

编译原理实验 3（许畅班）

091220097 陶冰阳 me@sunicy.com

实现内容

基本实验要求及所有选做。

文件结构

gtree.c/gtree.h: 语法树模块（模块命名 `gt_*`）
stable.c/stable.h:(Symbol Table)符号表模块，用以维护 Scope 和 Hash Table
semantic.c/semantic.h:语义分析模块，用以实现有限状态机
imnode.c/imnode.h:中间代码链表及其所有相关函数
interm.c/inerm.h:中间代码生成状态机
im_reduce.c/im_reduce.h:中间代码优化模块
main.c: 主程序

编译及运行

检查并编译（默认输出文件为 `cmmc`）: `make`

清空编译临时文件: `make clean`

强制重新编译: `make all`

运行（默认为标准输出，最高优化）: `cmmc <源文件名> <目标文件> [优化级别]`
其中优化级别为可选项，0-10 级别依次提高。

不进行优化: `cmmc <源文件名> <目标文件> no`

注：由于测试程度有限，如果高级别优化不能产生正确结果，烦请老师降低优化级别进行运行，谢谢！

优化

短路代码

如: $B \rightarrow B1 \mid B2$ 生成代码（链表）`B1.true=NULL|B1.false=B.false|`
`B2.true=B.true|B2.false=B.false|B1.code|B2.code`

常量计算

在生成中间代码的有限状态机中，直接处理形如“ $1+2$ ”、“ $1<2$ ”的表达式，其结果直接存储在节点中。
可以处理的表达式包括 $x=1+2*3+4$ 等，不能处理的表达式包括 $x=1+2+x$ （因为 2 与 x 不能求值）

赋值冗余变量约简

原始: <code>t10:=Expr</code> <code>t2:=t10</code>	约简后: <code>t2:=Expr</code>
---	-------------------------------

临时变量连续常数运算约简

原始: (<code>t2</code> 之后不再出现) <code>t2:=t1+#3</code> <code>t3:=t2+#10</code>	约简后: <code>t3:=t1+#13</code>
--	---------------------------------

这里要求运算符一致，分别对+、-、*、/进行约简。

对同一变量连续赋值（运算）约简

原始： t2:=t1+#3 t2:=t1+t17	约简后： t2:=t1+t17
--------------------------------	--------------------

IF GOTO 约简

原始： IF x REL y GOTO label1 GOTO label2 : LABEL label1:	约简后： IF x (REL 逆) y GOTO label2 LABEL label1:
---	---

GOTO-LABEL 约简

原始： GOTO label1 LABEL label1:	约简后： LABEL label1:
-------------------------------------	-----------------------

空 LABEL 约简

对没有访问过的 LABEL 进行删除。