版本: 不能过`int a[1][1]={{}};`, **存疑**

```
InitVal: Exp {

```

```
}
| InitValList {
    auto ptr = new Tree("InitListExpr");
    ptr->addSon($2);
    if(!$2->brothers.empty()){
        ptr->addSons($2);
    }
    $$ = ptr;
}
;

InitValList: Exp {
    // int a[1] = {1};
}
| InitValList T_COMMA Exp {
    // int a[2] = {1, 2};
    $1->addBrother($3);
    $$ = $1;
}
| T_L_BRACE InitValList T_R_BRACE {
    // int a[2][2] = {{1,2}};
}
| InitValList T_COMMA T_L_BRACE InitValList T_R_BRACE {
    // int a[2][2] = {{1,2}, {3,4}};
    $1->addBrother($4);
    $$ = $1;
}
| {
    // int a[1] = {};
}
;
```

5. 处理LVal的ImplicitCastExpr与()的父子关系
   对于

```
(a);
```

clang 输出:

```
|     `-ImplicitCastExpr 0x24761b0 <col:10, col:12> 'int' <LValueToRValue>
|       `-ParenExpr 0x2476190 <col:10, col:12> 'int' lvalue
|         `-DeclRefExpr 0x2476170 <col:11> 'int' lvalue Var 0x2475ea8 'a'
'int'
```

对于

```
(1);
```

clang 输出:

```
|-ParenExpr 0x21c4000 <line:17:5, col:7> 'int'
| `-IntegerLiteral 0x21c3fe0 <col:6> 'int' 1
```

解决方法:

1. 特判

```
PrimaryExp: T_L_PAREN Exp T_R_PAREN {
    // 特判ImplicitCastExpr
    if($2->kind == "ImplicitCastExpr"){
      auto ptr = new Tree("ImplicitCastExpr");
      $2->kind = "ParenExpr";
      ptr->addSon($2);
      $$ = ptr;
    }
    else{
        auto ptr = new Tree("ParenExpr");
        ptr->addSon($2);
        $$ = ptr;
    }
}
```

2. LVal增加RHS: `(` LVal `)`

```
LVal:
| T_L_PAREN LVal T_R_PAREN {
    auto ptr = new Tree("ParenExpr");
    ptr->addSon($2);
    $$ = ptr;
}
;
```

会带来s-r冲突:

```
/workspace/SYsU-lang/parser/parser.y: warning: 1 shift/reduce conflict [-Wconflicts-sr]
/workspace/SYsU-lang/parser/parser.y: warning: shift/reduce conflict on token T_R_PAREN [-Wcounterexamples]
    Example: T_L_PAREN LVal . T_R_PAREN
    Shift derivation
        PrimaryExp
        `-> LVal
            `-> T_L_PAREN LVal . T_R_PAREN
    Reduce derivation
        PrimaryExp
        `-> T_L_PAREN Exp                       T_R_PAREN
                    `-> AddExp
                        `-> MulExp
                            `-> UnaryExp
                                `-> PrimaryExp
                                    `-> LVal .
```

指定优先级LVal先与右括号`)`结合:

```
%right PrimLVal T_R_PAREN
PrimaryExp:
    | LVal %prec PrimLVal
```

6. 浮点精度
   https://github.com/arcsysu/SYsU-lang/discussions/77

7. 隐式转换1：

```
float a, b, c=a+b;
```

处理到`c`时符号表还没有`a,b`，方法：默认为int（实在没有办法了...）

8. 函数返回值的隐式转换
思路：DFS检查`return`语句

9. 补充完调用含参函数的`params_type`，再看看前面的`/workspace/SYsU-lang/tester/function_test2020/50_recursion_test1.sysu.c`寄掉了。显然这是来检查递归函数的，而递归函数调用自身时，在符号表中还没有自己的名字（因为函数的Block还没处理完）。此时不能使用`params_type[0]`，因为第一个参数类型并不存在（函数名不在符号表）。

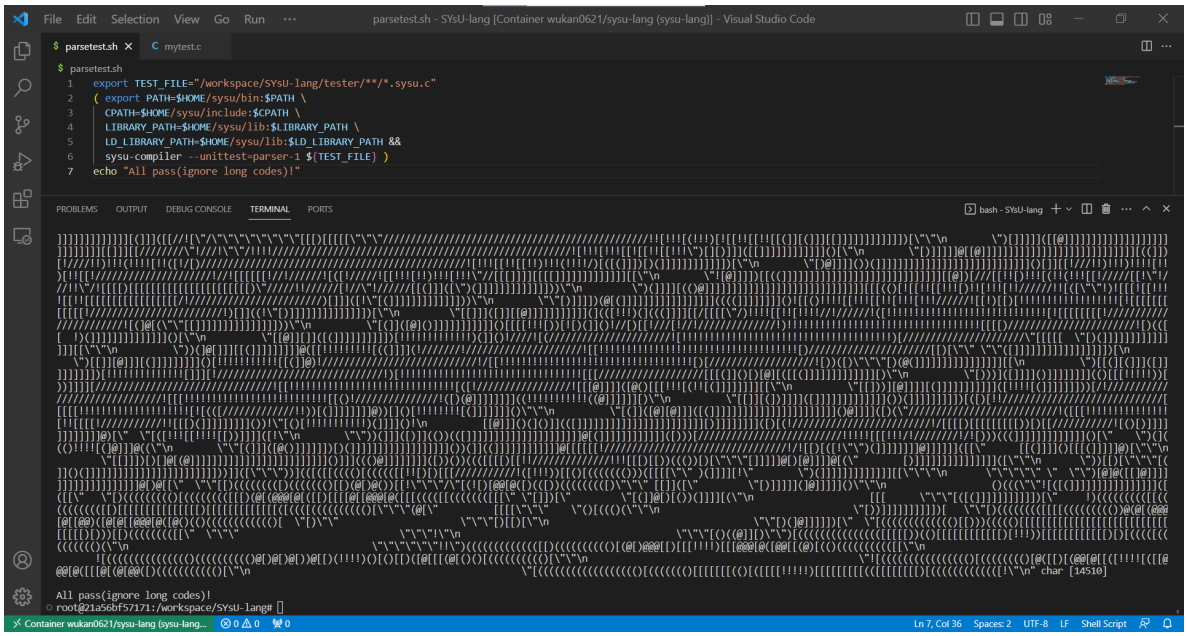
10. 解决`/workspace/SYsU-lang/tester/function_test2022/95_float.sysu.c`
    1. 函数参数列表含省略号
    2. 二元运算根据优先级进行类型隐式转换，注意不含逻辑运算符
    3. 解决`float arr[10] = {1., 2};`，涉及到隐式转换、`array_filler`。

运行结果

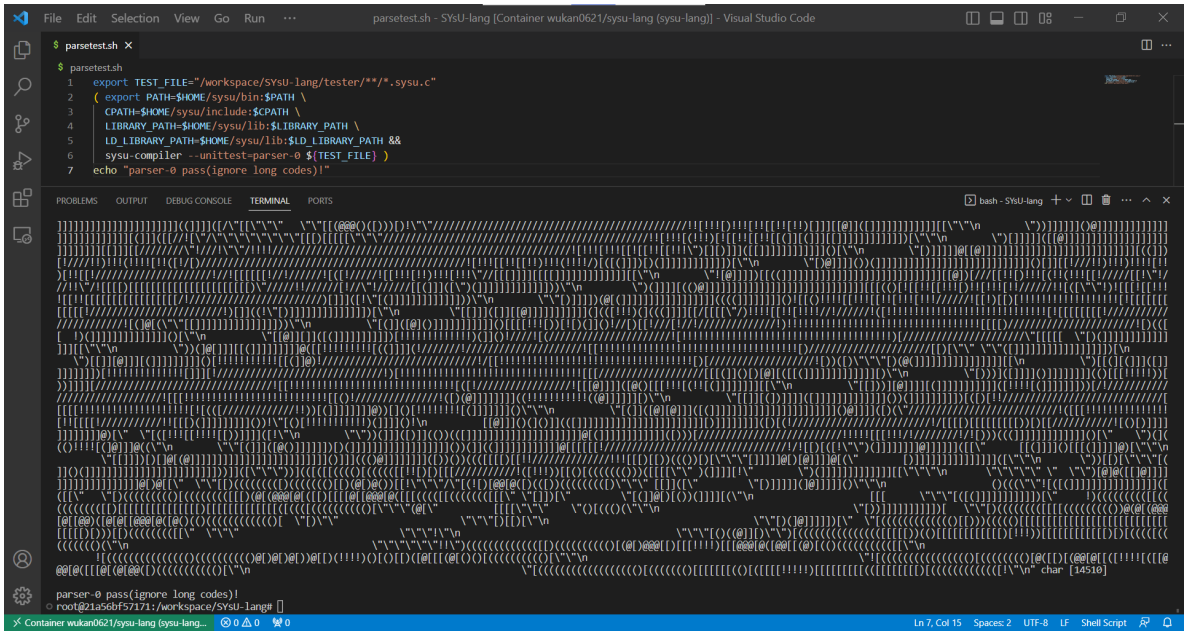
去除长样例：

- function_test2022/86_long_code2.sysu.c
- h_functional/107_long_code2.sysu.c

可以通过`parser-1`测试：



`parser-0`测试：


File Edit Selection View Go Run
parsetest.sh - SYsU-lang [Container wukan0621/sysu-lang (sysu-lang)] - Visual Studio Code
parsetest.sh
export TEST_FILE="/workspace/SYsU-lang/tester/**/*.sysu.c"
(export PATH=$HOME/sysu/bin:$PATH \
CPATH=$HOME/sysu/include:$CPATH \
LIBRARY_PATH=$HOME/sysu/lib:$LIBRARY_PATH \
LD_LIBRARY_PATH=$HOME/sysu/lib:$LD_LIBRARY_PATH &&
sysu-compiler --unittest=parser-0 ${TEST_FILE})
echo "parser-0 pass(ignore long codes)!"
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - SYsU-lang
parser-0 pass(ignore long codes)!
root@21a56bf57171:/workspace/SYsU-lang#
Container wukan0621/sysu-lang (sysu-lang...
Ln 7, Col 15 Spaces: 2 UTF-8 LF Shell Script