Lab3实验报告

PB20111633
阿非提

实验要求

- 需要完善 `src/cminusfc/cminusf_builder.cpp`

实验难点

- Debug
- 理解编译器生成 llvm 代码的过程

实验设计

- 在不同的函数中会生成llvm的指令，这些指令中会存在需要之前的指令返回的值作作为形参的指令。所以需要使用全局变量，在不同的函数里转递这些值。为此声明以下全局变量：

```
Value* factor_return_val;
Value* var_return_val;
Value* expression_return_val;
Value* additive_expression_return_val;
Value* term_return_val;
bool assignmentCallForVar;
```

- 每个全局变量都在相对应的函数里被赋值进行值传递。其中assignmentCallForVar用来在`void CminusfBuilder::visit(ASTVar &node)`函数中判别调用函数，减少不必要的指令的生成。
  如被`void CminusfBuilder::visit(ASTAssignExpression &node)`调用，则不需生成load指令。否则，需要生成load指令，并把load的结果返回给上层函数。

- 根据被返回的全局变量的数据类型和当前函数中所需要的生成的指令所需的数据类型进行比较，只有数据类型不匹配时才生成相应的类型转换从而减少不必要的指令。如在
  `void CminusfBuilder::visit(ASTAssignExpression &node)`
  中根据var_return_val指向的地址的数据类型可知需要在此地址中存放的值的类型。再跟expression_return_val的值进行比较判断是否需要类型转换。如需要生成相应的指令，再生成store指令。

- 在插入新的基本块时需要注意生成相对应的跳转指令，引导控制流的正常流向。如在
  `void CminusfBuilder::visit(ASTSelectionStmt &node)`
  中，根据语法结构，生产 2（if结构）个或3（if else 结构）个基本块。并在if的结果为真时和存在else时，在相对应基本块的最后插入跳转指令跳转到下一个基本块。

- 除了实验要求以外，通过数据类型进行判断，对不能执行相对应的类型转换的情况和在作用域中找不到相对应的变量或函数时进行报错提醒以及中断程序。如在
  ```
  void CminusfBuilder::visit(ASTCall &node)
  ```
  中对找不到相对应的函数时报错提示：
  ```
  std::cout << "[compiler error]: Use of undeclared identifier '" << node.id << "'" << std::endl;
  ```
  如有参数类型不匹配（不能通过类型转换解决时）或实惨和形参数目不一致时报错提示：
  ```
  std::cout << "[compiler error]: Invalid parameter in '" << node.id << "'" << std::endl;
  ```

实验总结

- 通过本次实验，我对中间代码生成有了更深的理解，认识了在实际实验中所需要注意的点，学会了在不使用IDE的情况下进行debug的方法。