

编译原理实验三报告

胡俊豪 181240020@smail.nju.edu.cn

1 功能

本实验完成了实验手册中所描述的所有功能，并通过了所有测试样例。

1.1 测试说明

在实验中，我们为一个非常奇葩的要求——数组间直接赋值专门设计了代码，被嵌入到“semantic.c”文件中，如果需要查看代码，可以搜索“array assignment”关键字，这个关键字作为注释被插入到源代码中。这个关键字所在的 if 条件语句分支专门处理数组间直接赋值。处理的逻辑是：将等式两边所代表的数组元素进行比较，以元素个数最少（大小最小）的那个数组为标准，按位复制相应数量元素。通过一个 for 循环，不断按位复制即可。

对于结构体，因为实验手册上明确说明不会有结构体之间的直接赋值，所以我们本来想略去了这部分代码，但是最后惊奇的发现，在 code assignment 注释下写出的代码，对于数组和结构体均适用。因为其基本思想，无非就是，把后面一连串的内存一次按位复制过去，结构体也一样。

2 如何编译

本实验的编写测试环境，文件夹结构，Makefile 内容均与手册上一致，只需要在 Code 文件夹下使用命令 make，就可以得到一个名为 parser 的可执行文件。输入“./parser test-file.cmm out.ir”即可完成对文件的解析，得到的中间代码保存在 out.ir 文件中。

3 设计特色

3.1 规约

本程序为了简化代码和程序逻辑，做出了一些取舍。比如，我们生成的中间代码异常冗长。这是因为我们做了如下规约：

首先，禁止立即数的直接使用。例如，禁止 $t1 = t2 + \#5$ ，允许 $t3 = \#5$ ， $t1 = t2 + t3$ 。

禁止取地址符号的直接使用。例如，禁止 $t1 = t2 + \&t3$ ，允许 $t4 = \&t3$ ， $t1 = t2 + t4$ 。

禁止 $*$ 的直接使用。例如禁止 $t1 = t2 + *t3$ ，允许 $t4 = *t3$ ， $t1 = t2 + t4$ 。

这样一来，所有和取地址符号，星号，立即数这些相关的复杂操作，全部可以被 ASSIGNOP 一种操作统一起来。换句话说，我们只需要在 ASSIGNOP 这个运算符里面讨论所有复杂情况，在别的运算符（例如加减乘除与或非）的分支下，只需要处理简单的临时变量之间的运算，例如 $t1 = t2 + t4$ 。

最后，如果所有的变量和临时变量都统一用前缀 t 表示，那么所有的运算，都可以用最简单的四元式表示。例如 $t12 = t15 + t100$ 可以表示为 (codeADD, 12, 15, 100) (codeADD 是 enum 类型的某一个取值)。唯一需要小心处理的是 ASSIGNOP 里面的各种情况，把所有复杂情况转换成单一的临时变量形式，传递给上层。

与此同时，我们对函数的名字做相同的处理，把函数进行编号 1, 2, 3... (除了 main 函数需要把名字专门写出来，让虚拟机识别到程序的开始)，并用前缀 L 表示函数，LABEL 也作相同处理。

这样一来，所有的操作真正可以用一个四元式表示，且这个四元式里面仅包含整型数字，形如 (codeFUNCTION, 2, 0, 0)，表示 FUNCTION F2:。

当然，如最开头所说，简洁代码和程序逻辑的代价是，生成的中间代码非常冗长。其中有大量的立即数赋值操作。不过这个问题非常好解决。我们生成的中间代码保存在一组四元式当中，我们令起一个模块，对这个四元式组进行代码优化即可。这样做可以把代码生成个代码优化完全分隔开，模块化。在最后的效应就是，可以加入编译选项，选择优化程度 (如-O1, -O2, -O3)。我们的编译器也可以分级，但是为了简便起见，就不分级了，只分“有优化”和“没有优化”。

最终，按照这种规约方式，我们在实验二的基础上，只增加了不到 200 行的代码 (包括自己就占据了六七十行的，用于把中间代码打印出来的 print_code 函数以及各种结构体定义)。核心代码应该只有一百多行。

3.2 代码优化

首先我们需要一个漫长的周期，进行基本块的划分，流图的建立，DAG 图维护的代码，以及其他各种数据结构的准备。这一部分编写和测试的时间，相比于之后真正开始写优化代码来说，要长得多，以至于，连一点代码都没优化，时间却过去了大部分。

最后我们进行增量式的优化。一点一点地探讨可优化的方案。按照教材上介绍的顺序，我们一次探讨块内优化以及块间优化。块内优化依次考虑利用公共子表达式，消除死代码，利用代数性质进行优化等。对于块间优化同样考虑公共子表达式，死代码等等。一点一点地加进去。

最后由于时间原因，我们在准备基本块的时候就花费了大量时间，加上近期各种夏令营，各种其他准备步步紧逼，实在没有时间完成这个有趣的工作，只能当作编译实验的一大

遗憾处理了。

4 致谢

感谢刘春旭和张思拓两位同学，课余饭后的讨论，使得在实验过程中累积的疑惑与不解得到逐一解决。