

编译原理 MiniDecaf 编译器实验

Step 6 实验报告

李祥泽

2018011331

lixiangz18@mails.tsinghua.edu.cn

实验内容

实现 if 语句

按实验指导书生成 IR 与汇编.

悬吊 else 二义性是通过产生式优先级处理的. 所有的 else 都和前面最接近的 if 相配对.

实现三目表达式

三目表达式可以用 if 语句来体现.

例如, 对于三目表达式 `expr_A ? expr_B : expr_C`, 它就可以被表示成 `if (expr_A) IR.push(expr_B) else IR.push(expr_C)`.

其区别在于, if 语句中所含有的是两个**语句**, 而三目表达式中是两个**表达式**, 后者的作用在于将其本身的值压入 IR 的运算栈, 以供更外层的语句或表达式使用.

思考题

Rust 和 Go 语言中的 if-else 语法与 C 语言中略有不同, 它们都要求两个分支必须用大括号包裹起来, 而且条件表达式不需要用括号包裹起来. 相比 C 的语法, 这两种语言的语法有什么优势?

条件表达式是否需要括号更多的是书写习惯的区别, 没有本质上的差别.

两个分支都强制要求括号包裹, 可以认为是通过大括号深度对 if 和 else 语句做了标记, 而只有同深度的 if 和 else 才能配对. 这样一来, 嵌套的 if 事实上变成了不同的符号, 不会对深度不同的 else 起反应了. 而在相同深度, 两个相连的 if 无歧义地表明前一个不含 else. 这就彻底地消除了悬吊 else 二义性.

Honor Code

主要参考实验指导书实现.