

我的程序实现了对函数定义、调用、局部变量的翻译。

对于函数定义、调用的翻译借助了HashMap保存函数的LLVMTypeRef,通过这个类型可以获得函数的返回值类型。对于变量是否需要load, 我设计了一个isLoadedValue的继承属性。如果遇到需要store的变量, 比如出现在return 语句中的exp, exp展开后的每一个变量都是需要load的。我在处理return ctx的子ctx之前将isLoadedValue赋为true, 在后续的调用中继承给子节点; 而对于赋值语句, 等号左边的值是不需要load的, 等号右边需要load, 就在处理lval函数处特殊判断其直接父节点是不是赋值stmt, 如果是, 说明它就是等号左边的值, 不需要load。

对于局部变量的翻译, 借助了符号表, 保存每个局部变量的信息。

本实验遇到的比较大的Bug是对函数调用起初没有考虑到函数类型的差异。对于void函数, 不应该将call的结果赋值给一个变量, 而是直接call ()。