

南京大学本科生实验报告

课程名称：编译原理

学院	计算机科学与技术	专业（方向）	计算机科学与技术
学号	201220215	姓名	田馥雯
Email	1259067849@qq.com	实验名称	实验五：中间代码优化

实验内容

使用数据流分析算法等消除效率低下和无法被轻易转化为机器代码的中间代码，从而将C—源代码翻译成的中间代码转化为语义等价但是更加简洁高效的版本。

完成了除B4和C3的全部内容

[summary]: your score is 98

[score-log]: {'A-1': 10, 'A-2': 10, 'A-3': 10, 'A-4': 10, 'A-5': 10, 'A-6': 10, 'C-1': 4, 'C-2': 4, 'C-3': 2, 'C-4': 4, 'B': 24}

```
1 [B-4-score]: Runtime overhead: 34%/40% (the runtime overhead of optimized IR is 499,
   expected is 439, borderline is 465), score: 94%
2 [C-3-score]: Runtime overhead: 3%/40% (the runtime overhead of optimized IR is 2068,
   expected is 2010, borderline is 2060), score: 63%
```

编译：通过 `makefile` 进行make、make test的指令即可执行

核心代码

在框架代码的基础上进行，下面给出需要做的todo位置所增加的东西

`available_expressions_analysis.c`

- forward analysis
- boundary: `OutFact[Entry] = empty set`
- initialization: `InitFact = universal set`
- must analysis

`constant_propagation.c`

- `meetValue`：计算不同数据流数据汇入后变量的 `CPValue` 的meet值

从大到小为UNDEF->CONST->NAC，而汇聚结果取得是最小集

需要注意的是，两者均为CONST时，若值一致则返回CONST，否则返回NAC

```

1  if (v1.kind == CONST && v2.kind == CONST)
2      {
3          if (v1.const_val == v2.const_val)
4              return v1;
5          else
6              {
7                  return get_NAC();
8              }
9      }

```

此函数整体在软件分析用java实现过

- `calculateValue`：计算二元运算结果的 `CPValue` 值

重点：当v2为0且运算符为÷时，无论v1是什么结果都是UNDEF

```

1  if (v2.kind == CONST && v2.const_val == 0 && IR_op_type == IR_OP_DIV)
2      return get_UNDEF();

```

此函数整体在软件分析用java实现过

- forward analysis
- 函数参数初始化为NAC

`copy_propagation.c`

- forward analysis
- boundary: empty
- initialization: universal
- 后面三个todo是类似的，举例一个

```

1  /* TODO:
2   * def is killed by new_def
3   * VCALL(fact->def_to_use/use_to_def?, delete, def/new_def?)
4   */
5  VCALL(fact->use_to_def, delete, new_def);
6  VCALL(fact->def_to_use, delete, def);

```

live_variable_analysis.c

- backward analysis
- meet: union
- 先执行kill再执行gen
- def→kill use→gen
- 不存在后续使用的代码为死代码

solver.c

backward的内容和forward相反即可

总结与感想

代码错误及解决方法

罕见的没debug的一次实验……

感悟

隔壁同学的框架代码写的实在是太好了，这是我这学期写的最轻松的实验了，只用做个关键操作的选择，只要理清不同数据流分析的direction、may/must、boundary、initialization、transfer和meet的条件就行了，而这些内容，软件分析那边都学过写过，就顺顺利利写完啦~

然后对于整个编译原理实验做个总结吧，虽然写的磕磕绊绊但还是都争取满分ac，还是很有成就感的，代码能力应该是有提升的……吧？

最后，感谢老师和助教gg们的指导，祝你们工作、学习顺顺利利~