类型检查和中间代码生成说明

1 进行语义分析的前置工作

你需要先确保你完全理解以下问题再开始本次实验，否则你会遇到非常大的困难。

1. 你是否理解 c++ 的虚函数，为什么调用时候的 TypeCheck() 函数都是 tree_node 类，但实际上执行的却是不同子类的 TypeCheck()。
2. 你是否完全理解了语法树所有类的定义和语法树整体的结构，语法树的根节点是什么类，子节点有哪些类，子节点的子结点呢？.....
3. 你是否理解树的前序遍历，中序遍历，后序遍历算法。
4. 对于语法树节点 node1，node1 的含义为 addexp + mulexp，假设你知道了 addexp 和 mulexp 的类型，能否推出 node1 的类型？这与树的后序遍历有什么关联？
5. 你是否理解代码框架中 include/symtab.h 中的符号表类，当遇到了 block 节点时应该对符号表做什么操作？当遇到变量定义节点时应该对符号表做什么操作？
6. 你是否理解了 include/type.h 中的 VarAttribute 类和 NodeAttribute 类，这两个类中的成员变量作用都是什么？应该在什么地方使用？

2 语义分析算法示例

```
int a=5.5;
float b=3;
int main()
{
  int a=5;
  int a=10;
  return b;
}
```

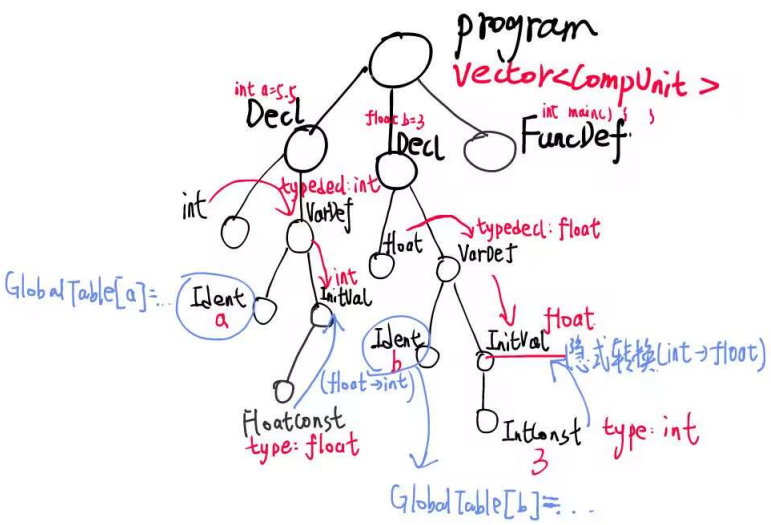

图 2.1: 语义分析示例

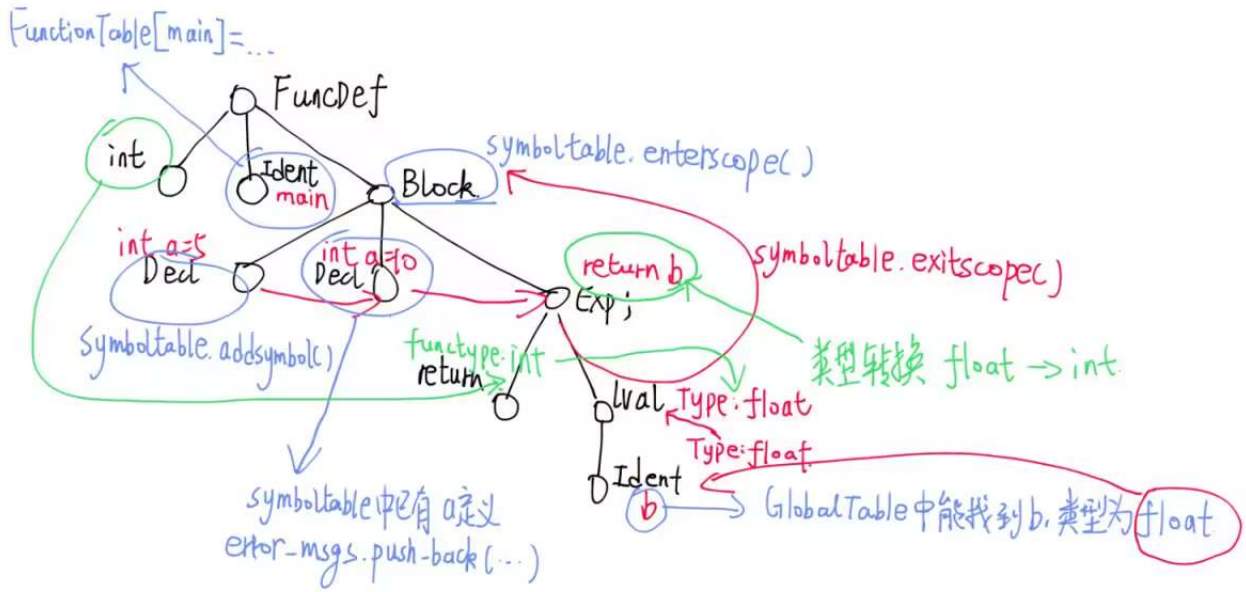

图 2.2: 语义分析示例 (接上一张图的 FuncDef)