编译原理实验：Lab5

指导老师：徐辉

2021 年秋季学期

1 实验介绍

Lab5 是整个系列实验的最后一步，其内容与 Lab4 一致，同样是完善编译器让其支持更多的语法，不过这个最终 lab 为一个发散性 lab，对最终完成的形式没有明确要求，可以按照个人的喜好进行自由发挥，尝试添加自己感兴趣的内容。

1.1 语法产生式

本实验虽然没有一个明确的最终实现要求，但为了方便验收，有个 baseline 的要求，即实现下述的语法产生式，如表 1所示：

其中，加粗部分为本次实验的语法产生式与之前不同的地方，主要是对可编译的代码增加了以下几个支持点：

- 增加了控制流语句，包括分支和循环。
- 分支表达式为 if 表达式，基本与 c 语言的一致。
- 循环表达式 baseline 要求为实现 while 表达式，比较容易。形式与 c 语言一致。
- baseline 的要求中控制流语句的分支只需要满足支持一个单独的 statement，不需要实现的 block 功能。

1.2 实验内容

本次实验要修改的部分同样是贯穿整个编译器的。对于控制流部分的 codegen 实现，需要进行 Basicblock 的创建和分支跳转语句的使用，具体可以参考 LLVM Tutorial 的 Control Flow 章节。

对于拓展部分的内容，同学可以结合各种语言的特性来进行实现，这里列出一些可以参考的实现内容：

- 增加 char 类型的实现
- 增加 for 类型控制流的实现
- 增加控制流的 block 的实现，即 $if(){}$, 分支中可有多个语句
- 增加 declare 时的直接赋值，如 $int\ i = 2$

$< program >$	$::=$	$< gdecl >^* < function >^*$
$< gdecl >$	$::=$	$extern < prototype >;$
$< function >$	$::=$	$< prototype >< body >$
$< prototype >$	$::=$	$< type >< ident > (< paramlist >)$
$< paramlist >$	$::=$	$\epsilon \mid < type >< ident > [, < type >< ident >]^*$
$< body >$	$::=$	$\{[< stmt >]^*\}$
$\boldsymbol{< stmt >}$	$\boldsymbol{::=}$	$\boldsymbol{< simp >; \mid < return >; \mid < decl >; \mid < control >}$
$\boldsymbol{< control >}$	$\boldsymbol{::=}$	$\boldsymbol{if(< exp >) < stmt > [else < stmt >]}$
		$\boldsymbol{\mid while(< exp >) < stmt >}$
$< decl >$	$::=$	$< type >< ident > [, < ident >]^*$
$< simp >$	$::=$	$< ident >=< exp >$
$< return >$	$::=$	$return < exp >$
$< exp >$	$::=$	$(< exp >) \mid < const > \mid < ident > \mid$
		$< exp >< binop >< exp > \mid < callee >$
$< callee >$	$::=$	$< ident > (\epsilon \mid < exp > [, < exp >]^*)$
$< ident >$	$::=$	$[A - Z\_a - z][0 - 9A - Z\_a - z]^*$
$< const >$	$::=$	$< intconst > \mid < doubleconst >$
$< binop >$	$::=$	$+ \mid - \mid * \mid <$
$< intconst >$	$::=$	$[0 - 9][0 - 9]^*$
$< doubleconst >$	$::=$	$< intconst > . < intconst >$
$< type >$	$::=$	$int \mid double$

表 1: 语法产生式

- 完善其他运算符
- .....

这些实现可以参考 Tutorial 以及在 LLVM File 中查找相关 api。有想实现的功能遇到困难也可与助教进行联系讨论。

2 实验测试

lab5 实验的 baseline 的测试用例为压缩包中的 *input_example.data*，*main.cpp* 以及 *output_example.data*, 该测试用例为编译一个求 Fibonacci 数列第 n 位的函数，在配置好环境后，在当前目录的命令行使用之前相同的编译方式可得到输出结果，可将其与 *output_example.data* 进行对比。

3 实验提交

实验完成之后，将实验代码在截止日期之前上传至 elearning，与实验代码需要一齐提交的有一个最终实现的编译器支持的语法产生式，以及一份描述本次实验实现的思路的实验报告 (产生式可包含在报告中)。将以上内容压缩到以学号命名的压缩包中 (zip 格式)，提交压缩包文件。代码的文件名为 lab5.cpp。对于完整实现 baseline 的作业，可以得到 50% 基本分数，对于额外实现了一些基本 c 语言语法功能的 (例如参考部分的内容)，根据实现的难度和实现的质量进行额外加分，该部分最多可以得到 40% 分数。对于额外实现了一些特色且有用的语法功能的，可以得到剩下的 10% 分数。

4 实验展示

在最后一节课上 (16 周)，上机课与上课连堂，课上由同学依次对最终实验实现的内容进行展示，展示内容包括介绍实现的语法功能，介绍思路，以及给出例子和现场展示，每人十分钟左右。