MLIR 编译框架的使用与探索

吴非 519021910924

0.实验环境：Unbuntu 20.04 5.13.0-40-generic

1.实验过程：

首先编写词法分析器：

查看要求：变量名以字母开头；变量名由字母、数字和下划线组成；变量名中有数字时，数字应该位于变量名末尾。

能够识别“return”、“def”和“var”三个关键字

可以看到这两个情况都是以字母开头的。根据提示查看isalpha函数：判断是否是字母。isalnum判断是否是字母和数字。

查看后面的官方代码，以if判断开头，因此我也通过if (isalpha(lastChar))进入标识符|关键字的判断逻辑，这样就满足了变量名以字母开头的要求。之后进入while循环逐个读取判断是否是字母，数组或下划线，通过bool类型的lastDigit判断是否如果出现数字后之后的字符是数字，如果不是那么直接返回它，然后打印错误：



如果while循环结束没有执行这句返回错的标识符，那么就是三种关键字或者一个合法的标识符。直接先用if判断关键字即可。注释比较详细。

其次是语法分析器：

观察TODO的位置，首先parseVarDeclaration处主要写的是判断逻辑有三个TODO，观察附近代码，parseDeclaration中就有tok\_var,tok\_identifier的判断，直接复制即可。值得注意的是需要loc变量，搜索即可发现其他位置有这句auto loc = lexer.getLastLocation();而此函数是得到现在token的开始位置的，显然这里直接也是复制这句就行。最后一个TODO位于parseType()，而此函数在上述的第三个TODO处被调用，可以观察到这个函数是检查[2][3]或者<2,3>等的合法性的（并返回一个type，存储具体的shape，type->shape.push\_back）。

同样我根据对<>的代码模仿编写[][]的:不同于<>，[2][3]需要判断一个[]和数字后改变一个bool变量ismedium从而辨别此时位于第一个[]还是第二个[]。

其整体判断逻辑是while循环判断是否是数字，是数字后加入type，得到下一个token后在当前循环判断是否是[]。

由于整个循环的条件是判断数字，循环内部单独处理[]的情况，因此如果出现%\*等奇怪的字符，while循环会直接结束，而最后这个字符如果不是]([]中没有数字也合法),会报错：return parseError<VarType>("]", "to end type");

注释也比较详细。

最后是优化部分：

阅读提示，其思想是，A=Transpose(Transpose(B)),A相当于代码中的op，通过getOperand得到A的操作数，或者说是输入：(Transpose(B)，对应代码中的变量transpose\_input)，对transpose\_input调用getDefiningOp得到它的操作符Transpose（如果是A=Transpose(B\*C),此时得到的就是\*），之后step 2对它用了一个智能指针的转换：dyn\_cast\_or\_null<TransposeOp>，就是如果得到的这个操作符是TransposeOp，那么返回这个指针，否则返回0。这样就实现了两个嵌套转置的判断。判断成功后进入step 3，通过replaceOp操作把op，也就是A整体替换为Transpose(B)的操作数，也就是B即可。

2.实验结果：

test1:



test2:



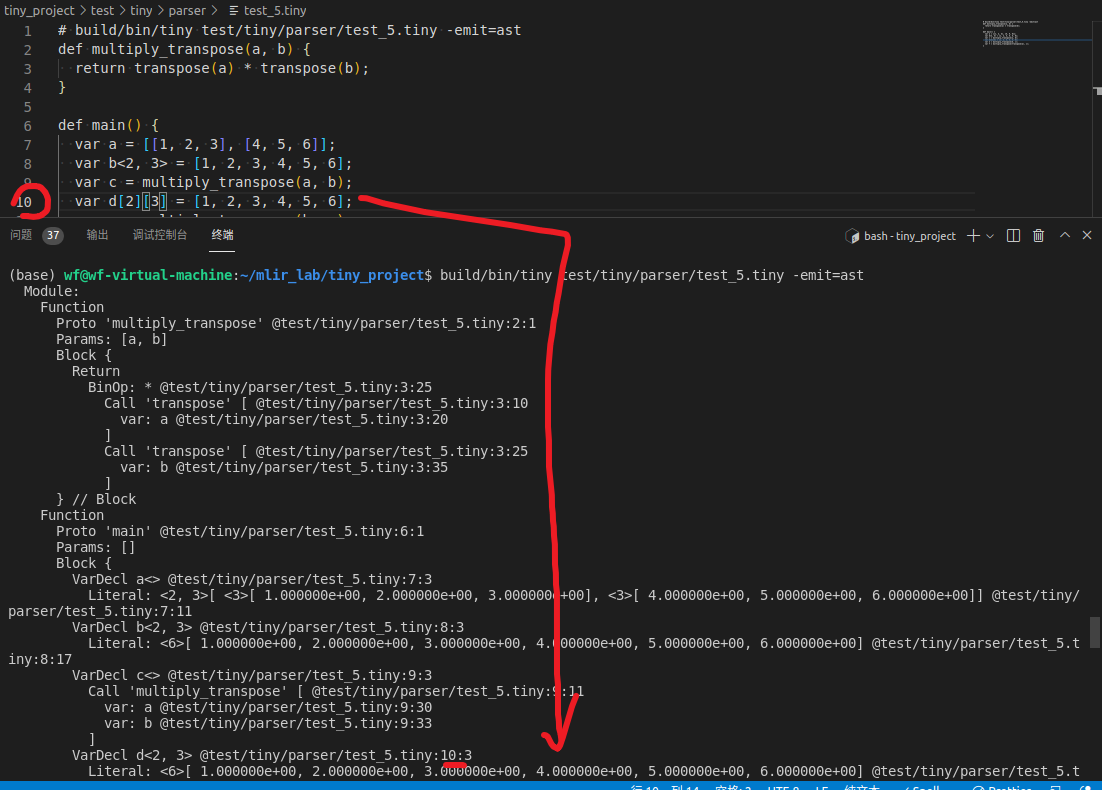
test3:

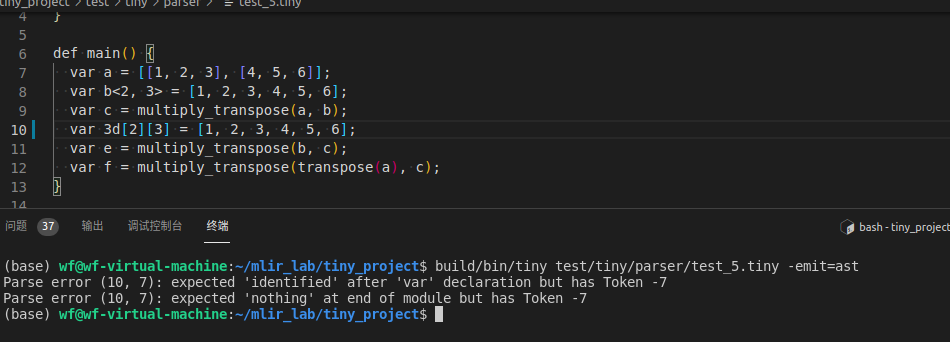


test4：

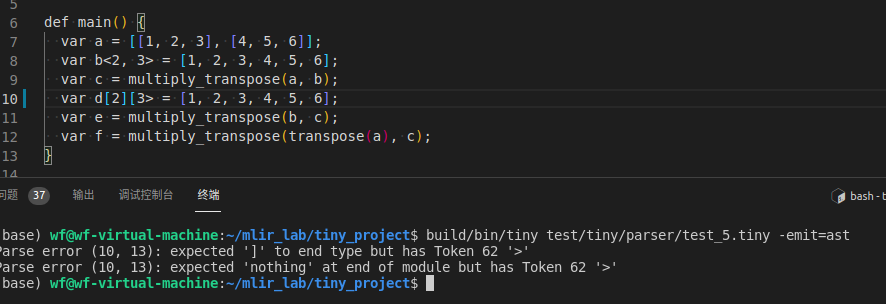


test5：









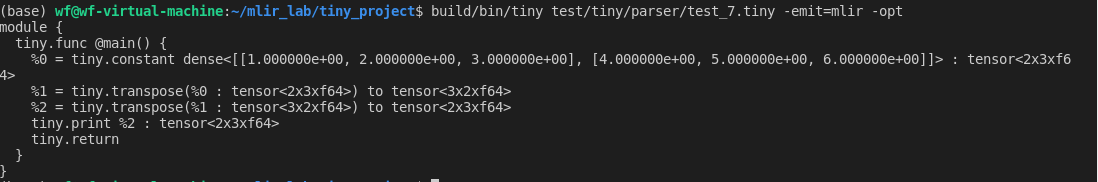
test6：



test7：

未优化：

opt:

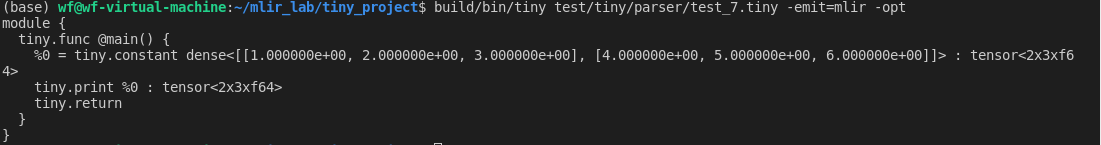
可以看到经过了两次转置

jit:



优化后：

opt:



可以看到没有多余的两次转置了。

jit:



矩阵结果与未优化一致。

3.总结：

本实验通过对MLIR编译框架中的词法分析，语法分析，转置冗余消除等代码的补充，锻炼了自己的代码能力，加深了对词法和语法分析器等的理解。

对MLIR的理解：提供一套中间模块，这个中间模块有两个作用：1. 对接不同的软件框架；2. 对接软件框架和硬件芯片。同时比较新颖的概念：方言，有方言内部和方言间的转换，这样提高了泛化和通用能力。

4.分工：

由吴非同学单独完成。

联系方式：邮箱：[legendary@sjtu.edu.cn](mailto:legendary@sjtu.edu.cn) 手机：19821286671

5.个人建议：

个人感觉任务难度和代码量都不是很大，可以一个人完成，小组作业懂得都懂。