

Lab5 实验报告

姓名：周琰轲

学号：201850187

功能

初步完成了必做内容，但依然具有较大的提升空间。

设计特色

输入解析方法

重新设计了针对中间代码的 lex 和 bison 解析文件，可以直接将输入的 ir 文件解析为相应的数据结构。

常量折叠设计

建立一张类似与DAG的表。分析可知在基本块内的局部中间代码中，由于中间代码是顺序执行的，因此在我们分析某一条中间代码时，只需看右边的变量是否在之前赋值为常数且未修改过。因此解决常量传播问题可以采用如下算法：

1. 建立一张表，记录下所有出现过的被赋值为常数或者可以计算出常数的变量的表。
2. 在碰到赋值或者三地址计算时，查询右边操作数在表中是否有效，如果有效，则可以替换为常量；否则不可替换。
3. 对于赋值、三地址计算、CALL、READ语句，其中都有某个操作数的值发生改变。每当出现这种情况，按照第二步的方法检查，如果该操作数的值可被计算则替换并加入该表；否则设为无效。
4. 循环上述三步来遍历一个基本块中所有的中间代码。

局部公共子表达式消除

分析可知一旦某个中间代码可以利用公共子表达式规约，则在基本块内一定能在之前的中间代码中找到一直的表达式。因此没遍历一个中间代码，查找该中间代码中哪些操作数发生了改变。对于值发生改变的操作数，选择在表中删去与之相关的中间代码。

同时，如果在表中找到相同的公共子表达式，则可以将该中间代码归约为 ASSIGN 类型中间代码。并将该公共子表达式插入表中方便后续查询检测。

编译方式

在Code目录下make parser得到相应的可执行文件parser。利用指令parser <file_name> <output_file>则可得到相应的翻译结果。