

# report

| 学号       | 姓名  |
|----------|-----|
| 20337091 | 马佳欣 |

## Deadcode

### 不使用的局部变量、函数参数、函数

在generator, 使用is\_promotable判断。

- 优点：在generator阶段就可以实现，不需要生成了ir再删除ir。
- 注意：main不会在其他地方被调用，所以判断函数是否可优化时要判断是否为main函数。

### 强度削减

在generator, 在BinaryExprToIR处判断。

- 优点：在generator阶段就可以简单实现，在optimizer阶段实现代码更复杂
- 优化方向：
  1. Algebraic Identity
$$x + 0 = 0 + x, x \times 1 = 1 \times x \Rightarrow x$$
 // 只能消除整型，不能消除浮点类型
  2. Strength Reduction
$$2^n \times x = x \times 2^n \Rightarrow x \ll n$$

## Mem2Reg

### 优化流程

- 在代码中已有详细的注释。

### 注意事项

- 值栈 `SmallDenseMap<AllocaInst *, std::stack<Value *>>value_stack_of_alloca` 需要 `undef` 兜底值：  
`phi` 节点对应的值可能是 `undef` 的，例如：

```
1 void func(){
2     int a;
3     while (a>0)
4     {
5         a = a-1;
6     }
7 }
```

对于 `while.cond` 中的 `a` 变量的 `phi` 节点，从 `entry` 进入的值是未知的。

```

1  define void @func() {
2  entry:
3      br label %while.cond
4
5  while.cond:                                ; preds =
    %while.body, %entry
6      %a0 = phi i32 [ undef, %entry ], [ %1, %while.body ]
7      %0 = icmp sgt i32 %a0, 0
8      br i1 %0, label %while.body, label %while.end
9
10 while.body:                                ; preds = %while.cond
    %1 = sub i32 %a0, 1
12     br label %while.cond
13
14 while.end:                                ; preds = %while.cond
    ret void
15 }
16

```

- `to_remove_loads` 和 `to_remove_stores` :

不要在遍历 `inst` 的时候删除, 会导致迭代器失效, 所以需要先暂存起来。

## 问题和解决方法

1. `tester/function_test2020/02_arr_defn4.sysu.c` :

```

1  %a = alloca [4 x [2 x i32]], align 4
2  %0 = bitcast [4 x [2 x i32]]* %a to i8*
3
4  /tmp/tmpb16m_r1m/a.ll:6:34: error: expected type
5  %0 = bitcast [4 x [2 x i32]]* <badref> to i8*

```

- `tester/function_test2020/35_array_test.sysu.c` :

数组取地址也是 `<badref>`

- 解决方法: 不优化数组。

2. `tester/function_test2020/10_break.sysu.c` :

```

1  while.cond:                                ; preds = %if.end,
    %entry
2      %0 = icmp sgt i32 10, 0
3      br i1 %0, label %while.body, label %while.end
4      %a0 = phi i32 [ 10, %entry ], [ %1, %if.end ]
5
6  /tmp/tmp73d4c49h/a.ll:11:35: error: '%1' defined with type 'label' but
    expected 'i32'
7  %a0 = phi i32 [ 10, %entry ], [ %1, %if.end ]

```

循环中 `a` 值会发生变化, 且根据 `a` 值判断是否继续执行循环, 但是上述 `phi` 节点被放在了 `block` 的最后面, 导致出错。

- 解决方法: 将 `phi` 节点插在 `block` 的最前面。

3. 使用了 `Mem2Reg` 优化会出错 (段错误) 的测例:

1. `tester/function_test2022/73_int_io.sysu.c`
2. `tester/h_functional/090_int_io.sysu.c` (与1是一样的)

3. `third_party/SYsU-lang-tester-performance/performance_test2021-public/median2.sysu.c`
4. `/workspace/SYsU-lang/long_tester/third_party/SYsU-lang-tester-performance/performance_test2022-private/vector_mul2.sysu.c`

解决方法：在 `main` 函数判断即将优化的函数是否为测例里面的函数，若是则不进行优化。

#### 4. 优化除法

强度削减中一开始包括除法优化，但是到了某个测例就出错了，因此放弃除法优化。

## 排行榜截图

| 名次 | 提交时间                | performance         | score |
|----|---------------------|---------------------|-------|
| 11 | 2023-06-13 22:29:03 | 0.32236309425226268 | 618   |
| 12 | 2023-06-12 21:54:46 | 0.3090350485604904  | 638   |
| 13 | 2023-06-12 11:37:36 | 0.30397913584877856 | 639   |
| 14 | 2023-06-14 16:29:04 | 0.29665277186288697 | 645   |
| 15 | 2023-06-18 03:10:33 | 0.25939452440309374 | 638   |
| 16 | 2023-06-09 15:16:29 | 0.2420917261194229  | 633   |
| 17 | 2023-05-26 12:54:58 | 0.22373143827305925 | 639   |
| 18 | 2023-06-13 19:54:58 | 0.22017170019715684 | 639   |
| 19 | 2023-06-22 15:56:33 | 0.2187655401970091  | 639   |
| 20 | 2023-05-16 19:06:03 | 0.20284044371482365 | 636   |

## 挑战工作

- lexer:

- 扩展更多 C 语言的 token

- 踩过的坑：
  - 不支持数据类型 `bool`，`clang` 将其识别为 `identifier`;
  - 不需要写 `long long` 类型对应的规则，`clang` 将其识别为两个 `long`。
- 更多 token:

|    | token    | name     |
|----|----------|----------|
| 1  |          |          |
| 2  |          |          |
| 3  | :        | colon    |
| 4  | auto     | auto     |
| 5  | double   | double   |
| 6  | long     | long     |
| 7  | enum     | enum     |
| 8  | register | register |
| 9  | short    | short    |
| 10 | signed   | signed   |
| 11 | unsigned | unsigned |
| 12 | static   | static   |
| 13 | struct   | struct   |
| 14 | union    | union    |
| 15 | typedef  | typedef  |

|    |         |              |
|----|---------|--------------|
| 16 | do      | do           |
| 17 | switch  | switch       |
| 18 | case    | case         |
| 19 | default | default      |
| 20 | for     | for          |
| 21 | goto    | goto         |
| 22 | sizeof  | sizeof       |
| 23 | extern  | extern       |
| 24 | ++      | plusplus     |
| 25 | +=      | plusequal    |
| 26 | --      | minusminus   |
| 27 | -=      | minusequal   |
| 28 | *=      | starequal    |
| 29 | /=      | slashequal   |
| 30 | %=      | percentequal |

2. 根据新扩展的 `token`，编写新的测例 `tester/mytest.c`。